Free!

KnowWare Basics!

Einfach & verständlich

Web Design

Text, Grafik, Links, HTML, Browser

Welche Programme?
Microsoft Frontpage
Adobe PageMill
Netscape Composer
Claris Home Page

Was sind:
Java • JavaScript
ActiveX • CGI •
StyleSheets ?

nützliche Adressen
Begriffserklärungen

Inhalt

- So erstellst Du Homepages und Websites für das World Wide Web oder ein Intranet...
- baust schnelle Seiten, die in jedem Browser funktionieren...
- und gestaltest gut aussehende Webseiten, die auch anwendbar sind: Planung, Struktur, Seitenlayout, Typografie, Navigation, Benutzeroberfläche
- Wie kommen Deine Seiten ins Netz? Wie finden andere zu ihnen?

▶ *Lies weiter...*

Olav Junker Kjær

www.KnowWare.de

KnowWare Basics!

Acrobat Reader: Wie ...

F5/F6 öffnet/schließt die Ansicht **Lesezeichen**

Strg+F sucht

Im Menü Ansicht stellst du ein, wie die Datei gezeigt wird

STRG+0 = Ganze Seite **STRG+1** = Originalgrösse **STRG+2** = Fensterbreite

Im selben Menü kannst du folgendes einstellen:: **Einzelne Seite**, **Fortlaufend** oder **Fortlaufend - Doppelseiten** .. Probiere es aus, um die Unterschiede zu sehen.

Navigation

Pfeil Links/Rechts: eine Seite vor/zurück

Alt+ Pfeil Links/Rechts: Wie im Browser: Vorwärts/Zurück

Strg+**+** vergrößert und **Strg**+**–** verkleinert

Bestellung und Vertrieb für den Buchhandel

Bonner Pressevertrieb, Postfach 3920, D-49029 Osnabrück
Tel.: +49 (0)541 33145-20
Fax: +49 (0)541 33145-33
bestellung@knowware.de
www.knowware.de/bestellen

Autoren gesucht

Der KnowWare-Verlag sucht ständig neue Autoren. Hast du ein Thema, daß dir unter den Fingern brennt? - ein Thema, das du anderen Leuten leicht verständlich erklären kannst?

Schicke uns einfach ein paar Beispielseiten und ein vorläufiges Inhaltsverzeichnis an folgende Adresse:

lektorat@knowware.de

Wir werden uns deinen Vorschlag ansehen und dir so schnell wie möglich eine Antwort senden.

www.knowware.de

Einleitung

Webdesign ist die Kunst, eine Website zu erstellen – also eine Seite oder mehrere miteinander verbundene Seiten, auf die man mit Hilfe eines Webbrowsers über das Internet zugreifen kann. Das Heft will Anfängern wie Fortgeschrittenen etwas bringen – also jedem, der lernen möchte, wie man Webdesign praktiziert. Wir wollen unser Thema vom simplen HTML bis zu komplizierten Fragestellungen betrachten.

Voraussetzungen

Ich gehe davon aus,

- daß Du mit dem Web einigermaßen vertraut bist, weißt, wie Du hier Informationen findest, und Ausdrücke wie Link, Browser und dergleichen kennst. Bist Du ein Neuling im Web, empfehle ich Dir das Heft *Auf ins World Wide Web*.
- daß Du weißt, wie Du einen Computer benutzt und ein Programm installierst oder was ein Betriebssystem oder eine Bildschirmauflösung ist, und daß Du keine Scheu davor hast, zu experimentieren und in Programmen herumzuprobieren.
- daß Du Englisch verstehst, da ein großer Teil der Software, der Hilfestellungen und der Dokumentation nur auf Englisch vorliegt.
- daß Du Zugang zum Internet hast. Einen Webserver benötigst Du allerdings nicht, um mit Webdesign zu experimentieren.

Wie Du dieses Heft benutzt

Du solltest das Heft am Computer lesen. Es wird Dir mehr bringen, wenn Du herumspielst und die Dinge ausprobierst, die ich beschreibe.

Du kannst das Heft systematisch wie ein Lehrbuch lesen oder es als Nachschlagewerk nutzen. Ich gehe nicht vom Aufbau bestimmter Programme aus, sondern von den Möglichkeiten und Begrenzungen im Web. Allerdings gehe ich darauf ein, wie Du verschiedene Editoren benutzt – vor allem *Adobe PageMill*, aber auch *Claris Home Page, Netscape Composer* und *Microsoft Front Page*.

Aufbau

Das Heft besteht aus vier Teilen. Der erste fängt mit den Grundlagen an und erklärt, was für eine einfache Website benötigt wird und wie Du sie zum Laufen bringst. Der zweite Teil befaßt sich näher mit verschiedenen Aspekten im Webdesign. Im dritten Teil sehen wir uns etwas kompliziertere Themen an wie Multimedia und Interaktivität – Themen, für die vermutlich nicht jedermann Zeit und Lust aufbringt.

Der vierte und vielleicht wichtigste Teil des Heftes beschäftigt sich mit der übergeordneten Planung einer Website: Struktur, Navigation, Textaufbau, Seitenlayout und Typographie. Dieser Teil ist der letzte – er erfordert nämlich einen gewissen Überblick über die technischen Möglichkeiten.

Abschließend findest Du ein Sachwortregister und einen Index. Das Inhaltsverzeichnis befindet sich auf dem hinteren Deckblatt.

Erster Teil – Webdesign für Anfänger

Was ist eigentlich Webdesign?

Webdesign ist eine Kunst, die folgende Bereiche umfaßt:

Planung und Struktur – Sollen die Benutzer Deiner Website sich problemlos in ihr orientieren, mußt Du ihren Aufbau und ihre Struktur genauestens überdenken. Ein notwendiger Punkt ist eine 'Benutzeroberfläche' für die Homepage, also Texte, Symbole und Schaltknöpfe, die dem Benutzer beim Navigieren helfen.

Herstellung der eigentlichen Seiten – Die Seiten erstellst Du in einem Webeditor, in dem der Text zusammengestellt und formatiert und Grafiken eingefügt werden. Hier erstellst Du weiterhin die Links zu den anzusprechenden Seiten.

Veröffentlichung – Ist die Website fertig, muß sie im Internet publiziert werden, damit andere auf sie zugreifen können.

Wartung und Aktualisierung – Eine Website ist im allgemeinen kein endgültiges Produkt. Sie muß also ständig modifiziert und aktualisiert werden.

Wie schwierig ist das eigentlich?

Bist Du mit der Arbeit in Textverarbeitungen vertraut, sollte Dir ein Webeditor keine Schwierigkeiten machen. An einige Dinge mußt Du Dich allerdings gewöhnen:

Webseiten erlauben, zumindest bisher, nicht soviel Kontrolle über Typographie und Layout wie Textverarbeitungen. Also mußt Du Dich mit einfacheren Möglichkeiten begnügen.

Du mußt mit den unterschiedlichen technischen Voraussetzungen Deiner Leser rechnen. Im Internet gibt es viele verschiedene Computertypen mit unterschiedlichen Bildschirmgrößen, Auflösungen, Zeichensätzen und so weiter, wie ja auch die verschiedenen Browser unterschiedliche Möglichkeiten haben. Es reicht also nicht, daß Deine Webseiten auf Deiner Maschine gut aussehen – sie müssen auch auf anderen Computern funktionieren.

Textverarbeitungen bieten keine Möglichkeiten für Links, also Verknüpfungen, zu anderen Seiten oder Websites. Doch gerade das ist einer der spannendsten Aspekte im Web.

Schließlich besitzen Textverarbeitungen ebenfalls nicht die Möglichkeit, eine Website zu publizieren. Das ist jedoch nicht weiter kompliziert.

Das Web bietet übrigens viele Möglichkeiten für Effekte – Stichworte sind Multimedia und Interaktivität. Dafür solltest Du Dich aufs Programmieren verstehen und außerdem künstlerisches Talent in irgendeiner Form mitbringen. Für die Erstellung einer professionellen und gut funktionierenden Website sind diese Effekte allerdings nicht absolut notwendig.

Der schwierigste Punkt beim Webdesign ist wohl, seine Informationen in einer logischen und funktionellen Struktur zu organisieren und die einzelnen Seiten ansprechbar und lesefreundlich zu gestalten. Das erfordert Planung, Nachdenken und ein Verständnis für die besonderen Möglichkeiten und Begrenzungen des Mediums. Vergiß nicht – dies ist ein neues Medium, das sich grundsätzlich von den bisher bekannten unterscheidet. Aber keine Angst – in diesem Heft findest Du alle notwendigen Informationen.

Wie läuft das eigentlich ab?

Sollen andere Leute über das Internet auf eine Website zugreifen, muß sie auf einem Webserver liegen – also einem unmittelbar an das Internet angeschlossenen Computer, der die erforderliche Kommunikationssoftware enthält. Im allgemeinen mietet man den notwendigen Raum bei einem Internet-Provider (siehe ab Seite 20). Ist man bei einem der großen Internet-Provider wie Compuserve, AOL oder T-Online Mitglied, kann man seine (private) Website ohne Zusatzkosten auf deren Server veröffentlichen.

Normalerweise solltest Du allerdings Deine Website auf einem gewöhnlichen PC oder Mac testen und sie erst dann publizieren, also auf den Server überführen, wenn sie fertig ist. Also kannst Du so gut wie alles in diesem Heft ausprobieren, auch ohne Platz auf einem Server gemietet zu haben.

Was für einen Computer brauchst Du?

Win95/NT-PCs oder der Macintosh eignen sich beide gut für Webdesign. Ich würde nicht empfehlen, die Sache unter Betriebssystemen wie Win3.1, DOS oder OS/2 auszuprobieren – nicht etwa daß diese Systeme nicht in Ordnung wären, aber so gut wie alle neue Internet-Software wird für Win95/NT oder das Mac-OS geschrieben. Im allgemeinen erscheinen neue Programm zuerst für Win95/NT und einige Monate später für den Mac – willst Du also unbedingt jederzeit das Neueste haben, kommst Du nicht an Win95/NT vorbei. Wir gehen hier von Win95 aus, auf dem Mac laufen die meisten Dinge aber genauso ab (die *rechte Maustaste* auf dem PC entspricht auf dem Mac im allgemeinen einer gedrückten Maustaste).

Und wie stark sollte Deine Maschine sein? Die unterschiedlichen Programme fürs Webdesign sind nicht besonders anspruchsvoll, aber vielfach benötigt man mehrere Programme gleichzeitig. Viel Arbeits-

speicher (RAM) ist also wichtiger als die Prozessorgeschwindigkeit und -kraft.

Ich habe selbst etwa ein Jahr lang professionelles Webdesign auf einem 486er mit 12 MB RAM erstellt. Das ist sicher etwas knapp, und weniger würde ich auf keinen Fall empfehlen – aber genau genommen braucht man auch nicht mehr.

Ein guter Bildschirm (wenigstens 15 Zoll) und eine solide Tastatur sind im allgemeinen mehr wert als hohe Prozessorkraft.

Selbstverständlich benötigst Du eine Internet-Verbindung auf Deiner Maschine.

Welche Programme brauchst Du?

Ein einziges Programm, das alle Bedürfnisse eines Webdesigners deckt, gibt es nicht. Du benötigst einen Werkzeugkasten mit verschiedenen Programmen, und die Kunst liegt darin zu wissen, wann man welches Werkzeug benutzt. Die wichtigsten Programme sind:
Einige Browser: *Netscape Navigator* und *Microsoft Internet Explorer* (MSIE).
Ein Webeditor. Dieses Programm benutzt Du, um die Seiten zu erstellen. Gute Webeditoren sind z.B. *Adobe Pagemill, Netscape Composer*, *Claris Home Page* oder *Microsoft FrontPage*.
Eine Textverarbeitung, z.B. *Microsoft Word*, zur Vorbehandlung umfangreicher Texte.
Ein Grafikprogramm, z.B. *Adobe Photoshop* oder *Paint Shop Pro,* um Grafik vorzubehandeln.

HTML

Webseiten sind Dateien im Dokumentenformat *HTML*. Das Kürzel steht für *HyperText Markup Language* – ein Dokumentenformat, das formatierten Text und Hyperlinks enthält und außerdem mit Bildern und anderen Elementen angereichert werden kann.

Als Webdesigner solltest Du Dir den Unterschied zwischen HTML-Dokumenten und den üblichen Text- und Grafikdokumenten bewußt machen. Eine Textverarbeitung hat die Aufgabe, Dokumente zu entwickeln, die im allgemeinen auf Papier ausgedruckt werden. HTML wurde speziell entwickelt, um Informationen in Netzwerken zu veröffentlichen – also Informationen, die vor allem auf Computern gelesen werden.

Das Internet und offene Standards

Die Stärke des Internets ist, daß es auf offene und plattformunabhängige Standards baut.
Der Ausdruck ‘offener Standard’ ist zwar etwas ungenau. Im allgemeinen bedeutet er folgendes:
Der Standard ist niemandes Eigentum, niemand hat Patent- oder Urheberrechte an ihm.
Der Standard wird von einer nicht-kommerziellen Organisation koordiniert.
Seine Spezifikationen stehen jedermann kostenlos zur Verfügung.
Jeder darf Software herstellen, die zum Standard kompatibel ist – kommerzielle wie nicht-kommerzielle.
Plattformunabhängig bedeutet, daß man nicht an bestimmte Hardware oder ein bestimmtes Betriebssystem gebunden oder davon abhängig ist. Das Internet stützt sich auf plattformunabhängige Standards und funktioniert dadurch unterschiedslos auf Plattformen wie PC, Mac, Amiga oder UNIX. Im Unterschied hierzu funktionieren die meisten Spiele, Programme, CD-ROMs und so weiter nur auf einer bestimmten Plattform, für die sie entwickelt wurden. Hier ist zu beachten, daß es die Standards sind, die offen und plattformunabhängig sind – nicht etwa die einzelnen Programme, die in Verbindung mit dem Internet benutzt werden. Diese sind im allgemeinen kommerziell und an eine bestimmte Plattform gebunden.

HTML ist plattformunabhängig

Webseiten müssen auf Computern jeder Art lesbar sein – auf Systemen mit den unterschiedlichsten Betriebssystemen, Bildschirmgrößen und grafischen Fähigkeiten. Eben das ist die Stärke im Web – man ist nicht mehr in einem Durcheinander inkompatibler Systeme gefangen, sondern jedermann kann mit allen kommunizieren.

Das bedeutet aber auch, daß man nie genau voraussehen kann, wie ein anderer Computer eine Seite wiedergibt. Z.B. unterscheiden sich die verschiedenen Systeme in Schriftgröße, Fensterumfang und Bildschirmauflösung. Eine Webseite erlaubt einfach nie den gleichen Grad an Kontrolle wie eine gedruckte Seite.

Der Formatierungscode in HTML gibt daher auch nicht das genaue Aussehen der Textteile an (‘dieser Satz steht in 24 Pkt Helvetica’), sondern stützt sich auf übergeordnete Anweisungen (‘dieser Satz ist eine Überschrift 3. Grades’). Grundsätzlich bestimmt der Empfängercomputer, wie diese Anweisungen wiedergegeben werden. In der Praxis geben die üblichsten Browser allerdings die HTML-Strukturen in etwa identisch wieder. Man kann also durchaus ein Layout in HTML herstellen, das die *meisten* Browser *in etwa* wie geplant wiedergeben werden.

HTML ist ein offenes Format

HTML ist ein offenes Format.

Es gibt eine Organisation namens *World Wide Web Consortium* (*W3C*), die den HTML-Standard koordiniert – sie gibt die offiziellen Spezifikationen heraus und schlägt anzuwendende Standards vor. Die Durchschlagskraft dieser Standards hängt aber ganz und gar davon ab, ob Webdesigner und Softwareproduzenten diese Empfehlungen befolgen oder nicht. Z.B. hat Netscape etliche selbstentwickelte Erweiterungen des HTML-Standards eingeführt, die später offiziell akzeptiert wurden.

Heute wird HTML vom sogenannten Browserkrieg geprägt – einem Kampf zwischen den führenden Entwicklern von Browsern, Netscape und Microsoft, um die Herstellung des 'potentesten' Browsers. So entstanden etliche neue, nicht standardisierte HTML-Erweiterungen, die vielfach nur auf einem einzelnen Browser wirken, oder aber verschieden, je nach dem Browsertyp. Ein Webdesigner sollte also einen kühlen Kopf wahren, wenn er mit HTML-Erweiterungen experimentiert.

HTML funktioniert jederzeit

HTML ist *degradable* (aufwärtskompatibel), was bedeutet, daß man neuen Code in ein HTML-Dokument setzen kann, ohne die Funktionsfähigkeit älterer Browser zu beeinträchtigen. Diese Browser ignorieren schlicht Code, der ihnen nicht bekannt ist, und verarbeiten das übrige Dokument wie gewohnt. Ein älterer Browser kann also ohne weiteres ein Dokument darstellen, das neuere oder nicht standardisierte HTML-Erweiterungen enthält.

Der Browser

Der Browser ist sozusagen das Publikum des Webdesigners. Darum solltest Du gut Freund mit ihm sein:
Navigator 2. Wird im Web nicht mehr angeboten, ist aber immer noch recht verbreitet. Wäre wohl der kleinste gemeinsame Nenner.
Navigator 3. Der meistbenutzte Browser.
Navigator Gold 3. *Navigator 3*, um einen Webeditor erweitert.
Netscape Communicator. Der Nachfolger von *Navigator 3*. Eigentlich eine 'Suite' von Internet-Software, wobei der Browser *Navigator 4* nur eine Komponente von mehreren ist. Eine weitere dieser Komponenten ist der *Composer*, ein mitgelieferter HTML-Editor. Dabei handelt es sich um eine Weiterentwicklung des Editors von *Navigator Gold 3*.

Netscape Navigator wird oft einfach Netscape genannt. Um Verwechslungen zu vermeiden, schreibe ich aber Navigator, wenn ich den Browser meine, Composer, wenn es um den Editor geht, und Netscape, wenn die Firma angesprochen ist.

Netscape-Software ist Shareware, die 90 Tage lang kostenlos benutzt werden kann. Du findest sie unter `www.netscape.com`.

Microsoft Internet Explorer (MSIE) ist der schärfste Konkurrent von Netscape. Er entspricht in etwa *Netscape Navigator/Communicator*. *MSIE* ist Freeware – Du findest das Programm unter `www.microsoft.com/ie`.
Natürlich gibt es auch andere Webbrowser, wie Apples *CyberDog*, die Textbrowser *Lynx* oder *Amaya*, den Browser für Individualisten *Opera*, den Nostalgiebrowser *Mosaic* und viele andere. Keiner dieser letzteren Browser ist aber recht verbreitet. Es gibt auch noch ältere Versionen von Navigator und MSIE, die sind aber inzwischen recht selten.

Jeder Webdesigner sollte wenigstens *MSIE 3* und *Navigator 3* haben. Auch wenn Du einen neueren Browser wie *Navigator 4* hast, solltest Du Deine Seiten möglichst in *Navigator 3* und *MSIE 3* testen.

Der Webeditor

Das wichtigste Werkzeug des Webdesigners ist der Webeditor. Ein solches Programm, das auch HTML-Editor heißen kann, wirkt auf den ersten Blick wie ein Zwischending zwischen einem Browser und einer Textverarbeitung: Du siehst Deine Webseiten in etwa so, wie sie in einem Browser erscheinen, kannst aber Text in die Seiten einfügen und sie redigieren und formatieren.

Welcher Editor?

Es gibt mehrere Editoren, die für verschiedene Bedürfnisse geeignet sind:

Adobe PageMill 2.0 ist meiner Meinung nach der beste Editor für kleinere Websites. Zur Zeit läßt sich eine Probeversion unter der Adresse `www.adobe.com` downloaden.

Claris Home Page ist nicht so unmittelbar einleuchtend wie PageMill, aber ebenfalls ein ausgezeichnetes Programm. Eine Probeversion findest Du unter `www.claris.com`.

Netscape Composer ist ein Bestandteil von *Netscape Communicator* (*Netscape Navigator 4*). Die vorherige Version hieß *Netscape Gold*. Der Composer ist Shareware – Du kannst das Programm 90 Tage lang kostenlos benutzen. Studenten können dieses Programm auch weiterhin kostenlos benutzen. Du findest es unter `www.netscape.com`. Das Programm hat nicht so viele Möglichkeiten wie PageMill, dafür ist es aber praktisch, weil es so eng mit dem Browser verbunden ist.

Microsoft FrontPage ist ein sehr guter Editor, der sich besonders für die Arbeit an größeren Homepages mit zahlreichen Seiten eignet. Eine Probeversion erhältst Du, indem Du Dich – kostenlos – bei Microsofts *Site Builder Network* anmeldest: `www.microsoft.com/sitebuilder`.

Alle diese Webeditoren gibt es für PC und Mac. Editoren dieses Typs werden oft 'WYSIWYG'-Editoren genannt. Die Abkürzung steht für 'What You See Is What You Get' – in diesen Editoren siehst Du Deine Webseite also in etwa so, wie sie von einem Browser dargestellt wird. Das ist ein ziemlicher Fortschritt gegenüber der vorherigen Generation von Editoren, in denen man unmittelbar im HTML-Code arbeitete und die Formatierungscodes von Hand einsetzte. Wenn man wissen wollte, wie die Seite aussehen würde, mußte man einen Browser starten.
Das populärste Programm dieses älteren Typs ist Hot Dog (`www.sausage.com`). Ich würde seine Verwendung allerdings nur besonders technisch interessierten ober aber leicht masochistisch veranlagten Webdesignern empfehlen.

Die neuesten Versionen von *Word*, *WordPerfect* und *PageMaker* können ein Dokument auch als HTML-Dokument speichern. Meiner Meinung nach wird es aber noch eine Weile dauern, bis sie sich mit den 'echten' Webeditoren messen können. Zukünftige Versionen dieser Programme mögen aber durchaus für die Produktion von Webseiten mittleren Niveaus geeignet sein. Für Webseiten von professionellem Standard wird man sicher immer spezielle Webeditoren vorziehen. (Eine sehr lobenswerte Ausnahme stellt das deutsche Produkt *StarOffice 4.0* dar, welches weit über die Möglichkeiten von Word und Co. hinausgeht.)

Grundsätzlich beschreibe ich konkrete Arbeitsaufgaben für Editoren in allen vier oben genannten Hauptprogrammen. Es geht mir aber mehr darum, die Möglichkeiten und Begrenzungen von HTML im Web als solche zu beschreiben, und weniger um technische Finessen einzelner Programme. Also sollte Dir das Heft auch dann etwas bringen, wenn Du einen anderen Editor als die vier genannten benutzt.

Fangen wir an!

Der Webordner

Bevor Du mit Deiner ersten Webseite loslegst, solltest Du einen Webordner erstellen. Hier speicherst Du alle Dateien, die zu Deiner Website gehören: HTML-Dateien, Grafikdateien und so weiter. Die Dateien können problemlos in Unterordnern des Webordners liegen – keinesfalls aber in anderen Ordnern auf Deiner Maschine. Da der Inhalt dieses Webordners auf Deinem Webserver gespeichert wird, sollte dieser Ordner auch ausschließlich Webdokumente enthalten.

Wie der Webordner heißt, das ist im Grunde gleichgültig – Du kannst ihn also z.B. schlicht 'Webordner' nennen. Es spielt auch keine Rolle, wo auf der Maschine er gespeichert wird. Das einfachste ist wohl, wenn Du ihn auf dem Desktop oder im Stamm des Laufwerks anbringst.

Änderst Du einen Dateinamen oder verschiebst eine Datei in einen anderen Ordner, *nachdem* Du Links zu dieser Datei oder aus ihr eingerichtet hast, funktionieren sie nicht mehr. Das solltest Du also möglichst bleiben lassen.

Regeln für Dateinamen

Es gibt bestimmte Regeln für die Benennung von Dateien und Ordnern im Web. Ein Dateiname darf nur englische Buchstaben, Zahlen, Bindestriche (-) und den sogenannten Unterstrich (_) enthalten. Zwischenraum, ä ö ü oder Sonderzeichen wie *!"#?* usw. sind also verboten.

Der Dateiname muß mit einer Endung abschließen, die den Dateityp angibt. Diese Endung wird durch einen Punkt vom eigentlichen Namen getrennt. Ein HTML-Dokument wird durch die Endung `.html` gekennzeichnet – wie z.B. `meine_erste_webseite.html` . Die Endungen müssen unbedingt korrekt sein. Ordnerbezeichnungen haben keine Endung – und also auch keinen Punkt. Benutze ausschließlich Kleinbuchstaben.

Die Regeln für Dateinamen unterscheiden sich je nach Server. Die hier angeführten Regeln gelten aber ganz allgemein – benutzt Du sie, werden also vermutlich keine Probleme auftauchen. Die Startseite Deiner Homepage sollte einen besonderen Dateinamen tragen. Im allgemeinen ist das `index.html`. Jeder Server hat aber seine eigenen Regeln, also solltest Du Deinen Provider fragen.

Ältere Server akzeptieren unter Umständen nur die alten 8.3-DOS-Dateinamen. In diesem Fall benutzt Du die Endung `.htm` anstatt `.html`. Darauf wirst Du aber vermutlich aufmerksam gemacht.

Deine erste Webseite

Wenn Du den Editor startest, stellt er eine leere Seite bereit. Du speicherst sie wie üblich mit File|Save (Datei|Speichern) – natürlich im Webordner oder einem Unterordner - und mit der Endung `.html`.

Neue leere Seiten erstellst Du über File|New (Datei|Neu). In *FrontPage* und *Composer* kannst Du eine neue Seite auch nach einer Vorlage erstellen, wie etwa Standard-Seite, Standard-Homepage, Standard-Clubseite oder dergleichen, die Du an Deine Bedürfnisse anpassen kannst. Allerdings sind diese Vorlagen im allgemeinen ziemlich geschmacklos und mißlungen.

Eine neue Seite zeigt sich zunächst einmal als leere graue oder weiße Arbeitsfläche. Darüber schweben die Menüleiste und ein paar Symbolleisten. Wie bei einer normalen Textverarbeitung kannst Du Deinen Text unmittelbar in der Arbeitsfläche schreiben.

Eigenheiten von Microsoft FrontPage

Microsoft FrontPage funktioniert auf etwas spezielle Weise. Das Programm besteht aus zwei Komponenten: dem *FrontPage Editor*, einem Webeditor wie die anderen hier beschriebenen, und dem *FrontPage Explorer*, einem 'Site-Editor', der die gesamte Website, also den Webordner samt Inhalt, sozusagen einkapselt. Die einzelnen Seiten der gesamten Website werden aus dem FrontPage Explorer heraus aktiviert und bearbeitet.

Der *FrontPage Explorer* erstellt einige Spezialordner im Webordner und führt eine etwas spezielle „doppelte Buchführung" mit zwei Ausgaben von jeder Datei – der Benutzer merkt aber nichts von diesen Spitzfindigkeiten. Dieses System gibt FrontPage einige Möglichkeiten zur Verwaltung der gesamten Website, die sich in den anderen Web-Editoren nicht finden.

Damit muß man aber richtig umgehen:

In diesem Programm erstellt man keinen Webordner – statt dessen wählt man im *FrontPage Explorer* File|New FrontPage Web (Datei|Neu - FrontPage Web).

Man kann auch nicht Dateien ohne weiteres in den Webordner kopieren – man muß File|Import (Datei|Importieren) aktivieren und so Dateien in das System importieren.

Außerdem kopiert man auch nicht den Inhalt des Webordners auf den Server – über das Hilfsprogramm *FrontPage Publishing Wizard (FrontPage Web publizieren)* werden nur die notwendigen Dateien überführt.

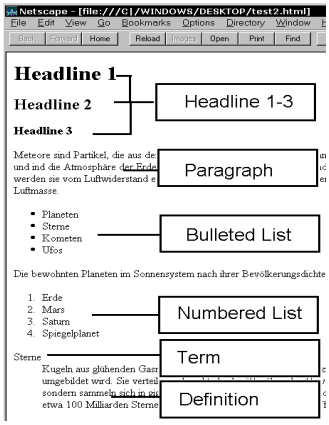
Text

Text schreibst Du genauso wie in einer Textverarbeitung. Der Umbruch erfolgt automatisch, wenn der Seitenrand erreicht wird. Neue Absätze erstellst Du mit der Return-Taste. Sie werden durch eine Leerzeile getrennt. Jeder Abschnitt wird in seinem Erscheinungsbild durch einen bestimmten *Typ (Stil, Vorlage)* gekennzeichnet.

Diesen Absatztyp wählst Du in einem Rollmenü auf der Symbolleiste. Normalerweise erscheint hier Paragraph (oder Standard bzw. Normal), also normaler Textkörper. Gehst Du das Rollmenü durch, siehst Du eine Reihe von weiteren integrierten Absatztypen. Zunächst wirst Du vermutlich in erster Linie die sechs möglichen Überschriftstypen benutzen. Aktivierst Du ein neues Absatzformat, wird der gesamte aktuelle Absatz entsprechend formatiert.

Absatzformate

Absatzformatierungen gelten für einen oder mehrere Absätze.



Paragraph (Standard) ist ein Absatz mit normalem oder Körpertext (auch *normal* genannt).

Heading 1-6 (Überschrift 1-6) sind die integrierten Überschrift-Niveaus – von der größten (1) bis zur kleinsten (6). Im allgemeinen erscheinen sie fett in graduierter Schriftgröße.

Listen sind Blöcke, die mehrere Absätze umschließen:

Bulleted List (Aufzählung) ist eine Liste, deren einzelne Punkte durch einen vorangestellten Punkt gekennzeichnet sind. **Directory List (Verzeichnisliste)** und **Menu List (Menüliste)** erscheinen wie **Bulleted List (Aufzählung)**.

Numbered List (Numerierung) ist eine Liste, deren Punkte numeriert sind.

Term (definierter Begriff) wechselt mit **Definition** in Definitionslisten. **Term** steht linksbündig und bezeichnet das zu definierende Wort. Der **Definition**-Absatz gibt die Definition des **Term** und hat diesem gegenüber einen größeren Einzug.

Absätze bzw. Punkte in Listen werden normalerweise nur durch einen Zeilensprung getrennt.

Address (Adresse) bedeutet, daß dieser Absatz eine Adresse angibt. Erscheint normalerweise kursiv und mit Einzug.

Preformatted oder formatted (Vorformatiert oder formatiert) – ein spezielles Format, das zum Beispiel für Computercode und modernistische Poesie verwendet wird, also Texte, bei denen Einzug und Zeilenwechsel inhaltlich wichtig sind. Im Gegensatz zu anderen Absatztypen erfolgt der Umbruch bei **preformatted** Text nicht nach der Breite des Browserfensters – willst Du die Zeile wechseln, mußt Du *Return* drücken. Einen Einzug erstellst Du durch mehrmaliges Drücken der Leertaste – was in normalen Absätzen nicht möglich ist. **Preformatted** Text wird mit Schreibmaschinenschrift dargestellt.

Absatzformatierung	
PageMill	Menü Format
Composer	Menü Format oder Paragraph/List Properties
FrontPage	Format|Paragraph oder Paragraph Properties (Format|Absatz)
Claris	Menü Format

Auf der Symbolleiste findest Du außerdem die Ausrichtung (rechtsbündig, linksbündig, zentriert) und Einzug bzw. negativen Einzug. Diese Formatierungen gelten ebenfalls für den gesamten aktuellen Absatz. Andere Formatierungen gelten nicht unbedingt für einen gesamten Absatz – wie etwa I (kursiv) und B (fett).

Zeichenformatierungen

Folgende Formatierungen gelten für ein Textstück innerhalb eines Absatzes:

Italic (*I*)	Kursiv
Bold (**B**)	Fett
Teletype (Schreibmaschine) (`TT`)	Text erscheint in Schreibmaschinenschrift, also einer Schrift, deren Buchstaben die gleiche Breite haben (meist *Courier*).
Font Size (Schriftgrad)	Schriftgröße – von 1 bis 7 graduiert, mit 3 als Normalgröße. Einige Editoren verwenden die Bezeichnungen von -2 bis +4, das bedeutet aber dasselbe.
Font Color	Bezeichnet die Textfarbe

Andere Zeichenformate (seltener verwendet):

Big	größerer Text; entspricht **font size** +1
Small	kleinerer Text; entspricht **font size** -1
Subscript	gesenkte Schrift (wie bei H_20)
Superscript	gehobene Schrift (wie bei $E=mc^2$)
Strike	durchgestrichener Text (~~wenn man denn unbedingt will~~)
Underline	unterstrichener Text (Vorsicht: – wird leicht mit Links verwechselt)

Es gibt auch Zeichenformate, die nicht genau angeben, wie ein Wort aussieht, sondern welche *Funktion* es im Text hat. Man könnte das *Sinnkodierungen* nennen:

	Bedeutung	*Aussehen*
Emphasis	Hervorhebung	kursiv
Strong	starke Hervorhebung	fett
Code	für Computercode	Schreibmaschinenschrift
Citation (Zitat)	Zitat oder Hinweis	kursiv
Sample	Beispiel für Output vom Computer	Schreibmaschinenschrift
Keyboard (Tastatur)	Benutzereingabe	Schreibmaschinenschrift
Variable	Variabel oder Argument für Befehl	kursiv im Navigator
Definition	eines Wortes, das am selben Ort definiert wird	kursiv im MSIE

Schriftformat ändern	
PageMill	Menü Style
Netscape Composer	Format|Character Properties, Registerblatt Character
FrontPage	Format|Font oder Font Properties (Format|Zeichen)
Claris	Menü Style

HTML – ein Rückblick

Wie Du siehst, enthält HTML sowohl Kodierungen, die die *Bedeutung* eines Textes kennzeichnen (Überschrift, Adresse, Zitat und so weiter) als auch Kodierungen, die sein *Aussehen* definieren (Schriftgröße, Farbe und so weiter). Die ursprünglichen Markierungen sind die Bedeutungscodes – in den ersten Ausgaben von HTML gab es nur solche.

Es hat zwei große Vorteile, ein Dokumentenformat auf Sinnkodierungen aufzubauen:
Zum einen hängen solche Kodierungen nicht von einem bestimmten Medium ab: eine 'Schreibmaschinen'-Kodierung ist sinnlos auf einem Terminal, wo alles in dieser Schrift erscheint, und kursiv bedeutet nichts, wenn der Text jemandem vorgelesen wird, der blind ist. Typographische Kodierungen sind von dem Medium abhängig, in dem sie erstellt wurden, während Sinnkodierungen in jedem Medium verwendbar sind.

Zum andern können Such- und Index-Maschinen einen Text auf seine Sinnkodierungen analysieren, also auf der Basis von Überschriften, Wortdefinitionen, Listen und so weiter einen Index generieren. So etwas ist für größere Websites äußerst nützlich, etwa für Universitäten. Und eben das ist einer der größten Vorzüge digitaler Information gegenüber bedrucktem Papier.

Als sich das Web allmählich weiter verbreitete, begann Netscapes Browser seinen Siegeszug, indem er die Formatierung des Erscheinungsbildes von Text unterstützte: Schriftgröße, Text- und Hintergrundfarbe und so weiter. Netscape erkannte früh, daß das Interesse an grafischen und layoutmäßigen Möglichkeiten in HTML mit der Verbreitung des Web anwachsen würde.

Das *World Wide Web Consortium* (W3C), das den HTML-Standard definiert, wollte zunächst nichts von Netscapes Erweiterungen wissen und arbeitete an einem eigenen Standard HTML 3.0. Die Öffentlichkeit, das heißt die Webdesigner und Browserbenutzer, wählte aber Netscapes Weg, wodurch W3C plötzlich die Kontrolle über die Entwicklung von HTML verlor. Erst als man in den sauren Apfel biß, HTML 3.0 aufgab und statt dessen HTML 3.2 vorschlug, einen Standard, der die populärsten und durchdachtesten Erweiterungen von Netscape einschloß, erhielt W3C wieder Bedeutung.

W3C versucht nun, durch die Einführung von *Style-Sheets* HTML wieder auf Kurs zu bringen. Das ist eine Erweiterung von HTML, die ohne doppelte Arbeit die technischen Vorteile von Sinnkodierungen und erweiterter visuell-typografischer Kontrolle miteinander zu verbinden versucht. Style Sheets werden aber noch nicht allgemein unterstützt.

Text und Zeichen

HTML unterstützt alle Zeichen im ISO-8859-1-Zeichensatz, der alle Buchstaben der westeuropäischen Sprachen enthält, also unter anderem *Ñ, é, ÿ, Ç*, und, nicht zu vergessen, ä, ö und ü.

Leider gibt es keine sichere Kodierung für einen Gedankenstrich – man muß den normalen Trennstrich benutzen. Typographische Anführungszeichen werden ebenfalls nicht unterstützt – also benutzt man das Zollzeichen " oder französische Anführungszeichen (» «).

Mathematische Symbole oder Zeichen anderer Alphabete lassen sich in HTML nicht unmittelbar benutzen.

Zeilenwechsel

Einen festen Zeilenwechsel ohne Absatzwechsel erzielst Du mit *Umschalt+Return*. Ein solcher Zeilenwechsel führt im Unterschied zum Absatzwechsel keine Leerzeile mit sich, und der Text vor und nach dem Zeilenwechsel gehört zum selben Absatz. Zeilenwechsel zwischen Überschrift und Textkörper ist zum Beispiel nicht möglich, da das verschiedene Absatztypen sind.

Übrigens hat *Netscape Composer* eine Besonderheit: Die *Return*-Taste setzt einen Zeilenwechsel statt eines Absatzwechsels bei Körpertext-Absätzen, aber einen Absatzwechsel beim Wechsel zu einem anderen Absatztyp.

Leertaste

In HTML läßt sich die Leertaste nicht für die Definition des Textabstandes benutzen. HTML akzeptiert Zwischenräume nur als Trennung von Wörtern gegeneinander: mehrere Leerstellen nacheinander werden als eine einzige betrachtet, und Leerstellen vor und nach einem Absatz, also vor und nach seinem Text, sowie leere Absätze werden ignoriert.

Willst Du dennoch Zwischenräume typographisch benutzen, etwa wenn Du die erste Zeile eines Absatzes einrücken willst, machst Du das mit dem Sonderzeichen *nonbreaking space*, also über *Umschalt + Leertaste*. *FrontPage* und *Claris Home Page* unterstützen leider keinen *nonbreaking space*, also mußt Du ihn in diesen Programmen mit HTML einsetzen, siehe Seite 29. Tabulatoren werden in HTML nicht unterstützt.

Übungen

Nun solltest Du in der Lage sein, ein normales Textdokument zu schreiben. Um Dich mit dem Editor

vertraut zu machen, solltest Du eine Seite mit Überschrift und Textkörper, ein paar Listen und eingezogenen und zentrierten Text schreiben, möglichst mit ein paar kursivierten oder fetten Wörtern und mit verschiedenen Schriftgrößen und -farben. Schreibe ein paar mehrzeilige Textabsätze und sieh Dir an, wie sie je nach Fenstergröße umbrochen werden.

Elemente und Eigenschaften

Alle Elemente einer Webseite haben bestimmte *Eigenschaften* – ein Bild hat eine Größe, eine Seite eine Hintergrundfarbe und so weiter. Jeder Editor hat seine eigene Methode, um diese Eigenschaften der Elemente darzustellen und zu modifizieren. Hier zeigen die Editoren sozusagen Persönlichkeit.

FrontPage und *Netscape Composer* haben etliche Dialogfelder für die Eigenschaften der verschiedenen Elemente, während *PageMill* und *Claris HomePage* mit einem 'schwebenden' Feld arbeiten, dessen Inhalt je nach den Eigenschaften des gewählten Elements wechselt.

FrontPage

Wenn Du in *FrontPage* mit der *rechten Maustaste* ein Element auf der Seite anklickst, erscheint ein kleines Menü, in dem Du die Properties oder Eigenschaften dieses Elements definierst. Normalerweise hast Du die Wahl zwischen mehreren. Page Properties sind Eigenschaften der ganzen Seite, Font Properties beziehen sich auf die Schrift (gegebenenfalls im markierten Bereich) und Paragraph Properties auf den gesamten aktuellen Absatz. Klickst Du ein Bild an, erscheinen die Image Properties , klickst Du auf eine Zelle in einer Tabelle, werden die Table Properties und Cell Properties aktiviert und so weiter. Wählst Du einen dieser Eigenschafts-Punkte (Properties), erscheint ein Dialogfeld, in dem Du die Eigenschaften des gewählten Elements sehen und modifizieren kannst.

Die Eigenschaften für das aktuelle Element – Bild, Formularfeld und so weiter – findest Du am Ende des Edit(Bearbeiten)-Menüs. Die Properties für Tabellen und Tabellenzellen findest Du im Table(Tabelle)-Menü.

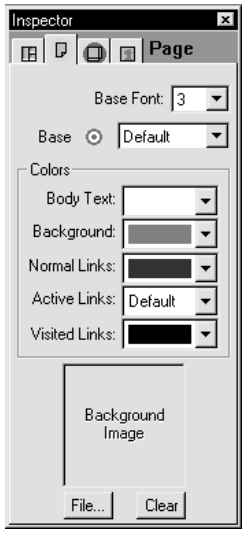
Netscape Composer

Klickst Du im *Netscape Composer* mit der rechten Maustaste die Seite oder ein Element auf ihr an, erscheint ein kleines Menü, in dem Du die Properties des aktuellen Elements festlegst. Im allgemeinen gibt es mehrere Wahlmöglichkeiten. Page Properties bezeichnen die Eigenschaften der gesamten Seite. Character Properties gelten für die Schrift (gegebenenfalls im markierten Bereich), Paragraph/List Properties für ganze Absätze.

Klickst Du ein Bild an, sind die Image Properties zugänglich, in einer Tabelle erscheinen die Table Properties und so weiter. Wählst Du einen dieser Properties-Punkte, erscheint ein Dialogfeld, in dem Du die Eigenschaften des gewählten Elements sehen und modifizieren kannst.

Diese *Properties*-Wahlmöglichkeiten der *rechten Maustaste* sind außerdem auch über das Menü Format zugänglich.

PageMill



PageMill hat ein 'schwebendes Fenster' namens Inspector. Ist dieses Fenster nicht sichtbar, aktivierst Du es über View|Show Inspector. Hier kannst Du die Eigenschaften des aktuellen Elements sehen und modifizieren.

Der Inspector hat vier Registerblätter: Frame zeigt die Eigenschaften für den aktuellen Frame, ist also nur aktiv, wenn die Seite ein *Frameset* enthält. Page zeigt die Eigenschaften, die für die gesamte Seite gelten – Hintergrundfarbe und dergleichen. Form bietet die Eigenschaften des aktuellen Formulars an, ist also nur aktiv, wenn die Seite ein Formular enthält. Object endlich bietet die Eigenschaften des aktuellen Objekts an: Grafik, Tabelle, Formularfeld oder dergleichen.

Ist kein Element markiert, öffnet sich der Inspector im Feld Page. Markierst Du eine Grafik, eine Tabelle oder dergleichen, wird in das entsprechende Eigenschaftsfeld gewechselt.

Claris Home Page

Die Eigenschaften eines Elements werden in einem 'schwebenden Feld' namens Object Editor festgelegt. Ist dieses Feld unsichtbar, aktivierst Du es mit Window|Show Object Editor. Der Object Editor hat drei Registerblätter: Basic, das die normalen Eigenschaften enthält, Advanced mit den eher speziellen, und Both, das beide gleichzeitig zur Verfügung stellt – falls man also genügend Platz auf dem Bildschirm hat.

Farbe und Hintergrund

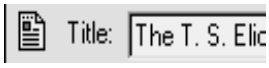
Dies sind Eigenschaften der gesamten Seite. Background bezeichnet die Hintergrundfarbe, Body text die Textfarbe – außer dort, wo Du dem Text eine andere Farbe zuteilst. Normal links ist die Farbe für Links – im allgemeinen Blau, Visited Links bezeichnet die Farbe für Links, die auf bereits besuchte Seiten verweisen – im allgemeinen Lila, und Active Links gibt die Farbe von Links an, die gerade durch Anklick aktiviert sind.

'Default' bedeutet, daß die Farbe im Dokument nicht spezifiziert wird. In diesem Fall erscheint die Seite mit den Farbeinstellungen des benutzten Browsers – normalerweise mit schwarzem Text vor grauem Hintergrund, weil der Browser standardgemäß so eingestellt ist.

Seiteneigenschaften ändern	
Adobe PageMill	Im Inspector, Registerblatt Page.
Netscape Composer	Format|Page Colors and Properties, oder Page Properties
Microsoft FrontPage	File|Page Properties (Datei|Seiteneigenschaften) oder Format|Background (Format|Hintergrund) oder Page Properties(Seiteneigenschaften)
Claris Home Page	Edit|Document Options

PageMill bietet über View|Show Color Panel eine Palette mit 16 Farben an, die Du auf die gewünschten Felder im Inspector oder auf markierten Text ziehen kannst.

Titel



Jedes HTML-Dokument hat einen Titel, der *nicht* mit dem Dateinamen identisch ist. Den Dateinamen sieht nur der Webdesigner, der Titel dagegen ist allgemein zugänglich: er steht in der Titelleiste des Browsers; wird ein Lesezeichen für diese Seite erstellt, erscheint dieser Titel im Lesezeichen-Menü. In *PageMill* gibst Du den Titel eines Dokuments im Feld Title: über dem Fenster an. In den anderen Editoren setzt Du ihn am gleichen Ort ein wie die übrigen Eigenschaften der Seite (siehe oben).

Der Titel kann Zwischenräume und Sonderzeichen enthalten. Die Seite mit der Dateiname `meine_erste_webseite.html` mag den Titel `Meine erste Webseite!` haben. Ein Titel darf nicht mehr als 63 Zeichen enthalten. Achte darauf, daß der Titel informativ ist und für sich stehen kann – er wird nämlich von Suchmaschinen, Indizes und Lesezeichen benutzt. 'Seite 1' oder 'Einführung' sagt wesentlich weniger aus als etwa 'T.S. Eliot-Forum – Inhalt". Die Wörter *Website, Webseite* oder *Homepage* solltest Du in einem Titel vermeiden.

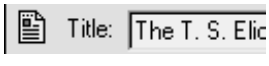
Links

Ein Link erfordert natürlich zwei Dateien: die Seite, von der der Link angerufen wird, und die Seite, die ihn enthält. Die letztere muß mit einem gültigen Dateinamen gespeichert sein. Zunächst markierst Du den Text, der zu einem Link werden soll.

Link zu einer lokalen Datei

Du ziehst die Datei, auf die Du verweisen willst, aus ihrem Ordner oder dem *Windows Explorer* über den markierten Text und läßt los. In *FrontPage* kannst Du eine solche Datei aus dem *FrontPage Explorer* ziehen.

PageMill	File|Place
Composer	Insert|Link, dann Choose File
FrontPage	Insert|Hyperlink (Einfügen|Hyperlink), dann den Knopf Browse (Durchsuchen), zur Dateiwahl
Claris	Insert|Link To File



In PageMill kannst Du außerdem einen Link zu einer weiteren augenblicklich offenen Seite in diesem Programm erstellen, indem Du das Seitensymbol in der Titelleiste der Zielseite auf den Text ziehst, von dem aus verwiesen werden soll, und losläßt.

Links zu anderen Websites

Natürlich kannst Du auch Verknüpfungen zu Seiten einrichten, die sich draußen im Web befinden, also sogenannte externe Links. Das ist am einfachsten, wenn die Zielseite, mit der verbunden werden soll, im Browser offensteht. In diesem Fall ziehst Du schlicht das Link-Symbol, das heißt des kleine Kettenglied neben dem Location(Adresse)-Feld in der Leiste von *Navigator,* auf die Seite im Webeditor und läßt los. Du kannst auch einen Link aus dem Browserfenster in den Editor ziehen. Schließlich kannst Du auch ein Lesezeichen aus dem Lesezeichen-Fenster im *Navigator*, das Du mit Windows|Bookmark (Fenster|Lesezeichen) aktivierst, in den Editor ziehen.

Einen URL von Hand einsetzen

Selbstverständlich kannst Du auch einen Link herstellen, indem Du seinen URL (die sogenannte Internet-Adresse) von Hand eingibst. Vergiß nicht – ein kompletter URL beginnt mit `http://`.

Den URL für ein Link einsetzen	
PageMill	Du klickst im Feld Link to: im unteren Teil des Fensters, schreibst den URL und drückst *Return*
Composer	Insert|Link
FrontPage	Insert|Hyperlink (Einfügen|Hyperlink)
Claris	Insert|Link to URL

Einen Link löschen

PageMill	Markiere den gesamten Text des Links, zum Beispiel indem Du den Link-Text dreifachklickst, lösche den Inhalt im Link to-Feld und drücke *Return*
Composer	Link Properties und den Knopf Remove Link
FrontPage	Hyperlink Properties (Hyperlink-Eigenschaften) und den Knopf Clear (Löschen)
Claris	Erfolgt im Link Editor, den Du mit Window|Show Link Editor aktivierst, über Remove Link

Relative und absolute URLs

Ein Link kann auf zwei verschiedene Arten auf ein anderes Dokument verweisen: Als absoluter URL, der die komplette Adresse des Dokuments enthält, oder als relativer URL, der angibt, wo sich das Dokument im Verhältnis zur Verknüpfungsstelle befindet. Mit einem absoluten URL stellst Du die Verbindung zu einem Dokument auf einem anderen Server her – zum Beispiel `http://www.server.de/hurra.html`. Ein relativer URL verbindet Dokumente auf derselben Website miteinander, zum Beispiel nur `gedichte/eliot.html`. Absolute URLs enthalten „`http:`" und die Serverbezeichnung, relative dagegen nur den Dateinamen und, falls die Dokumente in verschiedenen Ordnern liegen, die entsprechenden Ordnerbezeichnungen. Weiteres zu Webadressen findest Du im Heft *Auf ins World Wide Web* im Abschnitt *Jedes Dokument hat seine Adresse*. Richtest Du Links zwischen Dateien in Deinem Webordner ein, sorgt der Editor automatisch für relative Links.

Ankerpunkte – Links innerhalb einer Website

Ein Link verknüpft normalerweise mit dem Anfang der angesprochenen Seite. Du kannst aber auch einen Link erstellen, der mit einer bestimmten Stelle auf der Seite verknüpft. Das erfordert, daß Du einen sogenannten *Verankerungspunkt* an die Stelle des Textes einsetzt, die angesprochen werden soll. Du setzt den Cursor auf die gewünschte Stelle und gehst dann so vor:

Verankerungspunkt setzen	
PageMill	Edit|Insert Invisible|Anchor
Composer	Insert|Target (*Composer* nennt einen Verankerungspunkt *target*)
FrontPage	Edit|Bookmark (Bearbeiten|Textmarke) (*FrontPage* - nur engl. Version - nennt einen Verankerungspunkt bookmark, in der deutschen Version werden - wie in Word - *Textmarken* gesetzt.)
Claris	Insert|Anchor

Jeder Verankerungspunkt muß einen eigenen Namen haben. Diese Namen folgen denselben Regeln wie Dateinamen: keine Zwischenräume, Sonderzeichen oder dergleichen. Setzt Du einen Verankerungspunkt, generiert der Editor automatisch den jeweiligen Namen. Ist der Punkt markiert, kannst Du den Namen ändern.

Link mit Verankerungspunkt verbinden	
PageMill	Ziehe den Anker auf den markierten Text und laß los.
Composer	Insert|Link, dann Target aus der Liste Select a named target…
FrontPage	Insert|Hyperlink (Einfügen|Hyperlink), Registerblatt Open Pages (Geöffnete Seiten), wähle aus der Liste Bookmark (Textmarke).
Claris	Im Link Editor #Ankername im URL nach dem Dateinamen einsetzen – handelt es sich um einen Punkt im selben Dokument, fällt der Dateiname weg.

E-mail-Link

Vermutlich kennst Du die „Schreib an …"-Links, die den Browser auf Klick zur Öffnung des E-Mail-Fensters mit einem Briefformular mit ausgefülltem Adressenfeld veranlassen. So einen Link erstellst Du schlicht, indem Du den URL so formulierst: "`mailto:`" und dann die E-Mail-Adresse eingibst, zum Beispiel `mailto:beelzebub@unterwelt.de`. Achtung: hier gibt es keinen Schrägstrich vor der Adresse. Üblicherweise ist der Text des Links ebenfalls die E-Mail-Adresse – so sieht der Leser, worum es sich handelt.

Preview Mode (Vorschau-Modus) – einen Link testen

Der Editor hat einen Preview-Modus. In diesem Zustand kannst Du die aktuelle Seite nicht bearbeiten – dafür wird sie aber korrekt dargestellt, und Du kannst Deine Links testen – klickst Du sie an, erscheinen die entsprechenden Seiten (hoffentlich) auf dem Bildschirm (soweit es interne Links sind – externe werden nicht aktiviert), Hilfslinien in Tabellen verschwinden, und Zeilensprungsymbole und andere 'unsichtbare' Zeichen werden verborgen.

Wechseln in den *Preview-Modus*	
PageMill	Du wechselst zwischen *Preview*- und *Edit-Modus* über den großen Schaltknopf rechts in der Symbolleiste.
Composer	File|Browse page
FrontPage	hat keinen *Preview*-Modus, Du kannst Deine Links aber testen, indem Du die *Strg*-Taste hältst und sie anklickst. Hilfslinien und dergleichen aktivierst oder deaktivierst Du über View|Format Marks (Ansicht|Formatierungscode).
Claris	Window|Preview Page

Test im Browser

Ich würde Dir in jedem Fall empfehlen, die Seiten auch in einem Browser zu testen, um zu sehen, wie das Ergebnis ausfällt. Eine HTML-Datei auf der eigenen Festplatte aktivierst Du im Browser über File\|Open File (Datei\|Öffnen bzw. Datei\|Datei im Browser öffnen) oder aber, indem Du sie ins Browserfenster ziehst.

Wechselst Du zwischen Editor und Browser, darfst Du nicht vergessen, daß die Änderungen, die Du im Editor vornimmst, sich unmittelbar nicht im Browser zeigen – Du mußt die Datei zunächst im Editor speichern und sie dann im Browser erneut laden. Das geht beim PC am schnellsten mit der Funktionstaste F5!

Du kannst außerdem ein Dokument, das im Editor offensteht, unmittelbar an den Browser übergeben:

Eine Seite über einen Browser darstellen	
PageMill	View|Switch To (allerdings mußt Du zunächst einmal über Edit|Preferences|Switch To einen oder mehrere Browser definieren)
Composer	File|Browse page
FrontPage	File|Preview in Browser (Datei|Vorschau in Browser)
Claris	File|Preview in Browser

Übung

Erstelle nun zwei Seiten, die durch Links miteinander verknüpft sind. Stelle außerdem einen Link zu `www.yahoo.com` her. Teste diese Links zunächst im Editor und dann im Browser.

Grafiken

Willst Du eine Grafik in Deine Seite integrieren, muß die entsprechende Datei
1) in Deinem Webordner oder einem Unterordner liegen
2) im Grafikformat GIF oder JPEG erstellt sein – hast Du ein Bild in einem anderen Format, mußt Du es in einem Grafikprogramm save as/speichern als GIF oder JPEG. Später – auf Seite 21 – erfährst Du mehr zum Thema Grafik.
3) einen gültigen Dateinamen haben, der die Endung `.gif` oder `.jpg` (bzw. `.jpeg`) trägt – keine Zwischenräume, Sonderzeichen oder dergleichen verwenden.

Wird eine Grafik in eine Webseite integriert, bedeutet das, daß die Bilddatei mit der HTML-Datei verknüpft wird, aber weiterhin wie auch die HTML-Datei als selbständige Datei existiert. Die Bilddatei darf also keineswegs gelöscht werden, sondern muß neben der HTML-Datei vorhanden sein.

Eine Grafik integrieren	
Adobe PageMill	File|Place
Composer	Insert|Image
FrontPage	Insert|Image (Einfügen|Bild)
Claris	Insert|Image

(Achtung! *Netscape Composer* hat die schlechte Angewohnheit, Bilder, die in einem anderen Ordner liegen als das Hauptdokument, in den Ordner des Hauptdokuments zu kopieren. Das bringt viel Durcheinander, kann aber zum Glück deaktiviert werden – Du kreuzt in Image Properties den Punkt Leave image at original location an.)

Du wirst schnell feststellen, daß Grafiken nicht beliebig auf der Seite angebracht werden können. In einem HTML-Dokument wird das Bild in den Text eingefügt und folgt seinem Verlauf. Soll Deine Grafik für sich stehen, muß sie einen eigenen Absatz haben, das heißt, drücke vorher und nachher *Return*. Anschließend kannst Du diesen Absatz rechts- oder linksbündig stellen oder ihn zentrieren.

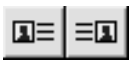
Erscheint das Bild in einer Textzeile, kannst Du definieren, ob es im Verhältnis zum Text hochgestellt, herabgesetzt oder gleichgesetzt werden soll. Das macht vor allem bei kleinen Symbolen Sinn, die als zum Text gehörig erscheinen, etwa ein Logo mit der Aufschrift „neu". Du plazierst die Grafik, indem Du die Eigenschaft **align** auf **top**, **middle** oder **bottom** setzt.

In *PageMill* kannst Du das mit den hier gezeigten Knöpfen in der Symbolleiste erreichen.

Grafiken erhalten die Eigenschaft **alternate text**. Eine solche Textalternative erscheint in einem Textbrowser oder in normalen Browsern bei deaktivierten Bildern. Dieser Text ist wichtig, besonders für Bilder, die Links oder Überschriften sind. Mehr und mehr Leute schalten die Grafik beim Surfen ab, um Zeit zu sparen, und es gibt auch Leute, die nur Text browsen können, weil ihr Computer oder Terminal keine grafische Oberfläche hat, oder weil sie blind sind und der Text verlesen oder in Blindenschrift konvertiert wird.

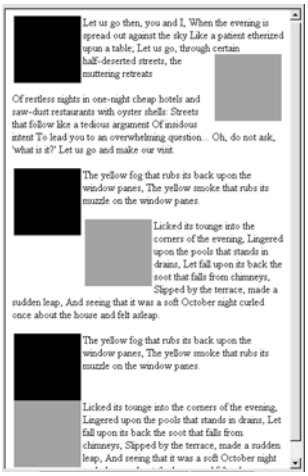
Alternativer Text wird außerdem in Suchmaschinen indiziert, was bei Überschriften, die nur als Grafik erscheinen, natürlich nicht der Fall ist.

Fließtext



Eine Grafik kann an die rechte oder linke Fensterkante gesetzt und vom Text umflossen werden. Das erreichst Du, indem Du die Eigenschaft **align** auf **left** oder **right** setzt. In *PageMill* benutzt Du dazu die beiden Schaltknöpfe in der Symbolleiste.

Nun gleitet die Grafik auf der Zeile, wo sie eingesetzt wird, an den Fensterrand. Es kann vorkommen, daß mehrere aufeinanderfolgende „fließende" Grafiken auf derselben Seite kollidieren. Das vermeidest Du, indem Du über Edit|Insert Invisible|Margin Break (in *FrontPage* Insert|Break, Einfügen|Wechsel) einen *margin break* einsetzt, das heißt eine Unterbrechung, einen Zeilensprung.



Anschließend wird der Text erst unter dem unteren Rand einer rechts- oder linksbündigen Grafik fortgesetzt. Enthält die Zeile mit dem *margin break* keine Grafik, hat er dieselbe Wirkung wie ein normaler Zeilenwechsel.

Eine Grafik, die seitenbündig ist, ist dennoch in den Text integriert. Wird der Text also verschoben oder gelöscht, wird auch die Grafik gelöscht oder verschoben.

Skalieren

Eine Grafik läßt sich unmittelbar im Editor skalieren – Du ziehst einfach an ihrer Ecke. Tu das aber lieber nicht! Die Dateigröße wird dadurch nämlich nicht geändert, und das bedeutet, daß Du ein körniges Bild oder eine viel zu große Bilddatei erhältst. Skaliere sie lieber in einem Grafikprogramm.

Eine Grafik als Link

Eine Grafik kann mit einem Text oder für sich allein als Link funktionieren. Du machst ein Bild zu einem Link, indem Du es markierst und dann den Link genauso erstellst wie bei einem markierten Text.

Eine Grafik erhält einen blauen Rahmen, wenn sie zu einem Link gemacht wird. Dieser Rahmen ist nicht unbedingt schön – Du kannst aber die **border**-Eigenschaft der Grafik auf 0 (Null) setzen und ihn so vermeiden. Allerdings fällt das Bild dann nicht unmittelbar als Link ins Auge. Seine Funktion wird nur deutlich, wenn die Maus darüber geführt wird und der Cursor zu einer Hand wird. Darum sollte eine rahmenlose Grafik als Link ihre Funktion auf irgendeine Weise *signalisieren,* etwa durch ihre Plazierung oder ihr Motiv.

Imagemaps

Eine Grafik kann in verschiedene Bereiche unterteilt werden, die jeweils einen Link beinhalten. Das nennt man eine *Imagemap*.

Eine Imagemap bearbeiten	
PageMill	Du doppelklickst auf die Grafik und benutzt die in der Werkzeugleiste aktive Schaltknöpfe.
Composer	unterstützt Imagemaps nicht
FrontPage	Du klickst das Bild an, worauf eine zusätzliche Image(Bild)-Werkzeugleiste erscheint.
Claris	Im Object Editor wählst Du den Schaltknopf Client-Side Image Map: Edit

Mit diesen Schaltknöpfen kannst Du einen viereckigen, runden oder vieleckigen Bereich der Grafik markieren und anschließend auf normale Weise einen Link herstellen: Bei FrontPage gehst Du jetzt folgendermaßen vor: Wähle die zweite Schaltfläche von rechts, in der deutschen Version Hotspots markieren. Klicke nun auf ein Werkzeug (Rechteck, Kreis etc.) und „male“ auf dem Bild bei gedrückter Maustaste dieses Rechteck. Jetzt öffnet sich automatisch eine Dialogbox, in die Du den Hyperlink für diesen eben definierten Bereich der Grafik eintragen kannst.

Imagemaps haben den Nachteil, daß ihre Links nicht erscheinen, wenn der Leser die Bilder im Browser deaktiviert hat. Also solltest Du außerdem Links erstellen, die als Text-Links zum Bild gehören. Zwar unterstützt HTML 3.2 **alternativen Text** für einzelne Link-Bereiche – der *Netscape Navigator* unterstützt das aber nicht, weswegen Text-Links weiterhin notwendig sind.

Überlege Dir genau, wie Deine Imagemap aussehen soll. Versteht der Benutzer unmittelbar, was die einzelnen Bereiche im Bild bedeuten und wohin die entsprechenden Links führen? Imagemaps sind für den Benutzer oft schwer verständlich.

Wir haben hier eigentlich die sogenannten *Client-side*-Imagemaps beschrieben – es gibt aber auch eine ältere Form, die *Server-side*-Imagemaps, die ein Programm auf dem Server erfordern, um zu funktionieren. Nach dem allgemeinen Erfolg der *Client-side*-Imagemaps sind *Server-side*-Imagemaps inzwischen bedeutungslos.

Hintergrundgrafik

Du kannst Deine Seite sozusagen mit einem Hintergrundbild ‘tapezieren’. Dieses Bild wird vertikal und horizontal bis an den Fensterrand wiederholt. Ist es größer als das Fenster, wird es beschnitten. Dieses Bild muß eine Grafik im Format GIF oder JPEG sein. Vielfach werden Texturen als Hintergrund angewendet.

Eine Hintergrundgrafik wird an der gleichen Stelle eingefügt wie die anderen Eigenschaften einer Seite.

Sei vorsichtig, wenn Du Hintergrundtexturen oder Bilder benutzt. Vor einem solchen Hintergrund wird der Text leicht unlesbar. Die Hintergrundgrafik sollte möglichst selbst kontrastarm sein, aber in großem Kontrast zum Text stehen.

Wird eine Webseite geladen, erscheint sie zunächst einmal nur mit ihrer Hintergrundfarbe – die Hintergrundgrafik erscheint erst nach einer Weile. Du machst den Einlesevorgang angenehmer für das Auge, indem Du der Seite eine Hintergrundfarbe gibst, die der dominierenden Farbe des Hintergrundbildes nahekommt. Ist das Hintergrundbild vollständig eingelesen, verschwindet die Hintergrundfarbe.

Übungen

Erstelle eine Seite mit Hintergrundbild und Vordergrundgrafiken. Hast Du keine Bilder, kannst Du sie im Web leihen: Du findest eine Seite mit Bildern, klickst mit der *rechten Maustaste* ein Bild an und wählst Save Image As… (Bild speichern unter...).

Bist Du fertig, liest Du die Seite in Deinen Browser ein, um ihr Aussehen zu testen. Ändere Deine Monitoreinstellungen, um die Wirkung bei unterschiedlichen Auflösungen (640 x 480, 1024 x 768 und so weiter) und verschiedenen Farbtiefen (256 Farben, 16 Bit (*thousands of colors*), 24 Bit (*millions*) zu testen.

Tabellen

Tabellen wurden eigentlich eingeführt, um Zahlen und Daten in Schemata aufzustellen. Heutzutage werden sie aber vor allem benutzt, um ein spannenderes Layout zu erzielen.

Ohne Tabellen wird das Layout ziemlich eindimensional: die Absätze kommen brav nacheinander und untereinander. Tabellen ermöglichen Blöcke, die nebeneinander liegen.

Eine Tabelle ist ein Rahmen, der in waagerechte und senkrechte Zellen unterteilt ist. Die einzelnen Zellen können Text, Grafik und so weiter enthalten. Du kannst zum Beispiel mit einer Tabelle, die die gesamte Bildschirmbreite füllt und drei Zellenreihen und -spalten enthält, ein dreispaltiges Layout erstellen.

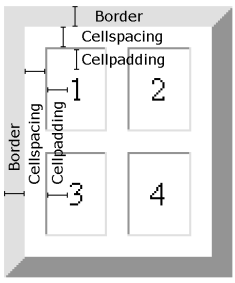
Du setzt eine Tabelle über den entsprechenden Schaltknopf auf der Werkzeugleiste ein, worauf Du nach der Anzahl der Reihen und Spalten gefragt wirst.

Eigenschaften einer Tabelle

Border (Rahmenstärke) gibt mit der Anzahl an Pixel an, wie breit der Rahmen um die Tabelle ist. Setzt Du 0 (Null), ist der Rahmen unsichtbar.

Cellspacing (Textabstand) gibt die Dicke der Balken zwischen den Zellen in Pixel an.

Cellpadding (Zellabstand) gibt den Abstand in Pixel zwischen Inhalt (Text oder Grafik) und Zellenrand an.



Setzt Du **border** auf 0, sind die Balken zwischen den Zellen ebenfalls unsichtbar – der Abstand ist aber der gleiche. Ob das so recht logisch ist, weiß ich nicht – aber so ist es nun einmal. Setzt Du **border**=1 und **cellspacing**=1, sind Rahmen und Balken schwarze Striche. Erhöhst Du den Wert, erscheinen sie dreidimensional, was eigentlich nicht besonders gut aussieht.

Width (Breite) bezeichnet die Breite der Tabelle. Sie wird in Prozent des Bildschirms oder in Pixel angegeben. Wählst Du das letztere, mußt Du natürlich damit rechnen, daß der Benutzer waagerecht scrollen muß, wenn sein Bildschirm

schmaler ist als die Tabelle.
Die geringste allgemein übliche Bildschirmauflösung ist 640 x 480. Erstellst Du also Tabellen mit pixeldefinierter Breite, sollte sie nicht mehr als 600 Pixel betragen. Allerdings füllt eine solche Tabelle bei einem Bildschirm von 1200 Pixel Breite nur ein Drittel!

Height (Höhe) bezeichnet die Höhe der Tabelle – ebenfalls in Prozent oder in Pixel. Die Tabelle wird aber in jedem Falle nach der Höhe des Inhalts formatiert. **Height** ist eine etwas problematische Eigenschaft, da sie nur vom *Netscape Navigator*, nicht aber vom *MSIE* verstanden wird. Also würde ich Dir von einer Definition der **Height**-Eigenschaft einer Tabelle abraten.

Eine Tabelle kann rechts- oder linksbündig oder zentriert auf der Seite angebracht werden. Außerdem kannst Du ohne weiteres eine 'Unter'-Tabelle in eine Zelle integrieren.

Zelle

Auch eine Zelle hat bestimmte Eigenschaften:

Width (Breite) bezeichnet die Breite der Zelle – in Pixel (abzuraten) oder in Prozent der Tabellenbreite. Enthält eine Zelle ein Element, zum Beispiel ein Bild, das breiter als die definierte Breite ist, wird sie in ihrer Breite angepaßt.

Height (Höhe) definiert die Höhe der Zelle in Pixel oder Prozent der Fensterhöhe. Siehe auch **Table height** oben.

Align (Ausrichtung) definiert die horizontale Ausrichtung: rechts- oder linksbündig oder zentriert.

VAlign (**vertical align oder vertikale Ausrichtung**) bestimmt die senkrechte Justierung des Inhalts: oben, zentriert oder unten.

Cellcolor (Farbe, Hintergrundfarbe): Jede Zelle kann eine eigene Hintergrundfarbe haben. Das wird allerdings nur von den neuesten Browsern unterstützt, also *Netscape Navigator 3.0* und *MSIE 3.0* – Du solltest also für eine Farbkombination sorgen, die den Text sowohl vor dem Zellenhintergrund als auch vor dem Seitenhintergrund lesbar macht.

In PageMill aktivierst Du eine Zelle, indem Du zunächst auf den Rand der Tabelle doppelklickst, so daß sie aktiv wird, und anschließend in der gewünschten Zelle klickst.

Du kannst Zellen waagerecht oder senkrecht miteinander verschmelzen (merge). Das ist praktisch bei Überschriften, die sich über mehrere Zellen erstrecken. Hierzu setzt Du die Zelleneigenschaften **Col Span** und **Row Span** auf die Anzahl an Spalten/Reihen, die miteinander verschmolzen werden – und zwar in der oberen bzw. linken der aktuellen Zellen:

Zellen verschmelzen	
PageMill	Du wählst eine Zelle, drückst *Umschalt* und wählst die nächste Zelle. Dann wählst Du Join cells in der Werkzeugleiste.
Composer	Table Properties, Registerblatt Cell: Du setzt die Cell spans row(s) and column(s)-Eigenschaften.
FrontPage	Cell Properties (Zelleneigenschaften): Du richtest die Spannweite der Zelle ein über Numbers of rows/columns spanned (Spannweite der Zelle).
Claris	Im Object Editor, Registerblatt Advanced: Die **Col Span-** bzw. **Row Span**-Eigenschaften der Zelle einstellen.

Übungen

Erstelle zwei Tabellen – eine mit fester Breite in Pixel und eine mit prozentualer Breite. Fülle ihre Zellen mit Text und Grafik. Lese diese Seiten in Deinen Browser und teste, wie sie auf verschiedene Fenstergrößen und Bildschirmauflösungen reagieren.

Test im Browser

Du solltest Deine Website unbedingt testen, bevor Du sie zum Server überführst.

- Teste sie sowohl im *Netscape Navigator* als auch im *MSIE*.
- Teste sie ohne Grafik – im Navigator wählst Du dazu Options|Auto Load Images Off (Optionen|Grafiken automatisch laden). Hast Du - wie empfohlen - für alle sinntragenden Grafiken und Text-Links Alternativtexte erstellt - beispielsweise für die „Hotspots“ in eventuellen Imagemaps - sollte die Website immer noch funktionieren.

Übung

Erstelle eine Website mit Farben, Grafik und Tabellen. Teste die Seiten in beiden Browsern und stelle fest, ob es Unterschiede in der Darstellung gibt.

Veröffentlichung

Serverplatz

Ein *Webhotel* bietet Platz auf einer Festplatte in einem sogenannten *WebServer* an, der permanent ans WWW angeschlossen ist. Als Kunde mietest Du ein Verzeichnis, einen Ordner, für Deine Webdokumente auf dieser Festplatte.

Viele Internet-Anbieter (T-Online, Compuserve, AOL etc.) stellen außer dem Internet-Zugang und einer E-Mail-Adresse einen gewissen Platz auf dem Server zur Verfügung. Abgesehen davon gibt es Webhotels in jeder Preisklasse – je nach Bedarf: 1) für private bzw. kommerzielle Benutzer, 2) nach Deinem Platzbedarf, 3) nach der ‘Abruf-Frequenz’ Deiner Website, 4) falls Du Zugang zu CGI-Programmen benötigst (über CGI erfährst Du mehr in den entsprechenden Abschnitten ab Seite 47), 5) nach Servicebedarf und so weiter. Unter `www.geocities.com` besteht die Möglichkeit einer kostenlosen Homepage.

Es gibt ziemliche Unterschiede zwischen der Bandbreite von Webhotels. Der Anbieter spart unter Umständen Geld, indem er eine größere Anzahl an Homepages zuläßt, als die Kapazität seines Servers eigentlich erlaubt. Du kannst probeweise ein paar Seiten im Webhotel aktivieren, um zu testen, wie schnell sie eingelesen werden.

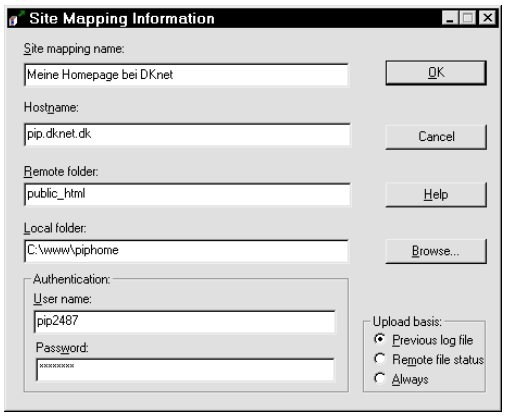
Upload

Du ‘uploadest’, das heißt überführst Deine Website, indem Du den Inhalt des Webordners von Deiner eigenen Festplatte in Deinen Ordner auf dem Webserver überführst. Mehrere Editoren verfügen über eine integrierte Upload-Funktion. Andernfalls benutzt Du ein sogenanntes *FTP-Programm*, das Dateien auf Deinem eigenen Computer und einem Webserver verwalten, also sie auch von dem einen zum anderen Computer überführen kann.

Die vier hier besprochenen Editoren haben alle eine Upload-Funktion. Allerdings ist ihre Qualität unterschiedlich – *PageMill* und *FrontPage* arbeiten sehr gut, während *Composer* und *Claris Home Page* in diesem Punkt Schwächen zeigen.

Benutzt Du eines der letzteren Programme, empfiehlt sich ein spezielles FTP-Programm. PageMill hat in der Macintosh-Version merkwürdigerweise keine Upload-Funktion, weswegen Du hier ebenfalls ein spezielles FTP-Programm benutzen mußt. Auch wenn Dein Editor eine integrierte Upload-Funktion hat, empfiehlt sich ein FTP-Programm, um Deine Dateien im Server aufzuräumen. Manche Dienstleister bzw. Webhotels bieten solche Programme kostenlos an, so können Mitglieder von CompuServe ihre Website mit dem HomePage *Publishing Wizard* veröffentlichen.

Notwendige Informationen für das Uploaden



Willst Du uploaden, benötigst Du folgende Informationen von Deinem Webhotel-Anbieter: **Username**. Im allgemeinen bestimmst Du diesen selbst. Oft lautet er so wie Deine E-Mail-Adresse – also `sofie`, wenn Deine E-Mail Adresse `sofie@hyperweb.de` lautet.

Paßwort. Der geheime Code, der sichert, daß nur Du Deine Webseiten modifizieren kannst.
Den Namen des Servers, zu dem Du einen FTP-Zugang benötigst, zum Beispiel `ftp.hyperweb.de`. Der Servername für FTP ist nicht unbedingt derselbe wie der Servername der Webadresse – zum Beispiel kann er mit `ftp` beginnen statt mit `www`.
Den Namen des Ordners, in dem Deine Website gespeichert wird. Dieser Ordner liegt in Deinem Benutzerordner und heißt meist `public_html`. Nur Du hast Zugang zu diesem Benutzerordner, da er etliche private Dinge enthält, wie etwa Deinen Briefkasten. Der Ordner `public_html` dagegen kann, wie sein Name andeutet, die Dokumente enthalten, die über das Web öffentlich zugänglich sind. Selbstverständlich kann dieser Ordner weitere Ordner enthalten.
Die **Webadresse** Deines Ordners. Das wäre zum Beispiel `www.webastoria.de/~sofie/`. Webadressen starten nicht unbedingt mit `www`, obwohl das meistens der Fall ist.

Die Tilde ~ wird vielfach in Webadressen benutzt. Sie deutet an, daß der Ordner einem Benutzer auf dem Server gehört. Eigentlich weiß man nicht, wo in der internen Dateistruktur des Servers sich der Benutzerordner befindet – das macht aber nichts. Der Benutzername verbindet den Benutzer automatisch mit dem richtigen Ordner. Übrigens brauchst Du nicht zu befürchten, Du könntest die Dateien anderer Leute oder die Festplatte des Servers löschen: Benutzername und Paßwort erlauben Dir nur den Zugang zu Deinem eigenen Benutzerordner.

Uploaden mit PageMill

Du wählst File|Upload Page. Im Upload-Dialog erstellst Du ein 'Site Mapping', das die Upload-Spezifikationen für eine Website enthält. (Site-Mapping bedeutet Zuordnen der Site.) Arbeitest Du mit mehreren Websites, kannst Du für jede ein eigenes Site Mapping erstellen. Erstellst oder modifizierst Du ein Site Mapping, gibst Du sowohl Deinen lokalen Webordner an als auch die Upload-Adresse einschließlich Paßwort und so weiter.

Außerdem wählst Du Upload Basis und legst fest, ob der gesamte Inhalt des Webordners, bestimmte Dateien oder kürzlich geänderte Dateien überführt werden sollen. Schließlich wählst Du Upload, wonach die Dateien nach den von Dir gemachten Angaben überführt werden.

Uploaden mit FrontPage

In FrontPage heißt das Hilfsprogramm *Microsoft Web Publishing Wizard* – Du startest es über den Menüpunkt File|Publish FrontPage Web (Datei|FrontPage Web publizieren). Wenn Du zum ersten Mal 'veröffentlichst', leitet Dich der Wizard durch die notwendigen Konfigurationen. Benutzt Du das Programm später erneut, mußt Du nur noch wählen, was Du überführen willst: 1) alles, 2) ausgewählte Dateien oder 3) nur seit der letzten Überführung geänderte Dateien.

Uploaden mit Composer

Der Composer kann leider nur Dateien jeweils *eines* Ordners überführen, nicht aber die Dateien eventueller Unterordner. Willst Du die aktuelle Seite uploaden, wählst Du File|Publish (Datei|Veröffentlichen). Gewählt wird automatisch die Seite, die im Editor offensteht. Außerdem kannst Du andere Seiten in demselben Ordner wählen, indem Du All files in page's folder (Alle Dateien im Dokumentenordner) anklickst und die zu überführenden Seiten markierst.

Uploaden mit Claris Home Page

Claris Home Pages integrierter Uploader kann nur jeweils eine Seite einschließlich Grafik überführen – hier empfiehlt sich also wirklich ein FTP-Programm. Willst Du die aktuelle Seite uploaden, wählst Du File|Remote|Remote.

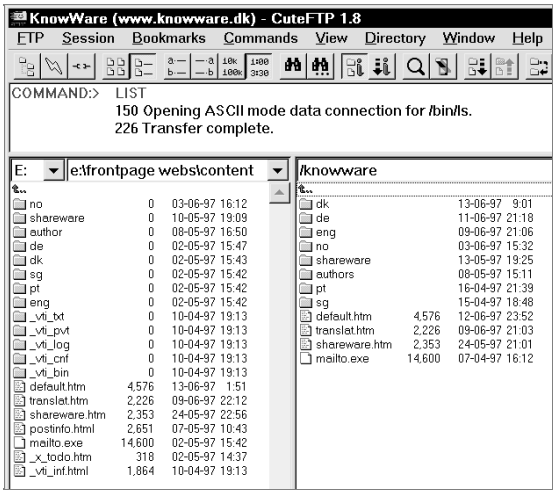
Upload mit einem FTP-Programm

Gute FTP-Programme sind zum Beispiel:

CuteFTP für Windows `www.cuteftp.com/`
Fetch für Mac
`www.dartmouth.edu/pages/softdev/fetch.html`

Wenn Du ein solches Programm startest, richtest Du zunächst eine 'Connection' (Verbindung) ein. Du solltest ihr einen Namen geben, damit Du sie von anderen Verbindungen unterscheidest. Das könnte zum Beispiel 'Meine Website bei WebHilton' sein.

Du wählst nun die bereits angeführten notwendigen Angaben für die FTP-Verbindung. Das FTP-Programm richtet sie ein, und nun erscheint ein Dateistruktur-Fenster mit dem Inhalt Deines Ordners. Nun kannst Du von Hand den Inhalt Deines lokalen Webordners in Deinen Webordner auf dem Server ziehen.



Manche FTP-Programme wollen wissen, ob die Dateien als Text (ASCII) oder binär (auch *Raw Data* genannt) gespeichert werden sollen. Grafik und andere Daten müssen binär überführt werden. HTML dagegen funktioniert unter beiden Formen. Bist Du Dir nicht sicher, wählst Du binär. Wähle *keinesfalls* merkwürdige Formate wie *uuencoding* oder *MacBinary*.

Manche FTP-Programme bieten außerdem an, ein `.bin` oder `.txt` nach dem Dateinamen hinzuzufügen. Das lehnst Du natürlich ab. *Fetch* zum Beispiel möchte Deine Dateien gern als MacBinary kodieren und ihrem Namen ein `.bin` hinzufügen. Das deaktivierst Du in den Preferences. Übrigens erlaubt Dir das FTP-Programm auch die Dateienverwaltung Deiner Benutzermappe auf dem Server.

Kontrolle

Wenn Du mit dem Überführen fertig bist, solltest Du im Browser, also unter der WWW-Adresse und nicht unter der FTP-Adresse, sicherheitshalber testen, ob alles wie geplant funktioniert. Fehlersymptome:

Statt Deiner Eingangsseite erscheint ein Überblick über die Dateien im Webordner. Sieh nach, ob die Eingangsseite den rechten Namen hat – `index.html, default.html` oder was immer Dein Webserver fordert. Vergiß nicht den Unterschied zwischen Groß- und Kleinbuchstaben und den zwischen `.htm` und `.html`.

Links funktionieren nicht, und statt Bildern erscheint ein Grafiksymbol mit einem Fragezeichen. Sieh nach, ob die Dateien überführt wurden, ob die Namen unverändert blieben und ob die Links relativ sind und keine Hinweise auf Deine eigene Festplatte enthalten. Stelle außerdem sicher, daß es keine Links zu Dateien oder Ordnern gibt, die nicht überführt wurden – zum Beispiel indem Du irrtümlich einen Link zu einem Dokument erstellt hast, das auf Deiner eigenen Festplatte nicht im Webordner, sondern in einem anderen Ordner liegt.

Statt der Bilder erscheint ein zerbrochenes Grafiksymbol. Die Bilder wurden nicht korrekt überführt. Sichere Dich, daß sie binär und nicht etwa als Text überführt wurden.

Sind die Dateien Deines Webordners überführt und werden von Deinem Browser über die Webadresse korrekt überführt (nicht vergessen: RELOAD!), ist Dein Website online.

Zweiter Teil – Alles über Webdesign

Inzwischen solltest Du imstande sein, eine einfache Website zu erstellen und zu überführen. Wir wollen uns nun die verschiedenen Aspekte im Webdesign genauer ansehen. Es ist durchaus möglich, die folgenden Abschnitte in beliebiger Reihenfolge zu lesen.

Grafikbehandlung

Grafikformate

GIF und JPEG sind die Grafikformate, die im Web benutzt werden. Der Grund: sie komprimieren effektiv, reduzieren also die Dateiengröße anderer typischer Grafikformate. Jedes dieser beiden Formate hat seine Vorteile:

GIF eignet sich vor allem für Zeichnungen, Logos und andere Bilder mit wenigen Farben, monochrome, also einfarbige Bilder und einfache Formen. Leider enthalten GIF-Bilder höchstens 256 Farben. Ein GIF-Bild kann transparente Bereiche enthalten. Das ist nützlich bei irregulären Bildern, die vor verschiedenen Hintergrundfarben oder Hintergrundbildern erscheinen.

JPEG eignet sich im Gegensatz zu GIF vor allem für Fotografien, Malereien und andere Bilder mit zahlreichen Nuancen und gleitenden Farbübergängen. JPEG-Grafiken stehen im 24-Bit-Modus, enthalten also Millionen von Farben. Andererseits ist ihre Komprimierung nicht echt – konvertiert man ein Bild zu JPEG, geht etwas von seiner Qualität verloren. Die JPEG-Komprimierung ist äußerst stark und ergibt ziemlich dramatische Reduzierungen in der Dateigröße.

Grafik von anderen Quellen

Bilder, die Du von anderen erhältst, stehen normalerweise im TIFF-Format. Vielleicht bietet der Lieferant Dir eine Konvertierung zu JPEG oder GIF an. Das solltest Du dankend ablehnen. Es ist besser, wenn Du das selbst tust. Eine Grafik sollte nämlich erst *nach* der Skalierung in die für die Website gewünschte Größe konvertiert werden. Im allgemeinen empfiehlt sich das TIFF-Format für Bilderaustausch. Bestelle das Bild lieber zu groß als zu klein – Du kannst es jederzeit verkleinern, vergrößern kannst Du es aber nicht, ohne daß die Qualität darunter leidet.

Scannen

Du solltest das Bild größer scannen als Du es benötigst – dann kannst Du es im Grafikprogramm skalieren. Gib beim Scannen an, wie viele DPI Du wünschst. DPI steht für *dots (Pixel) per inch*. Scannst Du also ein Bild von 5x5 Zoll (inch) in 150 DPI, erhältst Du eine Grafikdatei von 750x750 Pixel.

Text, Illustrationen und Anti-Alias

Importierst Du Illustrationen aus einem Zeichenprogramm wie etwa *Illustrator* oder erstellst Du Text in einer Bildverarbeitung, vergiß nicht, die Kanten mit *Anti-Alias* zu behandeln. Das ist eine Technik, die unregelmäßige Kanten mit Hilfe von Farbtönen ausgleicht. Dadurch sieht ein Bild natürlicher aus und vermittelt die Illusion einer besseren Bildschirmauflösung. Allerdings kann Text in kleinerer Schriftgröße durch Anti-Aliasing schwerer lesbar werden.

Transparenz

Es ist möglich, Bereiche in einem GIF-Bild transparent zu machen. Dazu gibst Du allen gewünschten Bereichen im Bild dieselbe Farbe und definierst diese Farbe beim Speichern als GIF-transparent. Leider kann man keine graduiert-durchsichtigen Pixel herstellen – entweder sie sind durchsichtig oder nicht. Du kannst die Kanten eines Bildes vor durchsichtigem Hintergrund also nicht ‘faden’ (abschwächen) oder mit Anti-Alias behandeln. Du erreichst diesen Effekt nur, indem Du das in der Farbe der Grafikdatei machst und dafür sorgst, daß der Seitenhintergrund dieselbe Farbe hat.

Skalierung

Skaliere Deine Bilder auf die richtige Größe. Willst Du sie retuschieren, solltest Du das vor dem Skalieren tun. Vergiß nicht – was zählt, ist die Größe der Bilder in Pixel. Die Größe in Zoll oder Zentimeter spielt nur für Bilder eine Rolle, die gedruckt werden.

Ich empfehle Dir, *vor* dem Skalieren eine Sicherheitskopie der Grafikdatei zu erstellen. Willst Du sie später modifizieren oder skalieren, kannst Du das in der Backupdatei tun und dann diese konvertieren.

Konvertierung zu JPEG

Bevor Du eine Grafik in JPEG konvertierst, stelle fest, ob sie im RGB-Modus steht. Bilder, die Du von anderen erhältst, sind vielleicht in einem Format namens CMYK gespeichert, einem Format, das für Bilder benutzt wird, die gedruckt werden sollen. Ist das der Fall, mußt Du die Datei nach RGB konvertieren.

Die Konvertierung erfolgt, wenn Du Save As (Sichern als bzw. Speichern unter): JPEG wählst. Du entscheidest selbst, wieviel Qualität Du opfern willst – von ‘sehr wenig’ bis ‘ziemlich viel’. Je größer der Qualitätsverlust, desto kleiner die Dateigröße. Es empfiehlt sich, die Datei in mehreren Qualitäten zu speichern, um sie auf Qualität und Dateigröße miteinander zu vergleichen. Der Qualitätsverlust ist selten sichtbar, allerdings werden größere gleichfarbige Flä-

chen leicht verfärbt, und Kanten verlieren ihre Schärfe. Je weniger Details ein Bild enthält, desto mehr kann man opfern. Ein Bild eines Sonnenuntergangs läßt sich stark komprimieren, ein Portrait dagegen weniger stark.

Achtung: Eigentlich heißt das JPEG-Dateiformat *JFIF* – JPEG ist nämlich nicht der Name des Dateiformats, sondern der der Kompressionsmethode. Normalerweise benutzt man dennoch JPEG, manchmal nennt man das Format aber auch *JPEG/JFIF*.

Progressive JPEG

Es gibt eine Variante von JPEG namens *progressive JPEG*. In einer solchen Datei liegen die Bildinformationen in einer anderen Reihenfolge, die dazu führt, daß das Bild beim Einlesen zunächst in einer unscharfen Version erscheint, um dann allmählich schärfer zu werden. Die Datei nimmt ebensoviel Raum ein wie eine übliche JPEG-Datei und benötigt ebensoviel Zeit zum Laden, der Leser erhält aber einen früheren Eindruck des Bildes. Nicht alle Programme können als progressive JPEG speichern. *Photoshop 3.x* erfordert hierzu ein Plug-In, zum Beispiel *Pro JPEG*.

Achtung! *PageMill* unterstützt kein Progressive JPEG – Du kannst progressive Bilder zwar mit place einfügen, PageMill zeigt sie aber nicht.

Konvertierung zu GIF

Soll ein Bild in das GIF-Format konvertiert werden, muß seine Farbtiefe auf „8 Bit" reduziert werden, also auf maximal 256 Farben.

In Paint Shop Pro erreichst Du das über Colors|Decrease Color Depth|256 Colors (8 Bit) (Farben|Farbtiefe verringern|256 Farben). (In Photoshop: Image|Mode|Indexed Color.)

Anschließend mußt Du festlegen, wie die Farbtiefe reduziert wird. In Paint Shop Pro wählst Du Palette: Optimized (Palette: Optimierung) (in Photoshop: Palette: Adaptive), um eine Palette zu generieren, die die 256 wichtigsten Farben des Bildes enthält. Enthält das Bild mehr als 256 Farben, mußt Du außerdem entscheiden, was mit den Farben geschieht, für die in der Palette kein Raum ist:

Reduction Method: Nearest Color (Reduktionsmethode: nächstliegende Farbe) (Dither: None in Photoshop) bedeutet, daß nicht in der Palette enthaltene Farben auf die nächstgelegene Farbe der Palette gesetzt werden.

Reduction Method: Error Diffusion (Reduktionsmethode: Fehlerstreuung) (Dither: Diffusion in Photoshop) bedeutet, daß das Programm die fehlende Farbe durch ein Muster von „Körnern" in nahegelegenen Farben der Palette nachzuahmen versucht – das Bild wird also gepunktet.

Es hängt ganz von der Art des Bildes ab, welche Methode sich jeweils empfiehlt. Diffusion bewahrt die Farben genauer, dafür wird das Bild aber körnig. Probiere die Sache mit mehreren Bildern aus, um den Unterschied festzustellen.

Enthält das Bild ohnehin nur 256 oder weniger Farben, treten derartige Probleme natürlich gar nicht erst auf. Photoshop deutet das mit der Wahlmöglichkeit palette: exact an, die bedeutet, daß alle Farben des Bildes bewahrt werden. Die Dither-Möglichkeit ist deaktiviert, da es ja keine „fehlenden" Farben gibt.

In Paint Shop Pro teilt Dir der Dialog nicht mit, ob das Bild mehr bzw. weniger als die 256 möglichen Farben enthält. Bevor Du aber die Farbtiefe reduzierst, kannst Du den Punkt Colors|Count Colors Used (Farben|Farbenanzahl) wählen und so feststellen, ob das Bild mehr als 256 Farben enthält. Ist das nicht der Fall, ist die Reduction Method (Reduktionsmethode) irrelevant.

Die Möglichkeit der Farbreduzierung in einem Bild birgt eine Gefahr: die meistbenutzten Farben werden verstärkt, während nur geringfügig vorkommende Farben des Bildes möglicherweise völlig verschwinden. Da diese Kontrastfarben aber vielfach dem Bild seinen eigentlichen Charakter geben, solltest Du sie vielleicht von Hand verstärken.

Hat das Bild eine Farbtiefe von 8 Bit, kannst Du es über File|Save As als GIF speichern.

Interlaced GIF

GIF-Bilder lassen sich auch auf spezielle Weise als *interlaced* speichern. Das bedeutet, daß die Informationen der Grafikdatei in einer anderen Reihenfolge gespeichert werden – das Bild wird nun nicht von oben an bis unten eingelesen, sondern erscheint zunächst in grober Auflösung, um dann allmählich weitere Einzelheiten zu zeigen. Ein „interlaced" Bild nimmt den selben Raum ein und braucht dieselbe Zeit, um eingelesen zu werden.

In Photoshop bestimmst Du im Dialogfeld des Befehls Save As… GIF, ob die Datei interlaced gespeichert werden soll.

In Paint Shop Pro erhältst Du im Dialog Save As (Sichern als) eine Wahlmöglichkeit namens Sub type (Sub-Format). Hier hast Du die Wahl zwischen den GIF-Versionen 87a und 89a, interlaced oder nicht-interlaced. Wähle 89a, die modernste Form des GIF-Formats.

256-Farben-Bildschirm und die 'sichere Webpalette'

Manche Computer können nur 256 Farben auf dem Bildschirm darstellen. Auf diesen Maschinen benutzt der Browser eine feste Palette mit 216 Farben. Diese Palette ist bei allen Browsern und auf allen 256-Farben-Plattformen dieselbe. Daß sie nur 216 Farben enthält, liegt daran, daß einige Farben vom System für Fenster, Schaltknöpfe und so weiter reserviert werden.

Die Farben dieser 216-Farben-Palette werden korrekt wiedergegeben, während die anderen Farben vom Computer durch *Dithern* oder Änderung der Farbe in eine Palettenfarbe erstellt werden.

Oft möchte man solche unvorhersehbaren Änderungen vermeiden, so etwa in Symbolen, Logos, Überschriften, die als Grafik erstellt wurden, anderen Motiven, die aus reinen Farben bestehen, oder größeren einfarbigen Flächen. Stell Dir etwa vor, daß sich ein einfarbiges Logo auf einem 256-Farben-Bildschirm in Nadelstreifen oder kleinkarierte Flekken verwandelt. Willst Du das vermeiden, mußt Du Deine Bilder so modifizieren, daß sie nur Farben der 216-Farben-Palette enthalten.

Neuere Bildverarbeitungen wie *Photoshop 4* haben die 216-Farben-Palette, die sogenannte 'sichere Web-Palette', integriert. In Photoshop 4 kannst Du zum Beispiel Palette: Web wählen, um die Farbtiefe zu reduzieren.

In Paint Shop Pro ist die Sache komplizierter, weil Du zunächst einmal die Palette erstellen mußt. Das machst Du so:

- Du gehst zu einer Website, der diese Palette anbietet, zum Beispiel `www.dsiegel.com/tips/wonk10/images.html`
- Downloade das Bild der Palette, das heißt rechtsklicke es und wähle Save Image As... (Bild speichern unter...).
- Öffne das Bild in Deinem Grafikprogramm
- Wähle Colors|Save Palette (Farben|Sichere Palette), um die Palette zum Beispiel unter dem Namen Webpalette zu speichern

Möchtest Du nun in Deinem Grafikprogramm ein Bild an die Webpalette anpassen, wählst Du Colors|Load Palette (Farben|Lade Palette) und aktivierst die Webpalette.

In jedem Fall mußt Du entscheiden, was mit den Farben geschehen soll, die nicht in der Webpalette enthalten sind. Wähle Nearest Color (nächstliegende Farbe) (in Photoshop: Dither: None) – geht es doch eben darum, das Dithern zu vermeiden. Das Bild wird nun an die 'sichere Palette' angepaßt, worauf Du es als GIF speichern kannst.

Nuancen, die nur auf besseren Bildschirmen wiedergegeben werden, gehen verloren, wenn Du das Bild an die strenge 216-Farben-Palette anpaßt. Darum solltest Du sie nur verwenden, wenn Du unbedingt Dithern vermeiden mußt. Bilder mit vielen Nuancen und Farbtönen – wie etwa Malereien oder Fotografien – dürfen nicht an die Webpalette angepaßt werden, sondern sollten statt dessen als JPEG gespeichert werden. Bist Du Dir nicht sicher, ob ein Bild auf einem 256-Farben-Bildschirm ordentlich aussieht, kannst Deine Bildauflösung auf 8 Bit (256 Farben) reduzieren.

Übrigens gibt es auch einen Unterschied in der 'Lichtstärke' der verschiedenen Plattformen - auf einem Macintosh sieht ein Bild im allgemeinen heller aus als auf einem PC. Erstellst Du z.B. ein Bild auf einem Macintosh, darf es nicht zu dunkel ausfallen, da es auf einem PC noch dunkler wiedergegeben wird.

Bildbehandlung in PageMill

PageMill enthält einen simplen Bildeditor, der aktiviert wird, wenn Du eine Grafikdatei über File|Open öffnest. Hier kannst Du die Datei *interlaced* machen und ausgewählte Bereiche als *transparent* setzen. Das ist angenehm und ebenso leicht durchführbar wie in einem Grafikprogramm. Du kannst auch die Bildgröße skalieren, und das ist gar nicht gut! Der Editor ändert nämlich nicht die eigentliche Datei, sondern nur die Maße, die im Dokument angegeben werden, und das bedeutet, daß das Bild entweder zu körnig wird oder die Datei größer als notwendig und dadurch beim Laden mehr Zeit beansprucht als nötig.

URLs

URL ist die technische Bezeichnung für eine Internetadresse. Als normaler Surfer hat man nur bei der Angabe einer Adresse mit einem solchen URL zu tun. Tatsächlich enthält ein HTML-Dokument aber intern jedes Mal den entsprechenden URL, wenn ein Link, eine Grafikdatei oder etwas entsprechendes angesprochen werden soll. Ein URL besteht aus vier Teilen: dem *Protokoll,* der *Serverbezeichnung,* dem *Path (Pfad)* und dem *Dateinamen.* Ein Beispiel:

```
http://www.inferno.de/pandaemonium/limbus/elysium.html
```

Das Protokoll definiert die Methode, mit der die Datei aktiviert wird. Bist Du im Web, ist das `http`. Nach der Protokollbezeichnung folgen ein Doppelpunkt und zwei Schrägstriche mit der nachfolgenden Serverbezeichnung. In den führenden Browsern kannst Du `http://` fortlassen, wenn Du das Adreßfeld ausfüllst, weil die Browser das automatisch hinzufügen. In der HTML-Datei dagegen *mußt* Du `http://` vor die Adresse schreiben.Der Path gibt den Ort der Datei auf dem Server an. In unserem Beispiel handelt es sich um eine Datei im Ordner `limbus` im Ordner `pandaemonium,` der im Stammverzeichnis des Servers liegt. Bist Du an DOS gewöhnt, mußt Du darauf achten, daß der Schrägstrich in einem URL *nicht* der Backslash ist. Die kürzeste mögliche Path-Angabe ist '`/`' – sie bezeichnet eine Plazierung im Web-Stammverzeichnis. Eine Serverbezeichnung macht keinen Unterschied zwischen Groß- und Kleinbuchstaben. `www.lego.com` ist dasselbe wie `WWW.LeGo.Com` . Im Path - und im Dateinamen spielt die Groß- und Kleinschreibung dagegen eine Rolle: `exempel.HTML` ist *nicht* dasselbe wie `Exempel.html` Das beachten etliche Benutzer nicht – sie schreiben jeden URL mit Kleinbuchstaben. Darum empfiehlt es sich, Dateinamen nur mit Kleinbuchstaben zu erstellen.

Dateiname der Startseite

Ein URL muß nicht unbedingt einen Dateinamen enthalten. Fehlt dieser, sucht der Server im angegebenen Ordner nach einer Datei mit einem bestimmten Namen, den er aufgrund seiner Einstellung dort vermutet. Meist sucht er nach der Bezeichnung `index.html`, es kann aber auch `default.html`, `welcome.html` oder ähnliches sein. Dein Provider kann Dir sagen, wie der richtige Name lautet. Findet der Server eine Datei mit dem erwarteten Namen, schickt er diese. Andernfalls schickt er einen Überblick über den Inhalt des Ordners.

Mit anderen Worten ist `http://www.olav.dk/einordner/` dasselbe wie `http://www.olav.dk/einordner/index.html`

Darum solltest Du die Eingangsseite in Deiner Website als `index.html` bezeichnen, damit der URL so kurz wie möglich wird. Übrigens kannst Du den abschließenden Schrägstrich auch weglassen, so daß der URL lautet: `http://www.olav.dk/einordner`

Relative URLs

Verweist ein Link auf eine Datei im selben Computer, muß der URL nicht in kompletter Form mit Protokoll- und Servernamen erscheinen. Du kannst Dich damit begnügen, die Position der Datei in der Dateistruktur *im Verhältnis* zu dem Dokument zu beschreiben, von dem aus verknüpft wird. Das nennt man einen *relativen URL.*

Am einfachsten erklären wir die Sache mit ein paar Beispielen. Unser Bild zeigt eine Dateistruktur mit Webseiten, die in einander untergeordneten Ordnern gespeichert sind.
Auf ein Dokument in demselben Ordner verweist Du schlicht mit dem Dateinamen. Soll `busch.html` mit `gras.html` verknüpft werden, ist der URL einfach `gras.html`.

Eine Datei tiefer im Dateisystem wird durch ihren Suchpfad angesprochen, also mit den Namen der Ordner, durch die man zur gewünschten Datei gelangt. Jedem Ordnernamen folgt ein Schrägstrich:

- `busch.html` verknüpft mit `gurke.html` über `kuechengarten/gurke.html`
- `gurke.html` verknüpft mit `fliegenpilz.html` über `krautgarten/fliegenpilz.html`
- `busch.html` verknüpft mit `fliegenpilz.html` über `kuechengarten/krautgarten/fliegenpilz.html`

Ein „Aufwärts"-Verweis, also ein Verweis auf den Ordner, der den aktuellen Ordner enthält, erfolgt mit zwei Punktzeichen anstelle des Ordnernamens:

- `gurke.html` verknüpft mit `gras.html` über `../gras.html`
- `fliegenpilz.html` verknüpft mit `gras.html` über `../../gras.html`

Schließlich kann man auch auf Dateien in anderen Verzweigungen verweisen, indem man „aufwärts und wieder abwärts" geht:

- `gurke.html` verknüpft mit `nachtschatten.html` über `../blumenbeet/nachtschatten.html`
- `nachtschatten.html` verknüpft mit `fliegenpilz.html` über `../kuechengarten/krautgarten/fliegenpilz.html`

Relative URLs sind praktisch, weil sie auch dann funktionieren, wenn der Inhalt Deines Webordners von Deinem eigenen Computer auf den Webserver überführt wird.

Beginnt ein URL mit einem Schrägstrich, bedeutet das, daß sein Suchpfad vom Stamm des Servers ausgeht, wie auch bei einem absoluten URL. Auf dem Server `www.danhave.dk` verweist der URL `/index.html` auf dasselbe Dokument wie der absolute URL: `http://www.danhave.dk/index.html`, und der URL `/bilder/ko.gif` verweist auf `http://www.danhave.dk/bilder/ko.gif`

So kann man auf einfache Weise auf einen festen Ort der Dateistruktur verweisen, ohne den kompletten Servernamen zu benutzen. Leider funktioniert das nur auf dem Server selbst, nicht aber, wenn man HTML-Dokumente unmittelbar von der eigenen Festplatte einliest.

Ankerpunkte

Ein URL kann durch Zufügung eines '#' und des Ankernamens auf einen Ankerpunkt in einem Dokument verweisen: „`pudrbaer.html#kapitel2`". Ein solcher Ankerpunkt kann in absoluten und in relativen URLs benutzt werden. Verweist ein Link auf einen Ankerpunkt im selben Dokument, reicht es, wenn Du den Punkt angibst: `#kapitel2`.

Frames

Frames (Rahmen) ermöglichen die Einteilung des Browserfensters in mehrere Teilfenster. Jeder dieser Teilbereiche enthält sein eigenes Dokument, das Du unabhängig von den anderen scrollen (rollen) kannst. `www.microsoft.com/workshop/prog/ie4` ist ein gutes Beispiel dafür, was man mit Frames machen kann. Die Sache funktioniert folgendermaßen: Du erstellst ein 'Mutterdokument', das sogenannte *Frameset*, das das Hauptfenster unterteilt und angibt, welche Dokumente in den einzelnen Frames erscheinen. Der Inhalt der einzelnen Frames besteht aus separaten Dokumenten (mit den eventuellen weiteren Dateien), die auch wie normale Ganzfenster-Dokumente geladen werden könnten.

Die Funktionalität eines Links wird hier erweitert. Auf einer normalen Webseite aktiviert ein Link stets eine neue Seite (oder einen anderen Bereich der ursprünglichen Seite) auf dem Bildschirm. Ein Link in einem Dokument eines Frames kann eine neue Seite in diesen Frame laden, sie in einen anderen Frame des Framesets setzen oder in ein Ganzfenster. Zu diesem Zweck erhalten die Links einen neue Eigenschaft namens **target**, die bezeichnet, wo das vom Link angesprochene Dokument erscheinen soll. Jeder Frame hat einen eigenen Namen, der nicht zu verwechseln ist mit dem Namen des Dokuments, das in diesem Frame erscheint.

Framesets können mehrere Ebenen enthalten. Eine Seite kann mit einem einzelnen Frameset in vier Frames unterteilt werden, sie kann aber auch aus zwei Frames bestehen, die jeder wiederum ein Frameset mit zwei Frames enthalten. Es läßt sich nicht unmittelbar aus einer Seite mit zahlreichen Frames ablesen, wie viele Frames-Ebenen sie enthält.

Unter den von mir getesteten Editoren erlaubt meiner Meinung nach nur *PageMill* einen akzeptablen Umgang mit Frames. Der *Netscape Composer* unterstützt Frames gar nicht erst, und *FrontPage* und *Claris HomePage* gehen damit ziemlich umständlich um.

In *PageMill* kannst Du ein Dokument relativ einfach in Frames unterteilen. Du wählst Edit|Split Frame Horizontally oder Edit|Split Frame Vertically. Anschließend hast Du drei Dokumente: das eigentliche Frameset sowie ein Dokument für jeden Frame. Du speicherst einen Frame, indem Du ihn durch Klicken markierst und dann

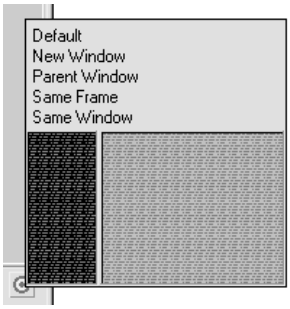
File|Save frame wählst. Das Frameset speicherst Du über File|Save Frameset. Ein Frame hat folgende Eigenschaften:

- **Name** – der Name des Frames
- **Width/height** – Breite/Höhe in Pixel oder Prozent des Fensters
- **Margin** – Abstand zwischen dem Frame-Rand und seinem Inhalt an Text oder Grafik
- **Scrollbars** gibt an, ob Laufleisten erscheinen sollen. Ist der Wert **auto,** erscheinen sie nur, wenn das Dokument größer ist als der Frame.
- **Resize** gibt an, ob der Frame eine andere Größe haben kann, zum Beispiel wenn der Benutzer die Fenstergröße ändert.

Diese Eigenschaften werden im Inspector über das Registerblatt Frame eingestellt. Außerdem enthält der Inspector für jeden Frame das normale Page-Registerblatt, in dem Du Hintergrundfarbe und so weiter für das Dokument im Frame einstellst.

Targets

Hast Du einen Link in einem Frame markiert, wird das 'Ziel'-Symbol in der rechten Seite der Link-to-Leiste sichtbar. Klickst Du es an, kannst Du festlegen, in welchen Frame oder welches Fenster das angesprochene Dokument eingelesen wird. Du siehst eine Darstellung der im Fenster enthaltenen Frames und kannst außerdem Parent Window wählen, was bedeutet, daß die neue Seite das Gesamtfenster füllt statt des Framesets.



- **Same Window** (`_top`): Die Seite wird anstelle des oberen Frameset-Dokuments eingelesen. So erstellst Du einen Link, der das Frameset verläßt.
- **Parent Window** (`_parent`): Die Seite wird anstelle des Frameset-Dokuments eingelesen, das den Frame enthält, von dem aus verknüpft wird – dasselbe wie `_top`, wenn das Frameset nur einen Level enthält.
- **Same Frame** (`_self`): Die Seite erscheint im selben Frame. Das ist üblicherweise der Fall, wenn **Base Target** nicht definiert wurde.
- **New Window** (`_blank`): Die Seite wird in ein neues, für sie eingerichtetes Fenster eingelesen.

In *FrontPage* und *Claris Home Page* mußt Du ein Link-Target (Link-Ziel) von Hand in den Link-Dialog einfügen. Hier schreibst Du den Namen des Frames, der den Link ansprechen soll, oder einen der speziellen Frame-Namen, die mit einem Unterstrich _ beginnen. Gibst Du als Target einen Namen an, der *nicht* zu einem existierenden Frame oder Fenster gehört, erstellt der Browser ein neues Browserfenster mit dem angegebenen Namen, wenn Du den Link anklickst. Das angesprochene Dokument wird in dieses Fenster eingelesen.

Eine Seite hat die Eigenschaft **base target**, die das Target für alle Links auf der Seite angibt, die nicht besonders definiert wurden. So kannst Du auf einen Schlag das Target für alle Links zu einem anderen Frame für etwa ein Inhaltsverzeichnis setzen.

Vorsicht!

Paß auf, wenn Du Frames durch Links miteinander verbindest! Enthält Dein Frame einen Link zu einem weiteren Frameset, wird dieses Set in den einzelnen Frame geholt, es sei denn das Target des Link ist `_parent` oder `_top`. Entsprechend mußt Du, wenn Du von einem Frame aus einen Link aus Deiner Website hinaus setzt, sein Ziel als `_top` angeben – sonst findet der Benutzer nie den Weg aus Deinem Frameset.

Unsichtbare Rahmen

Die Rahmen zwischen den einzelnen Frames sind nicht gerade schön. *Netscape Navigator 3* und *MSIE* können unsichtbare Rahmen wiedergeben. Das wird aber bisher von den WYSIWYG-Editoren nicht unterstützt. Also mußt Du den notwendigen Code selbst einsetzen – siehe den Abschnitt über die manuelle Änderung von HTML-Codes auf Seite 39. Du machst das, indem Du dem **frameset**-Tag oder -Code, der im Frameset-Dokument auftritt, eine Eigenschaft zuweist. *Netscape Navigator* und *MSIE* benutzen dafür jeweils verschiedenen Code, die Kombination der drei folgenden Eigenschaften funktioniert aber in beiden Programmen:

```
<FRAMESET
  BORDER=0
  FRAMEBORDER=0
  FRAMESPACING=0>
```

In *PageMill* kann man dummerweise den Code eines Frameset-Dokuments nicht einfach ändern, was man ja bei anderen Dokumenten kann. Du mußt das Frameset-Dokument also in einem Texteditor öffnen, etwa in *Wordpad*. Einfachheitshalber solltest Du also *Wordpad* ins switch to-Menü einfügen – über den Menüpunkt Edit|Preferences|Switch to (`Notepad.exe` findest Du im `Windows`-Ordner). Anschließend kannst Du, wenn das Frameset im Editor offensteht, den Menüpunkt Edit|Switch to|Notepad.exe aktivieren.

Links, die den Inhalt mehrerer Frames gleichzeitig ändern

Ein Gedankenexperiment: Du hast ein Frameset, das das Fenster in drei Frames aufteilt. Einer der Frames enthält ein Dokument mit einem Link, der bei Aktivierung den Inhalt von zwei Frames gleichzeitig ändert. Ein normaler Link hilft Dir hier leider nicht weiter. Es gibt aber mehrere Auswege:

1. Erstelle ein neues Frameset mit der gleichen Unterteilung, das eine neue Kombination von Dokumenten in den Frames enthält, und verknüpfe es mit dem Link - (Das **target** setzt Du auf **same window**).
2. Liegen die beiden Frames, deren Inhalt gewechselt werden soll, nebeneinander, kannst Du einen Unter-Frameset mit den beiden Dokumenten erstellen, der durch ein neues Frameset mit den erwünschten Dokumenten ersetzt wird.
3. Mit JavaScript kannst Du einen „Doppellink“ erstellen. Du erstellst einen Link, der den einen Frame auswechselt, und fügst dem Link-Code die folgende JavaScript-Eigenschaft hinzu:

```
onClick="top.framename.location='dateiname.html'"
```

(Siehe im übrigen den Abschnitt JavaScript, Seite 46, und den Abschnitt Eigenschaften im WYSIWYG-Zustand einfügen, Seite 40). Beispiel für den HTML-Code für den Link:

```
<A HREF="seite1.html" TARGET=frameNr1 on-
Click="top.frameNr2.location='seite2.html'">
```

Im Grunde würde ich Dir aber empfehlen, Links, die gleichzeitig zwei Frames ändern, zu vermeiden. Überlege lieber, ob Du nicht dasselbe mit weniger Frames erreichst.

Noframes

Kann der Browser nicht mit Frames umgehen, kann er natürlich nichts mit den Definitionen im Frameset-Dokument anfangen. Also sollte das Dokument für diesen Browsertyp auch einen konventionellen Webseiten-Inhalt enthalten. Hier hilft Dir der Menüpunkt Edit|No Frames Message. Es gibt Leute, die hier schlicht schreiben „Besorg Dir einen ordentlichen Browser” und einen Link zu Netscapes Website einfügen – empfehlen würde ich das nicht gerade. Da ist es schon etwas klüger, wenn die NoFrames-Seite eine ungefähre Kopie der Eingangsseite oder des Inhaltsverzeichnisses enthält, so daß die Website auch über eine frame-lose Version verfügt.

Frames und ihre Probleme

Willst Du Frames anwenden, solltest Du die folgenden Nachteile bedenken:

- Nur die ursprüngliche Frame-Aufstellung hat einen URL. Wird der Inhalt der einzelnen Frames durch Anklicken von Links gewechselt, kann man keine Lesezeichen machen und nicht mit diesem Frame verknüpfen.
- Ein Frameset läßt sich nicht drucken – das geht nur mit einzelnen Seiten.
- Es ist unpraktisch, daß man seine Website sozusagen in zwei Versionen erstellen muß, damit sie mit und ohne Frames funktioniert.
- Im *Netscape Navigator 2.0* funktionieren die Vor- und Zurück-Schalter nicht auf Frames. Willst Du innerhalb eines Frames zurück, mußt Du die rechte Maustaste benutzen, was die Navigation in Framesets ziemlich erschwert. Im *Navigator 3.0* funktionieren Vor- und Zurück-Schalter auch in Frames.

So ‘mißbrauchst’ Du HTML für ansprechendes Layout und gute Typographie

Vermutlich hast Du bereits festgestellt, daß HTML unmittelbar kein besonders flottes Layout anbietet: Doppelter Zeilenabstand zwischen Absätzen, Überschriften grundsätzlich fett und so weiter. Das liegt nicht an den HTML-Spezifikationen, sondern ist eher eine Konvention der Browser. Außerdem hat HTML nur recht primitive Möglichkeiten zur Kontrolle von Typographie und Layout – jedenfalls bis die Style Sheets sich durchsetzen, siehe Seite 41. Dabei handelt es sich im großen und ganzen um Tricks, die die Möglichkeiten von *Netscape Navigator* und *MSIE* zur Formatierung von Webseiten ausnutzen – wir wollen sie hier ohne Garantie für ihre Funktionalität in anderen Browsern vorstellen.

Text als Grafik

Gestaltest Du Deinen Text als Grafik, hast Du perfekte Kontrolle über Dinge wie Schriftsatz, Engstellung und dergleichen. Mit dieser Methode umgehst Du alle typographischen Einschränkungen in HTML. Das ist besonders praktisch bei Titeln für die Eingangsseite und bei Firmenbezeichnungen, da hier das Aussehen eine große Rolle spielt. Text ergibt als Grafik meist recht kleine GIF-Dateien, da er wenige Farben und einfache Formen enthält. Vergiß nicht, seine Ränder mit Anti-Aliasing zu behandeln und einen **alternativen Text** für die Grafik einzufügen.

Kapitälchen

Du schreibst mit großen Buchstaben und vergrößerst den ersten Buchstaben über **font-size** (Schriftgrad).

Gesperrter Druck

Du setzt nach jedem Buchstaben einen Zwischenraum und steuerst seine Größe über **font-size** (Schriftgrad).

Kompaktere Absätze

Der Browser setzt grundsätzlich eine Leerzeile zwischen zwei Absätze. Dadurch wird das Schriftbild aber unrein. Du kannst das vermeiden mit einem festen Zeilenwechsel (*Umschalt* + *Return*) statt eines Absatzwechsels (*Return*). Einen neuen Absatz markierst Du, indem Du seine erste Zeile um eine oder mehrere feste Leerstellen einrückst (*Strg* + *Leertaste*). Dadurch wird allerdings der gesamte Text technisch zu einem einzigen Absatz, also kannst Du auch nicht etwa eine Absatzformatierung nur für die Überschrift benutzen. Statt dessen kannst die Überschrift mit Schriftformatierungen wie Schriftgröße und Schriftfarbe hervorheben.

Größerer Zeilenabstand

Einen doppelten Zeilenabstand erreichst Du mit einem Absatzwechsel nach jeder Zeile. Genauere Kontrolle über den Zeilenabstand erreichst Du so: Du setzt einen festen Zeilenwechsel (*Umschalt* + *Return*) nach jeder Zeile, beendest jede Zeile mit einem sogenannten Nonbreaking Space (feste Leerstelle) (*Umschalt* + *Leertaste*) und machst die Schriftgröße der Leerstelle größer als die des Textes.

Seitenränder

Seitenränder lassen sich auch mit einem Einzug erstellen, genauer geht das aber über Tabellen. In einer Tabelle stellst Du einen Seitenrand mit einer Spalte her, die nur die passende Anzahl fester Leerstellen enthält.

Luft

Luft und Abstand gehören zu den wichtigsten Hilfsmitteln beim Layout. Leerfelder von kontrollierter Größe erreichst Du mit einem 1 x 1 Pixel großen transparenten GIF. Das ergibt ein unsichtbares Bild, um das Du mit Hilfe der Bildeigenschaften **vspace** und **hspace** den Seitenrand steuern kannst. Diese Methode ist genauer als die Änderung der Schriftgröße einer festen Leerstelle, funktioniert aber nicht, wenn die Bildfunktion im Browser deaktiviert wurde.

Netscape Navigator 3 verfügt übrigens über ein neues Element namens **Spacer**, mit dem Du Leerfelder einfacher erstellen kannst – da das aber bisher nur vom *Navigator 3* unterstützt wird, würde ich seine Anwendung nicht empfehlen.

Mehrere Zeichensätze

HTML 3.2 erlaubt eigentlich nur zwei Zeichensätze – eine Standardschrift, meist *Times*, und eine ‘Schreibmaschinenschrift’, meist *Courier*.

Netscape Navigator 3 und *MSIE 2* und später unterstützen das Tag **font face**, mit dem man angibt, welcher Zeichensatz für den Text benutzt wird. Das hört sich praktisch an – nur muß der Endbenutzer dann auch den angegebenen Zeichensatz auf seiner Maschine haben, sonst funktioniert das nicht, und nicht alle Leute haben dieselben Schriften. Glücklicherweise erlaubt die Eigenschaft für dieses Tag die Angabe mehrerer Zeichensätze, die durch Kommata voneinander getrennt werden. Der Browser wählt nun den ersten Zeichensatz in dieser Liste, den der Endbenutzer in seinem System hat.

Ein Beispiel: `FONT FACE="Optima, Futura, Helvetica, Arial"`. Systeme, die den Zeichensatz Optima installiert haben, zeigen diese Schrift. Ist sie nicht vorhanden, wird Futura gewählt, dann Helvetica und so weiter. Ist keine der angegebenen Schriften vorhanden, wird die Standardschrift benutzt.

Eigentlich gibt es nur drei Schriften, die man bei jedermann voraussetzen kann. Im PC werden sie anders bezeichnet als im Mac, also müssen beide Namen mit einem Komma dazwischen angegeben werden:

- **Times, Times New Roman** – die Standardschrift, in der auch dieses Heft geschrieben wird. Experten bezeichnen diesen Zeichensatz - gerade für lange Texte - als den lesefreundlichsten.
- **Arial, Helvetica** – eine Groteskschrift, das heißt eine Schrift ohne 'Füßchen'. Wird oft für Überschriften benutzt – so auch in diesem Heft.
- **Courier, Courier New** – die klassische Schreibmaschinenschrift.

Achtung! Mit dem Tag **font face** darfst Du niemals, ich wiederhole, niemals mathematische Symbole, griechische Schriftzeichen oder so etwas formatieren. In Browsern, die dieses Tag nicht verstehen, führt das zu Unsinn – aber damit nicht genug, sind die Ergebnisse auch noch je nach Plattform recht verschieden.

Tricks mit Hintergrundgrafik

Ein Hintergrundbild wird senkrecht und waagerecht über das gesamte Fensterareal wiederholt. Willst Du diese Wiederholung vermeiden, machst Du das Bild entsprechend groß. Bei einer Größe von 1600 x 1200 Pixel entfällt die Wiederholung selbst auf den größten Bildschirmen – es sei denn die Seite ist länger. Das eigentliche Motiv muß nicht unbedingt die ganze Bildfläche füllen – im Grunde sollte es höchstens 640 x 400 Pixel einnehmen. Der Rest der Bildfläche ist einfarbig.

Es gibt zahlreiche Seiten, die in mehrere **senkrechte** Hintergrundfarben aufgeteilt sind, zum Beispiel im ersten Drittel rot und für den Rest der Breite weiß. Das erreicht man mit einer einzelnen langen und schmalen Grafik, deren erster Teil rot und der Rest weiß ist. In der Höhe reicht für diese Grafik ein Pixel, die Breite muß aber mindestens 1600 Pixel betragen.

Bist Du neugierig, wie eine bestimmte Grafik erstellt wurde, klickst Du mit der rechten Maustaste darauf, wählst Save Background As (Hintergrund speichern unter) und speicherst die Hintergrundgrafik auf Deinem Desktop. Anschließend kannst Du sie wieder in den Browser einlesen.

Der *Netscape Navigator* erlaubt nicht, daß Text oder Grafik bis an den Rand des Fensters reicht. Außerdem gibt es je nach System leichte Unterschiede in der Größe des Abstands vom Fensterrand zum Fensterinhalt. Es ist also nicht möglich, Vorder- und Hintergrund gänzlich zu synchronisieren. Allerdings kann man in einem Frame die Randgröße zwischen der Frame-Kante und dem Inhalt angeben. Soll eine Vordergrundgrafik bis zum Fensterrand reichen, muß das also in einem Frame erstellt werden. Im *MSIE* kannst Du den oberen und linken Seitenrand mit den Eigenschaften **body margintop** und **marginleft** angeben, aber das funktioniert leider nicht im *Navigator*.

Zeilenlänge

Die Zeilenlänge/Spaltenbreite eines Textes kannst Du kontrollieren, indem Du diesen Text in eine Zelle einsetzt, deren Breite in Pixel definiert ist. Vollständige Kontrolle über Zeilenlänge und Umbruch erhältst Du so wohlgemerkt nicht, da die Größe eines 'Normaltexts' etwa bei PC und Macintosh nicht gleich ist, Du hast aber wesentlich mehr Einfluß auf den Text in Tabellen als auf den auf einer Normalseite, wo die Zeilenlänge grundsätzlich an die Fensterbreite angepaßt wird.

Spalten

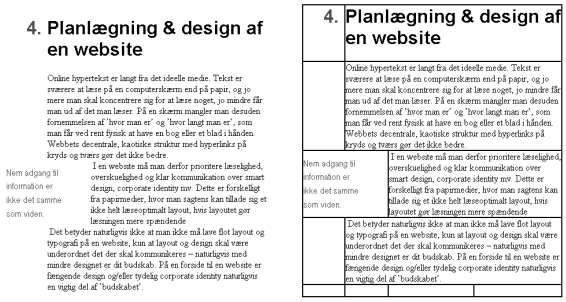
Ein Spalten-Layout erstellst Du, indem Du Deinen Text in kleinere Bruchstücke aufteilst, die Du dann in mehrere waagerechte Zellen einer Tabelle verteilst. Eine Spalte sollte möglichst nicht mehr als 300 Pixel hoch sein, da man sonst durch den Bildschirm scrollen muß, um vom Ende der ersten Spalte zum Anfang der nächsten zu kommen.

Der Nachteil dabei ist, daß Du Deinen Text von Hand so aufteilen mußt, daß Deine Zellen korrekt ausgefüllt werden. Aber so ist das eben…

Leerzellen

Zellen ohne Inhalt sind nützlich für das Layout – allerdings befolgen sie nicht unbedingt die Pixelgrößen, mit denen Du sie definierst. Zellen, die Text enthalten, können zum Beispiel 'leere' Zellen durch ihre Breite „wegdrücken". Setzt Du eine Zeile mit einer Reihe von *nonbreaking spaces* in eine leere Zelle, zwingst Du sie dazu, mindestens die entsprechende Breite einzunehmen.

Merkwürdige Einteilungen



Die Balken zwischen den Zellen ziehen sich waagerecht wie senkrecht durch die gesamte Tabelle. Unmittelbar ist es nicht möglich, eine breitere Zelle unter eine schmalere zu setzen. Das kannst Du aber umgehen, indem Du Zellen miteinander verschmilzt.

In unserem Beispiel habe ich ein Layout erstellt, das zunächst lauter Zellen von 4 x 5 enthält. Anschließend habe ich Zellen auf verschiedene Weise waagerecht miteinander verschmolzen und dann meine Textteile in diese Zellen eingesetzt. Um die Breite der Säule zu sichern, habe ich in die untere linke Zelle eine Reihe von festen Leerstellen gesetzt.

Die Artikel in `www.hotwired.com/synapse/` sind ein gutes Beispiel für die Benutzung von Tabellen im Layout.

Vom Textdokument zu HTML

Die perfekte Methode, um Text aus einer Textverarbeitung in HTML zu verwandeln, gibt es nicht. Zum einen sind die Unterschiede zwischen dem Web- und dem Papiermedium (für das Textverarbeitungen entwickelt wurden) groß, zum andern sind die typographischen und layoutmäßigen Möglichkeiten von HTML immer noch recht primitiv, verglichen mit denen einer normalen Textverarbeitung. Layout, also Seitengestaltung, Seitenränder, Textfelder, Grafikplazierung, Spalten, lassen sich nicht ohne weiteres in HTML überführen.

Nur wenige typographische Formatierungen wie fett, kursiv, Schriftgröße (wenige Stufen), Einzug und Justierung sind anwendbar. Formatierungen wie Zeilenabstand, Einzug der ersten Zeile in einem Absatz, Tabulatoren, Seitenränder, Abstand zwischen Absätzen und so weiter werden in HTML nicht unterstützt.

Viele Spezialzeichen und mathematische Symbole werden von HTML nicht unterstützt.

Bilder *können* von Textverarbeitungen in HTML überführt werden, meist ist es aber einfacher, die ursprüngliche Bilddatei in GIF oder JPEG zu konvertieren und dann über den Webeditor in die Seite einzufügen. Überschriften, Listen und Tabellen lassen sich dagegen ohne weiteres in HTML übersetzen.

Im Endergebnis kann also nur der Text mit seiner funktionellen Struktur, also Überschriften, fett, kursiv, Listen und Tabellen, in einer Textverarbeitung erstellt und in HTML überführt werden. Die visuelle Seite der Sache, als da wären Layout, Farben, Grafik und so weiter, muß nach dem Import des Textes im Webeditor erstellt werden. Auch die Links – und eventuelle Multimedia- oder interaktive Elemente – müssen selbstverständlich im Webeditor eingefügt werden, da sie eng mit dem HTML-Format verbunden sind.

Text läßt sich auf zwei Arten von einer Textverarbeitung in HTML überführen: Du kannst das Dokument von der Textverarbeitung in HTML konvertieren lassen oder aber das Dokument der Textverarbeitung in den Webeditor einlesen und dort die notwendige Konvertierung vornehmen. In jedem Fall wird nur die grundlegende Struktur überführt – alles andere muß im Webeditor überarbeitet und mit Layout versehen werden.

PageMill und *FrontPage* können Dokumente der üblichsten Textverarbeitungen lesen wie auch Dokumente im Format RTF, einer Art 'Austauschformat' für Textdokumente. Alle Textverarbeitungen können Text als RTF speichern.

Netscape Composer und *Claris Home Page* können keine Textverarbeitungsformate lesen, sondern nur HTML, oder reinen, nichtformatierten Text. Hier mußt Du Deinen Text also entweder bereits in der Textverarbeitung als HTML speichern oder aber als reinen Text, wodurch natürlich alle Formatierungen verloren gehen.

Die neuesten Versionen von *Word* und *WordPerfect* können ein Dokument unmittelbar als HTML speichern. Etwas ältere Versionen können über Makros oder Hilfsprogramme um die Möglichkeit der Speicherung als HTML erweitert werden, zum Beispiel mit Microsofts *Word Internet Assistant*.

QuarkXPress, Pagemaker etc.

Es ist möglich, Dokumente aus Layoutprogrammen wie *QuarkXPress* und *Pagemaker* in HTML zu überführen, wie bei Textverarbeitungen geht unterwegs aber ein Großteil an Formatierung und Layout verloren. Es lohnt sich also nicht, Webseiten in diesen Programmen zu erstellen und sie dann in HTML zu konvertieren. Normalerweise wird ein solches Layout-Dokument mit Text aus einer Textverarbeitung erstellt. Also ist es sicher einfacher, den Text der Textverarbeitung und die gewünschten Bilder im Webeditor zu sammeln.

Welcher Browser unterstützt was?

Man muß wissen, welche Browser welche Möglichkeiten unterstützen.

Bekanntlich wird HTML regelmäßig um neue Ideen der führenden Browserentwickler Netscape und Microsoft bereichert. W3C, Entwickler und 'Wächter' des HTML-Standards, denkt im allgemeinen länger und gründlicher über alles nach, weswegen der von dort empfohlene Standard meist der reellen Situation im Netz ein Jahr hinterherhinkt. Das bedeutet, daß experimentelle Erweiterungen ausprobiert werden können, bevor sie akzeptiert werden. Zum Beispiel waren viele Webdesigner begeistert von Frames, als sie von Netscape eingeführt wurden, inzwischen haben sie sich aber als etwas problematisch erwiesen, weswegen sie vermutlich nie in den empfohlenen Standard integriert werden. W3C und die Browserproduzenten experimentieren nun mit einem Frame-Modell, das über Style Sheets erstellt wird, was die Sache hoffentlich besser macht. Andererseits benutzen viele Leute natürlich Frames, bis eine Alternative mit entsprechender Funktionalität erscheint, ohne Rücksicht auf die Meinung von W3C.

Heute heißt die empfohlene Version HTML 3.2, die den Großteil der aktuellen Möglichkeiten enthält. Hältst Du Dich an die Tags von HTML 3.2, kannst Du davon ausgehen, daß alle durchschnittlichen Browser Deine Dokumente darstellen. Es folgt nun eine Liste der Erweiterungen, die nicht zum Standard gehören, sowie die Angabe der Browser, die sie unterstützen.

Navigator 2.0 und MSIE 2.x

Folgende Tags sind nicht HTML-Standard, werden aber so gut wie allgemein unterstützt:

- Frames (**frameset, frame, target, base target**)
- Einsatz von Plug-Ins (**embed**)
- Animierte GIFs

Navigator 3.0 und MSIE 3.0

Folgende werden weitgehend, aber nicht allgemein unterstützt:

- Farbige Framekanten oder unsichtbare Frames (allerdings mit verschiedenen Tags!).
- Angabe des Zeichensatzes (**font face**).
- Hintergrundfarbe in Tabellenzellen. (**td/tr/th bgcolor**)

Nur Navigator

Folgende Tags werden vom meistbenutzten Browser unterstützt, es ist aber fraglich, ob sie anderswo akzeptiert werden:

- Blinkender Text (**blink**, ab Navigator 1.1) Es gibt komplette Websites, die sich dem Hohn und Spott dieses für viele offensichtlich irritierenden Tags widmen. Netscapes Rechtfertigung findest Du, wenn Du in Netscapes Adressenfeld `about:mozilla` schreibst.
- Spalten (**multicol**, ab Navigator 3) Dieselbe Wirkung erreichst Du mit Tabellen.
- Leere Felder (**spacer**, ab Navigator 3) Eigentlich ein recht praktisches Tag, mit dem Du Leerfelder erstellen kannst. Mit Style Sheets erreichst Du aber dasselbe.
- Den Inhalt eines Formulars an eine E-Mail-Adresse anstelle eines CGI-Programms zu senden. Das ist eigentlich sehr praktisch – schade, daß MSIE das nicht unterstützt.

Nur MSIE

Diese Tags genießen keine breitere Unterstützung und werden das vermutlich nie erreichen:

- Hintergrundmusik (**body bgsound**) Mit Plug-Ins kannst Du dasselbe auf allgemein unterstützte Weise erreichen.
- Wasserzeichen (**body bgproperties=fixed**), ein Hintergrundbild, das nicht mit dem Text gescrollt wird. Dasselbe erreichst Du mit Style Sheets (obwohl ich nicht weiß, was das eigentlich soll …).
- Seitenrand (**body leftmargin/topmargin**) Gibt den Abstand von der Fensterkante zum Inhalt der Seite an. Eigentlich sehr praktisch! In Style Sheets kannst Du dies ebenfalls angeben.
- Erweiterungen zu **img**, um Filme darzustellen (**img dynsrc**). Mit Plug-Ins erreichst Du dasselbe auf allgemein unterstützte Weise.
- Schwebende Frames (**iframes**). Das ist eine Webseite, die als Fenster über eine andere Webseite gelegt wird. Das Fenster scrollt mit der Webseite.
- Waagerecht rollender Text (**marquee**). Der Grund dafür, daß ich Navigator vorziehe. Mit der ActiveX-Technik kannst Du auch senkrecht rollenden Text erstellen.
- Erweiterungen zum Aussehen von Tabellen, zum Beispiel farbige Schatten und Hintergrundbilder in Zellen.

Folgende Tags werden noch nicht allgemein unterstützt, das kommt aber sicher:

- W3C-entwickelte Erweiterungen zu Tabellen.
- **Object**-Tag
- Style Sheets

Plug-Ins, Java, JavaScript, ActiveX

Im Teil 4 erfährst Du mehr darüber, wie diese Techniken funktionieren und in wieweit sie unterstützt werden. Im allgemeinen kann man nicht unbedingt damit rechnen, daß sie benutzt werden – der Browser unterstützt sie zwar, sie können aber deaktiviert werden.

CGI

Formulare und CGI werden von allen Browsern unterstützt.

Für welche Browser soll man planen?

Solltest Du die neuesten Erweiterungen benutzen und Deine Besucher auffordern, den neuesten Browser downzuloaden? Oder solltest Du Deine Seiten so entwerfen, daß selbst die ältesten Browser sie darstellen können?

Das ist eine vieldiskutierte Frage. Und dabei kannst Du ohne weiteres Seiten erstellen, die die neuesten Möglichkeiten ausnutzen und dennoch auch auf älteren Browsern zufriedenstellend dargestellt werden.

Allgemein würde ich sagen, daß Du für die Bedürfnisse Deines Publikums entwerfen solltest. Und man hat nun einmal den Browser, den man hat – Du wirst vermutlich niemanden zum Downloaden eines neuen Browsers veranlassen, wenn Du schreibst: ‘diese Seite wirkt am besten mit …’

Browser und ihre Verbreitung

Drei Browser sind weitverbreitet: *Netscape Navigator 2.0, Netscape Navigator 3.0* und *MSIE 3.0*. Zusammen beherrschen sie vermutlich etwa 95 Prozent des Marktes. Netscape ist entschieden der größte. *MSIE* hat vermutlich etwa 25 Prozent und *Navigator* den Rest. Diese Zahlen sind aber recht grob geschätzt und ändern sich konstant. Die 4.0-Browser werden sich vermutlich im kommenden Herbst ziemlich weit verbreiten, der Navigator 2.0 bleibt aber wohl noch lange der kleinste gemeinsame Nenner.

Den Navigator gibt es für viele Plattformen, einschließlich VMS und OS/2. MSIE gibt es für Win95/NT, Windows 3.11 und Macintosh, das Programm hat aber nur für Win95/NT größere Bedeutung.

Rückversicherung

Benutzt Du Tags, die nicht von allen Browsern verstanden oder angewendet werden, solltest Du das so tun, daß kein Fehler oder Informationsverlust entsteht.

- **Farben**: Die Definition von Text- und Dokumentfarben wird zwar von allen neueren Browsern unterstützt, der Benutzer kann aber eigene Farben definieren, die gegenüber den von Dir im Dokument (über Options|General Preferences|Color (Optionen|Allgemeine Einstellungen|Farben) im *Navigator*) definierten Farben den Vorrang haben. Du solltest also keinesfalls sinngebende Markierungen *ausschließlich* über Farben definieren – etwa rot statt fett benutzen.
- **Zeichensätze**: das **Font face**-Tag wird nicht allgemein unterstützt. Darum solltest Du es nicht als einzige sinngebende Markierung benutzen.
  Font face darf niemals für Zeichen außerhalb des normalen Zeichensatzes benutzt werden – also für griechische Schriftzeichen, Symbole oder ’Wingdings’. Das funktioniert weder in unterschiedlichen Browsern noch auf verschiedenen Plattformen!
- **Grafiken** – Viele Leute deaktivieren Grafiken beim Surfen (Options|Auto Load Images (Optionen|Grafiken automatisch laden) im *Navigator*). Darum solltest Du grundsätzlich alternativen Text angeben, es sei denn das Bild hat ausschließlich dekorative Bedeutung.
- **Clientside Imagemaps** – Manche ältere Browser unterstützen dieses Imagemap-Format nicht. Außerdem nützen Imagemaps wenig, wenn die Grafik deaktiviert wurde. Darum solltest Du die entsprechenden Links immer auch als Text-Links setzen.
- **Frames** – Frames sind ganz besonders vertrackt. Hast Du nicht eine komplette alternative Seite für den Frameset erstellt, sehen Leute, deren Browser Frames nicht unterstützt, schlicht eine leere Seite! Frames werden heute allerdings von den meisten Browsern unterstützt.
- **Zellenfarbe** – ist gefährlich! Hast Du weißen Text auf schwarzem Zellenhintergrund auf einer weißen Seite, wird der Text in allen Browsern, die farbigen Zellenhintergrund nicht unterstützen, unsichtbar.

Andere Dokumentenformate

Alle möglichen Dateiformate lassen sich über das Web überführen. Ein Link kann sich an ein downzuloadendes Programm richten, an einen Videofilm, ein *WordPerfect 4.1*-Dokument… Das einzige Problem ist, ob der Browser weiß, was er mit der empfangenen Datei anfangen soll.

Einen Link zu einer Nicht-HTML-Datei erstellst Du genauso wie einen Link zu einer HTML-Datei. Der einzige Unterschied ist, daß der Dateiname eine andere Endung hat als `.html` .

Sendet der Server eine Datei an den Browser, teilt er zunächst mit, zu welchem *MIME-Typ* die Datei gehört. MIME-Typen sind eine standardisierte Angabeweise für Datenformate. Sie enthalten den Namen einer Hauptgruppe und einer Untergruppe, getrennt durch einen Schrägstrich, zum Beispiel `text/html`, `image/jpeg` oder `world/vrml`.

Woher weiß der Server aber, welchen MIME-Typ eine Datei hat? Ist er nicht speziell konfiguriert, stellt er aufgrund der Dateiendung im Namen eine Vermutung auf – wie übrigens auch der Browser, wenn er eine Datei von der Festplatte öffnet. Darum solltest Du darauf achten, daß Dateien die richtigen Endungen haben.

Aufgrund des MIME-Typs entscheidet der Browser, wie die Datei zu behandeln ist.

HTML-Seiten und GIF/JPEG-Grafik werden unmittelbar im Browserfenster dargestellt. Dateien, die von einem installierten Plug-In verarbeitet werden können, werden von diesem im Browserfenster wiedergegeben. Manche Dateitypen werden an externe Programme weitergegeben – wie zum Beispiel komprimierte Dateien an ein Auspacker-Programm. Andere Dateien werden auf der Festplatte gespeichert, zum Beispiel Programme. Kennt der Browser das empfangene Datenformat nicht, fragt er, was er mit der Datei machen soll. Weiteres findest Du im Heft *Auf ins Word Wide Web* im Abschnitt *Hilfsprogramme für Spezialformate*.

Willst Du vermeiden, daß die Datei von einem externen Programm auf dem Client-Computer geöffnet wird, kannst Du sie komprimieren – dann wird sie nämlich in jedem Fall auf der Festplatte gespeichert.

Welche Dokumentenformate sind anwendbar?

Im Internet geht es grundsätzlich um den Austausch von Informationen in Formaten, die von Absender und Empfänger verstanden werden. Es ist nämlich durchaus möglich, Text im Format *Wordstar 2.5.1 b für Atari* ins Web zu legen – ob der Empfänger dieses Dokument versteht, ist aber nicht unbedingt sicher…

Verläßt Du HTML, solltest Du Dich an die allgemein anerkannten Dokumentenformate halten:

Textverarbeitungsdokumente: RTF

RTF ist ein Austauschformat für Textverarbeitungsdokumente. Die meisten modernen Textverarbeitungen können in diesem Format öffnen und speichern. RTF empfiehlt sich, wenn man ein Textverarbeitungsdokument ins Web legen will. Das Format bewahrt die meisten der Formatierungen, die in einer Textverarbeitung möglich sind, sowie Sonderzeichen wie *ä, ö* und *ü*. RTF wurde von Microsoft entwickelt.

Dokumente mit Layout: PDF

PDF, ein Format, das von der Firma Adobe entwikkelt wurde, ermöglicht den Austausch von Dokumenten mit fertigem Layout.

PDF kann Layout von professionellem Standard ebenso präzise wiedergeben, wie man es in Layout-Programmen wie *Quark XPress* oder *Pagemaker* erstellen kann. PDF eignet sich vor allem für die Überführung von Dokumenten, die ausgedruckt werden sollen, und wird besonders auch für die Veröffentlichung von Handbüchern, Büchern, Broschüren und so weiter benutzt.

PDF erfordert kein Layoutprogramm, um eine Datei zu öffnen. Adobe hat ein kleines kostenloses Programm namens *Acrobat Reader* entwickelt, mit dem man PDF-Dokumente öffnen, lesen und ausdrucken kann.

Den *Acrobat Reader* findest Du bei `www.adobe.com`.

Und dahinter: die HTML-Codes

Bisher haben wir uns an das freundliche WYSIWYG-Interface gehalten, um die Arbeit an Websites zu beschreiben. Nun geht es darum, wie man mit dem eigentlichen HTML-Code arbeitet.

In den Webeditoren kann man sich dafür entscheiden, im Hauptfenster den hinter dem Dokument verborgenen HTML-Code zu zeigen. Hier kann man diesen Code modifizieren oder ihn gänzlich neu schreiben.

Hinein in den HTML-Code	
PageMill	Edit|HTML Source
Composer	Edit|HTML Source (allerdings wird der Code an ein externes Programm gegeben, z.B. *Notepad*.)
FrontPage	View|HTML (Ansicht|HTML)
Claris	Window|Edit HTML Source

Warum sollte man aber im Grunde im HTML-Code arbeiten, wenn man seine Seiten mit WYSIWYG erstellen kann? Normalerweise ist das auch nicht nötig, ein Super-Webdesigner kommt aber aus mehreren Gründen nicht daran vorbei:

- Ein Webeditor hinkt immer einen Schritt hinter einem Browser her. Wird eine neue Erweiterung in einen Browser integriert, kommt ein Webeditor mit dieser Funktion meist erst ein halbes Jahr später heraus. Will man ganz vorne sein, muß man daher vermutlich direkt in HTML schreiben.
- Kein Webeditor ist perfekt. Soll etwas optimal funktionieren, muß man unter Umständen selbst im HTML-Code feinjustieren.

HTML – von Anfang an

HTML ist Text mit Tags. Ein Tag ist ein Formatierungscode, der ein Stück Text markiert und Angaben dazu macht, zum Beispiel wie es aussehen soll. Ein Tag besteht aus einem Code, der von < und > eingeschlossen wird. Die Markierung, an deren Eingang ein Tag steht, wird von einem weiteren Tag mit demselben Namen und mit einem / vor dem Namen abgeschlossen. Hier folgt ein Beispiel für HTML:

```
Standard <B>Das ist eine fette Sache!</B> Standard
```

B ist das Tag für fette Schrift (*Bold*). Ein Tag wird grundsätzlich mit einem einzelnen Wort ohne Zwischenraum oder Sonderzeichen bezeichnet, wie es auch keinen Zwischenraum zwischen den Spitzklammern und der Tagbezeichnung gibt.

Tags können ineinander verschachtelt werden:

```
<B>Baden im Regen macht <I>nicht</I> fett</B>.
```

I ist das Tag für kursive Schrift (*Italic*). Absätze werden ebenfalls durch Tages gekennzeichnet:

```
<H1>Baden im Regen</H1>
<P>Regenschirme sind irrelevant. </P>
<P>Aber Vorsicht bei Gewittern!</P>
```

P ist das Tag für einen Standardabsatz (**Paragraph**), H1 bezeichnet eine **Headline 1** (Überschrift 1). Also sind es die Tags und nicht etwa die Zeilenwechsel, die einen Absatz definieren. Zeilenwechsel (also der Druck auf *Return*) spielen im HTML-Code keine Rolle für das Aussehen der Webseite.

Die meisten Tags bestehen aus einem Anfangs- und einem End-Tag, es gibt aber auch alleinstehende Tags:

```
Nie wieder
<BR>will ich sehen
```

Das BR-Tag bezeichnet einen Zeilenwechsel, was *Umschalt* + *Return* entspricht, und muß nicht geschlossen werden.

Ich schreibe hier alle Tag-Bezeichnungen mit Großbuchstaben. Das ist allgemein üblich und außerdem übersichtlich, es ist aber nicht notwendig – HTML-Tags kennen keinen Unterschied zwischen Groß- und Kleinbuchstaben.

Attribute

Die meisten Tags lassen sich durch *Attribute* erweitern – genauere Angaben zur Funktion dieses Tags. Ein Attribut besteht im allgemeinen aus einem Namen und einer Wertangabe, die durch ein Gleichheitszeichen miteinander verbunden werden. Ein Link kann zum Beispiel folgendermaßen aussehen:

```
<A HREF="anderswo.html">Hier nicht klicken!</A>
```

A bedeutet *Anchor* (Anker, Ziel), das Attribut HREF bedeutet *Hyper Reference* (Zielverweis), und sein Wert ist ein URL – in diesem Falle ein relativer.

Eine Grafik wird über das IMG-Tag folgendermaßen eingefügt:

```
<IMG SRC="/imgs/ente.gif" WIDTH=100 HEIGHT=150 ALT="Ente">
```

Die Attribute eines Tags entsprechen in etwa den Eigenschaften, die man in *PageMills* Inspector oder im Properties(Eigenschaften)-Dialog in *FrontPage* und *Composer* einfügen kann, wenn das entsprechende Element im Editor gewählt ist.

Der Name eines Attributs besteht wie ein Tagname aus einem einzelnen Wort ohne Zwischenraum oder Sonderzeichen. Attributwerte dagegen können alle möglichen Zeichen enthalten, solange sie in Anführungszeichen stehen. Enthält der Wert nur Buchstaben, Zahlen oder Bindestriche, sind die Anführungszeichen überflüssig.

Ein Attribut kann auch als alleinstehendes Wort erscheinen – wie in `<BR CLEAR>`.

Tagname und Attribut müssen durch einen Zwischenraum getrennt werden, wie auch mehrere aufeinander folgende Attribute. Zwischen Attributnamen, Gleichheitszeichen und Attributwert dagegen entfällt der Zwischenraum.

Bei mehreren Attributen spielt ihre Reihenfolge keine Rolle.

Attribute werden nicht in den End-Tags wiederholt:

```
<FONT SIZE="+1">Diese Schrift ist groesser</FONT>
```

Syntaxregeln

Tags dürfen sich nicht überschneiden:

Falsch: `<P>normal<B>fett</P><P>fett</B> normal</P>`

Richtig: `<P>normal<B>fett</B></P><P><B>fett</B>normal</P>`

Struktur

Ein HTML-Dokument enthält grundsätzlich zwei Blöcke: den HEAD- und den BODY-Block. Der HEAD-Block enthält Informationen, die nicht direkt im Fenster erscheinen, wie zum Beispiel den Titel. Der BODY-Block enthält den sichtbaren Inhalt, das heißt alle Textbereiche, Grafiken und so weiter. Das Gesamtdokument ist logischerweise ein HTML-Block:

```
<HTML>
 <HEAD>
 <TITLE>Dokumenttitel</TITLE>
 …weitere Meta-Information Meta-Information…
 </HEAD>
 <BODY>
 …der sichtbare Inhalt der Seite…
 </BODY>
</HTML>
```

Kommentar

Ein Kommentar ist ein Text, der von Browser und Editor ignoriert wird. Er steht zwischen `<!--` und `-->`, wie zum Beispiel `<!--dies wird vom Browser ignoriert -->`

Ein Kommentar darf weder `--` noch `>` enthalten, also auch keine Tags. In *PageMill* wird ein Kommentar durch das abgebildete Symbol gekennzeichnet.

In *PageMill* fügst Du einen Kommentar über Edit|Insert Invisible|Comment ein.

Zwischenraum

HTML 'minimiert Zwischenräume' – gibt es mehr als eine Leerstelle zwischen zwei Wörtern, werden die überzähligen ignoriert, ebenso wie Leerstellen vor oder nach einem Absatz.

Ein Zeilenwechsel wird in HTML wie ein Zwischenraum behandelt – er trennt zwei Wörter, und mehrere Zeilenwechsel hintereinander werden genau wie mehrere Leerstellen nur als normale Leerstelle zwischen zwei Wörtern wiedergegeben.

Mit anderen Worten kann man im HTML-Code etliche Zeilenwechsel und Leerstellen einsetzen, ohne daß das irgendeinen Effekt auf die Darstellung der Seite im Browser hat.

Sonderzeichen

Die Zeichen des US-ASCII-Zeichensatzes werden in einer HTML-Datei ganz normal geschrieben. Ä, Ö, Ü und andere europäische Sonderzeichen, werden mit speziellen Codes eingesetzt, die mit einem *&*-Zeichen beginnen und mit einem Semikolon enden, etwa `&auml;` für ä, `&ouml;` für ö und `&uuml;` für ü.

Der Text „Nüsse und Äpfel sind etwas Schönes" sieht in HTML so aus: `N&uuml;sse und &Auml;pfel sind etwas Sch&ouml;nes`. Wie Du siehst, entfallen bei diesen Zeichencodes `<` und `>`.

Im ISO-8859-1-Zeichensatz gibt es einen Code, eine sogenannte *Character Entity Reference*, jedes Zeichen, das nicht zum US-ASCII-Zeichensatz gehört. Außerdem gibt es die Codes `<` für <, `>` für >, `&quot;` für Anführungsstriche und `&` für *&*. Die letzteren werden benutzt, wenn diese Zeichen nicht als Teile des HTML-Codes dienen, sondern vom Leser gesehen werden sollen.

Man kann sich auch mit einer Codenummer statt eines Namens auf ein Zeichen beziehen, zum Beispiel À für das Zeichen Nr. 192 im Zeichensatz ISO-8859-1 – das verstehen aber nicht alle Browser korrekt.

Übersicht über alle HTML-Tags und -Attribute

Willst Du Dich tiefer in HTML vergraben, brauchst Du eine Übersicht über alle HTML-Tags, ihre Syntax und ihre Attribute. Eine solche Übersicht findest Du vielerorts im Web. Die *Web Design Group* hat eine ausgezeichnete und gut geschriebene Übersicht über *Wilbur*, wie der Spitzname von HTML 3.2 lautet, unter der Adresse `www.htmlhelp.com`. Diese Übersicht gibt es auch als Windows-Hilfedatei, was eigentlich ziemlich praktisch ist.

Die offiziellen Spezifikationen findest Du selbstverständlich beim W3C unter `www.w3.org`.

Eine Übersicht über HTML-Tags und Attribute findest Du im Online-Supplement des Heftes.

The NCSA Beginner's Guide to HTML ist ein guter Ausgangspunkt für weitere Experimente.
`www.ncsa.uiuc.edu/General/ Internet/WWW/HTMLPrimer.html`

HTML im Editor einfügen

Tags einfügen

In manchen Editoren kann man vom Editor aus Tags einfügen.

Tag einfügen	
PageMill	Über Edit|Insert Placeholder kannst Du willkürliches HTML einfügen.
Composer	Über Insert|HTML Tag kannst Du ein einzelnes Tag einfügen.
FrontPage	Über Insert|HTML Markup (Einfügen|HTML-Code) kannst Du „handgeschriebenes" HTML einfügen.
Claris	Läßt sich nicht machen – BUH!

Setzt Du ein Tag ein, das *PageMill* nicht versteht, deutet der Editor das mit einem kleinen Symbol mit Fragezeichen an. Markierst Du das Fragezeichen, kannst Du das entsprechende Tag im Inspector sehen und modifizieren.

Über Edit|Insert Placeholder kannst Du eigenen HTML-Code in PageMill einfügen. Es erscheint ein Symbol, das einen Kegel zeigt, und wird dieses Symbol markiert, kannst Du den Inhalt im Inspector

ändern. Du kannst sogar ein Bild einfügen, das anstelle des Kegelsymbols im Editor erscheint. Wozu das gut sein soll, weiß ich eigentlich nicht – findest Du eine Erklärung, würde ich mich über eine E-Mail freuen.

Ein Attribut im WYSIWYG-Zustand einfügen

Manche Editoren erlauben das Einfügen eines Attributs zu einem Tag, auch wenn der Editor im WYSIWYG-Zustand steht. Das Dialogfeld für die Änderung der Eigenschaften eines Elements, das Du mit der rechten Maustaste aktivierst, enthält einen Schaltknopf, mit dem Du weitere Attribute einfügst. Diese werden in das Tag eingesetzt, das dem Element entspricht, dessen Eigenschaften Du ändern willst.

Attribut im Editor einfügen	
PageMill	Läßt sich nicht machen – BUH!
Composer	Extra HTML in *Properties*
FrontPage	Advanced (Erweitert) in *Properties (Eigenschaften)*
Claris	Extra HTML auf dem Advanced-Registerblatt im Object Editor

Meta-Information zu einem Dokument

Mit dem **meta**-Tag kannst Du Metainformationen in ein Dokument einfügen, das heißt Informationen, die dieses Dokument beschreiben. Zum Beispiel kannst Du einige Kernbegriffe zum Dokument angeben, die dann von Such- und Indexierungsmaschinen benutzt werden können.

Das **Meta**-Tag gehört in den Kopf des Dokuments und kann folgendermaßen aussehen:

```
<META
 NAME="keywords"
 CONTENT="sex, l&uuml;gen, video">
```

Im **Name** gibst Du an, was Du beschreibst, und **Content** ist der eigentliche Inhalt/die Beschreibung.

Ein Meta-Tag einfügen	
PageMill	Ein **Meta**-Tag muß von Hand über Edit|HTML source eingegeben werden. Vergiß nicht, es in die Head-Markierung zu setzen.
Composer	Page Properties, Registerblatt META Tags
FrontPage	File|Page Properties (Datei|Seiteneigenschaften), Registerblatt Custom (Benutzerdefiniert)
Claris	Das komplette **Meta**-Tag wird eingesetzt mit Edit|Document Options, Registerblatt HTML Extras, Feld Text in Head Section

Die HTML-Spezifikationen legen nicht fest, welche **Name**-Werte benutzbar sind, jeder muß also selber nützliche Metabeschreibungen finden.

Die üblichsten Meta-Namen sind **Keywords** und **Description**, die von mehreren Suchmaschinen benutzt werden, wie zum Beispiel *AltaVista* und *Infoseek*. **Description** ist eine kurze Beschreibung der Seite:

```
<META
 NAME="Description"
 CONTENT="T. S. Eliot Forum. Gesammelte Gedichte, Biographie, Bibliographie,
Analysen sowie ein Statusleisten-Fliesstext">
```

AltaVista zeigt nun diese Beschreibung an, statt, wie sonst üblich, die ersten Textzeilen. Die meisten Editoren setzen automatisch eine 'Signatur' in ein Dokument, zum Beispiel:

```
<META NAME="Generator" CONTENT="Adobe PageMill 2.0 Win">
```

Es ist durchaus denkbar, daß irgend jemand irgendwo diese Signatur indexiert … Du kannst auch selbst die Seiten signieren, die Du erstellst:

```
<META NAME="Author" CONTENT="Matthias Claudius">
```

Style Sheets

Über Style Sheets kannst Du die typographische Gestaltung einer Webseite beeinflussen. Du hast vermutlich bereits festgestellt, daß HTML nur eine geringe Variierung des Layouts erlaubt – eigentlich muß man sozusagen mogeln, um in HTML etwas hinzukriegen, das gut aussieht. Neuerdings hat W3C aber die Standard Style Sheets entwickelt, mit deren Hilfe man Punktgröße, Zeilenhöhe, Seitenrand, Zeichensatz, Schriftschnitt, Einzug und so weiter einrichten kann. Nun kann man tatsächlich flotte Webseiten machen!

Dieser Standard ist allerdings so neu, daß er bisher nur vom Navigator 4 unterstützt wird, und in eingeschränkter Form vom MSIE 3. MSIE 4 wird diesen Standard ebenfalls unterstützen, also wird er sich sicher durchsetzen und mit der Zeit zur geltenden Richtschnur für das Layout von Webseiten werden. Darum will ich diesem Phänomen auch etwas Platz einräumen.

Style Sheets bauen auf einem einfachen Prinzip auf: Der Kopf eines Dokuments enthält einen Block, der die Regeln oder Deklarationen dafür enthält, wie die Markierungen in HTML typographisch wiederzugeben sind. Man kann zum Beispiel festlegen, daß alle Überschriften mit Helvetica 30 Punkt gezeigt werden. So trennt man Form und Inhalt: im eigentlichen HTML wird die Struktur des Textes und die Funktion seiner Elemente definiert, im Style Sheet dagegen sein typographisches Erscheinungsbild.

Das Style Sheet kann in den Kopf eines Dokuments gesetzt werden – es kann aber auch ein selbständiges Dokument sein, das mit dieser Seite und außerdem auch mit anderen Seiten verknüpft wird.

Style Sheets sind aus mehreren Gründen genial:

- Die Änderung einer einzelnen typographischen Spezifikation beeinflußt das Layout einer ganzen Seite oder gar mehrerer Seiten.
- Da typographische Tags überflüssig werden, wird das eigentliche HTML einfacher.
- Weil ihre Regeln ein Überbau über den existierenden Tags sind, kann man Style Sheets mit existierenden Seiten verknüpfen, indem man nur eine einzelne Zeile hinzufügt.
- Man kann die sinntragenden Markierungen benutzen, die für Such- und Indexierungsmaschinen wichtig sind, und dennoch ein flottes Layout erreichen.

Soweit ich weiß, werden Style Sheets bisher von keinem Editor unterstützt. Es ist aber relativ einfach, mit ihnen zu arbeiten: Du öffnest die entsprechende Seite in einem Texteditor und hast gleichzeitig ein Dokument in Deinen Browser geladen, das das Style Sheet implementiert. Jede Änderung im Style Sheet kannst Du unmittelbar testen, indem Du das Dokument speicherst und es im Browser neu lädst.

Style Sheets werden in einer simplen typographischen Sprache namens CSS (Cascading Style Sheets) geschrieben. Das sieht zum Beispiel so aus:

```
H1 { font-family: Helvetica }
```

bedeutet, daß alle Absätze im H1-Format, also die größten Überschriften, im Zeichensatz Helvetica dargestellt werden. Ein Style Sheet ist schlicht ein Regelsatz dieses Typs.

Style Sheets einfügen

Ein Style Sheet wird in den Dokumentenkopf mit einem neuen Tag namens **Style** eingefügt:

```
<HEAD>
 <TITLE>Beispiel eines Style Sheets im Dokumentenkopf</TITLE>
 <STYLE>
 <!--
 P { color: blue; font-size: 12pt }
 H1 { font-size: 24pt }
 -->
 </STYLE>
</HEAD>
```

Das Style Sheet steht in einem Kommentar-Tag, was bedeutet, daß Browser und Editoren, die auf Style Sheets nicht vorbereitet sind, sich hierdurch nicht verwirren lassen. Unterstützt ein Browser Style Sheets, ignoriert er die Kommentarzeichen im **Style**-Tag und liest das Style Sheet aus.

Ist das Style Sheet ein selbständiges Dokument, mußt Du über ein **Link**-Tag im Dokumentenkopf auf dieses Dokument verweisen:

```
<LINK REL=StyleSheet
 HREF=„derneuestil.css“
 TYPE=„text/css“>
```

Das Style Sheet wird wie eine normale Textdatei in einem Texteditor wie zum Beispiel *Wordpad* geschrieben.. Die Datei mit dem Style Sheet muß die Endung `.css` haben.

Syntax

Ein Style Sheet besteht aus einer Anzahl von Regeln. Jede Regel setzt sich zusammen aus einem *Selector* – einem oder mehreren Tags, für die die Regel gilt – und einer Schweifklammer, die eine oder mehrere Deklarationen enthält:

```
P { color: blue; font-size: 12pt }
H1 { font-size: 24pt }
```

P und H1 sind hier *Selectors*. Jede Deklaration besteht aus dem Namen einer Eigenschaft (*Property*), einem Doppelpunkt und einem Wert, auf den die Eigenschaft gesetzt wird. Gibt es mehrere Deklarationen, werden sie durch ein Semikolon getrennt.

Spezifikationen für CSS

Es gibt zahlreiche Möglichkeiten in CSS, zum Beispiel Schriftschnitt, Schriftgröße, Kapitälchen, Durchschuß, Zwischenraum, Einzug, Textgröße und -farbe, Seitenränder, Rahmentyp und Hintergrundfarbe oder -bild für alle Elemente.

Die kompletten Spezifikationen sowie eine gute Gebrauchsanweisung findest Du bei der bereits erwähnten *Web Design Group* unter der Adresse `www.htmlhelp.com`.

Layers und Absolute Placement

Der nächste Schritt für CSS heißt *Layers and absolute placement*. *Layers* bedeutet, daß man mehrere Ebenen von Grafik und Text übereinander legen kann, und *Absolute Placement* besagt, daß Du genau definierst, wo ein Text- oder Grafikelement auf dem Bildschirm angebracht wird. Die Kombination dieser Möglichkeiten versieht HTML mit ziemlich weitreichenden Layoutfähigkeiten.

Die Spezifikationen für diese Punkte findest Du unter `www.w3.org/TR/WD-positioning.` Navigator 4 und MSIE 4 unterstützen beide diese Spezifikationen.

Style Sheets – die Zukunft

Der Mechanismus der Style Sheets ist so flexibel, daß viele verschiedene Ausformungen mit HTML-Dokumenten verknüpft werden können. Zur Zeit plant man Mechanismen, mit denen man verschiedene Style Sheets an verschiedene Situationen binden kann, zum Beispiel ein Style Sheet für die Darstellung einer Seite auf einem Taschencomputer und ein anderes für ihre Darstellung auf einem großen Farbbildschirm.

Style Sheets werden natürlich erst dann wirklich interessant, wenn sie allgemein in Browsern unterstützt werden. Das dauert aber mindestens noch ein halbes Jahr. Und dann gibt es sicher auch neuere Versionen der Editoren, die die WYSIWYG-Arbeit an Style Sheets unterstützen. Mehr zu diesem Punkt erfährst Du im Online-Supplement.

Web Style Sheets alles über Style Sheets: `www.w3.org/Style`
CSS Gallery Beispiele für Style Sheets: `www.microsoft.com/truetype/css/`

Mach Deinen Seiten Beine!

Ein routinierter Websurfer hat eine Geduldsspanne von etwa 10 Sekunden – ist die Seite dann noch nicht im Browser, geht er einfach weiter. Anfänger haben etwas mehr Geduld, das ändert sich aber schnell.

Ein normales Modem für den Hausgebrauch schafft ungefähr 28.800 Bits pro Sekunde. Das sind etwa dreieinhalb Kilobyte pro Sekunde oder 35 Kilobyte in 10 Sekunden – nicht gerade viel!

Willst Du wissen, wie viel eine Seite einschließlich Grafik füllt und wie lange es dauert, sie downzuloaden, schlägst Du das in *PageMill* unter Edit|Download Statistics und in *Claris Home Page* unter Edit|Document Statistics nach.

Das eigentliche HTML-Dokument nimmt nicht viel Raum ein, besteht es doch nur aus Text. Grafiken dagegen sind zeitraubend.

- Umfangreiche Grafiken zu Beginn einer Eingangsseite sind gefährlich. Was die Geduld des Lesers auf die Folter spannt, ist nämlich die Zeit, die vergeht, bis der *sichtbare* Teil eines Dokuments eingelesen ist. Füllt das Dokument mehr als eine Fensterhöhe, dauert es etwas, bis der Benutzer scrollt, und inzwischen können Text und Grafik am unteren Ende der Seite eingelesen werden.
- Tabellen werden erst dann gezeichnet, wenn sie komplett eingelesen sind. Liegt der gesamte Inhalt einer Seite in einer Tabelle, bleibt die Seite leer, bis alles eingelesen wurde. Je nachdem kannst Du eine Tabelle in kleinere Bruchstücke aufteilen.
- Bei Grafik empfiehlt sich mehrmalige Verwendung. Ist ein Bild erst in den Browsercache eingelesen, wird es von dort geholt, wenn es auf einer anderen Seite wieder benötigt wird. Hast Du zum Beispiel eine Leiste auf jeder Seite, deren erster Teil je nach dem Seiteninhalt anders aussieht – etwa als Überschrift – während der Rest – etwa einige Symbole – immer gleich bleibt, solltest Du diese Leistengrafik in zwei Dateien teilen, von denen nur die erste gewechselt wird.

Die Größe einer Grafik in Kilobyte hat wenig mit ihren Ausmaßen zu tun – sie hängt eher von der Effektivität der Komprimierung ab:

- JPEG-Grafiken speicherst Du mit einem Qualitätsverlust, den Du für ästhetisch noch akzeptabel hältst.
- Einfarbige Flächen füllen in der GIF-Komprimierung so gut wie nichts – hier sind es die Übergänge, die Raum einnehmen. Je einfacher eine Grafik ist, desto weniger Kilobyte nimmt sie ein – je weniger Farben, desto weniger KB. Reduzierst Du Farben in GIF-Bildern, solltest Du das Dithern vermeiden. Gepunktete Bereiche füllen wesentlich mehr als einfarbige.

Sicherheit

Eine Website muß nicht unbedingt für jedermann zugänglich sein. Du kannst auch Seiten erstellen, für die der Benutzer ein Paßwort benötigt. Das kannst Du allerdings nicht über HTML erledigen – es muß auf dem Server konfiguriert werden, und die Vorgangsweise ist bei jedem Server verschieden. Dafür ist es aber nicht besonders kompliziert. Rede also mit Deinem Provider. Paßwort-Schutz ist allerdings keine hundertprozentige Sache – ein gewiefter Hacker kann die Seiten immer noch auffangen, indem er jemanden 'abhört', der Zugang hat.

Eine wirklich sichere Kommunikation erfolgt über einen sogenannten SSL-Server. SSL steht für *Secure Socket Layer*, was bedeutet, daß alle Kommunikation zwischen Server und Benutzer in kodierter Form erfolgt. SSL wird unter anderem für Geldüberführungen und andere Transaktionen benutzt, die hohe Sicherheit erfordern. Auch hier entscheidet die örtliche Technik, ob der Server SSL unterstützt, also solltest Du mit Deinem Provider reden.

www.ich.de

Die Adresse `www.etwebhotel.de/~sofie` ist etwas kompliziert. Einfacher und cooler wäre es, wenn Du die Adresse `www.sofie.de` hättest. Dann sieht man nämlich nicht, daß Du in einem Webhotel wohnst. Allerdings benötigst Du dafür den Domänennamen `sofie` unter der deutschen Oberdomäne `de`. Für einen erklecklichen Preis kann Dein Provider die Sache für Dich erledigen. Übrigens kannst Du auch unter den internationalen Topdomänen `com`, `org` und so weiter einen Domänennamen registrieren.

Eigentlich gibt es keinen technischen Grund für einen eigenen Domänennamen, man erhält eben nur eine einfachere Adresse, die obendrein professionell aussieht. Ob eine Website nun einen eigenen Domänennamen hat oder nicht, über Suchmaschinen wie Yahoo oder DINO wird sie auf die gleiche Weise gefunden.

Übrigens läuft ein FTP-Transfer auch bei eigenem Domänennamen gleich ab.

Dritter Teil: Interaktivität und Multimedia

Objekte

Neben reinem Text sind Objekte wie Animationen, Java-Applets, Film(e), Musik und so weiter, gut geeignet, der Website den letzten Schliff zu geben. Grafiken sind - im technischen Sinne -ebenfalls Objekte, obwohl man sie im allgemeinen nicht so auffaßt. Objekte haben eine Reihe von Gemeinsamkeiten:

- Ein Objekt ist nicht Bestandteil der eigentlichen HTML-Datei. Es besteht vielmehr aus Daten in einer oder mehreren Dateien, auf die im HTML-Code mit einem speziellen Tag verwiesen wird.
- Ein Objekt nimmt auf der Webseite einen viereckigen Raum ein – abgesehen von Objekten für das Musik-Plug-In, die unsichtbar sein können.
- Ein Objekt erfordert, daß der Browser damit umgehen kann. Manche Objekttypen, wie zum Beispiel GIF- und JPEG-Grafiken, werden von allen Browsern unterstützt, andere wie Java oder ActiveX von manchen, und wieder andere, wie die Plug-In-Objekte, erfordern die Erweiterung des Browsers mit einem Plug-In, einem Hilfsprogramm, das die Daten verarbeiten kann.
- Die meisten Objekte reagieren auf Parameter, die zusammen mit dem Tag ins HTML eingesetzt werden, der das Objekt einfügt. Zum Beispiel versteht ein Ton-Plug-In den Parameter **loop=true**. Es hängt von dem einzelnen Objekt ab, auf welche Parameter es reagiert.

Objekte werden mit verschiedenen Tags eingefügt: **img** für Grafik, **applet** für Java, **embed** für Plug-Ins. Alle haben in etwa dieselben Eigenschaften – den URL zur Datei, Breite, Höhe und so weiter – und werden genau wie Grafiken ins Layout eingefügt: sie folgen dem Textfluß, können rechts- oder linksbündig stehen, in Tabellen eingefügt werden und so weiter.

W3C hat ein **object**-Tag entwickelt, das die Funktionalität von **img**, **applet**, **embed** in sich vereint und außerdem für andere Zwecke benutzt werden kann. Bisher unterstützen allerdings nur die neuesten Browser dieses Tag. ActiveX-Komponenten werden mit dem **object**-Tag eingesetzt.

Ein Objekt einfügen	
PageMill	File|Place. *PageMill* erkennt die Art des Objekts an seiner Dateiendung.
Composer	unterstützt abgesehen von Grafik nicht den Einsatz von Objekten.
FrontPage	Insert|Other Component|Plug-in / Java Applet / ActiveX Component (Einfügen|Andere Komponenten|Plug-In / Java-Applet / ActiveX Steuerelement)
Claris	Insert Plug-in, Insert Applet

GIF-Animationen

Die meisten Animationen auf Webseiten sind *GIF-Animationen*. Hier handelt es sich um eine Erweiterung des GIF-Grafikformats, die es ermöglicht, mehrere Bilder in einer einzelnen GIF-Datei zu sammeln und die Wiedergabegeschwindigkeit und andere Parameter festzulegen. Eine GIF-Animation hat wie andere GIF-Dateien die Endung `.gif` und wird wie ein normales Bild in die Webseite eingefügt. Die Wiedergabe von GIF-Animationen erfordert kein Plug-In oder Hilfsprogramm. Ältere Browser können die Animation nicht wiedergeben, sondern zeigen nur das erste Bild – genau wie vermutlich auch der Editor, wenn Du eine Animation einsetzt.

Eine GIF-Animation kann in mehreren Programmen erstellt werden, zum Beispiel *GifBuilder* für Macintosh (`iawww.epfl.ch/Staff/Yves.Piguet/clip2gif-home/GifBuilder.html`) oder
Gif Construction Set für PC (`www.mindworkshop.com/alchemy/ gifcon.html`).

Diese Programme arbeiten so, daß mehrere einzelne Bilder gesammelt und in einer GIF-Animationsdatei vereint werden. Man kann festlegen, wie schnell die Bilder wechseln, ob die Animation ein einziges Mal läuft oder immer wieder, und dergleichen mehr.

Der Browser aktiviert eine GIF-Animation schon während des Einlesens, die Wiedergabegeschwindigkeit beruht also auf der Einlesegeschwindigkeit. Wiederholt sich die Animation, laufen die Wiederholungen schneller, weil jetzt die gesamte Datei eingelesen ist.

Plug-Ins

Plug-Ins sind Hilfsprogramme, die dem Browser die Verarbeitung neuer Dokument- oder Medienformate ermöglichen. In *Start ins Internet* im Abschnitt *Plug-Ins* erfährst Du mehr über die Installation und Benutzung von Plug-Ins.

Du integrierst ein Plug-In-Objekt, das heißt eine Datei, die von einem Plug-In wiedergegeben wird, auf ähnliche Weise wie eine Grafik in Deine Seite. Das Dateiformat legt fest, mit welchem Plug-In der Browser das Objekt wiedergibt. Es ist also nicht notwendig, speziell ein anzuwendendes Plug-In-Programm zu benennen.

Natürlich kannst Du nie sicher sein, ob der Benutzer das Plug-In tatsächlich downloadet und installiert. Außerdem stehen nicht alle Plug-Ins für alle Betriebssysteme zur Verfügung.

Plug-ins für Ton in den Formaten MIDI, WAV und AIFF und für Film im QuickTime- oder AVI-Format werden standardgemäß mit dem Navigator geliefert. Darüber hinaus gibt es zahllose Plug-Ins für ein Unzahl an Dateiformaten. Die populärsten sind VRML (3D-Welten), Shockwave (Animation und Multimedia) und Realaudio (Realtime-Ton, also Internet-Radio).

Das Schwierige an der Sache ist natürlich die Erstellung der Datei, die vom Plug-In wiedergegeben werden soll. Standardformate wie MIDI, VRML und Quicktime lassen sich in vielen verschiedenen Programmen erstellen. Andere Formate wie etwa Shockwave Flash müssen in einem speziellen Programm entwickelt werden, das ebenfalls vom Hersteller des Plug-Ins vertrieben wird. Wieder andere Plug-Ins, wie zum Beispiel Realaudio, erfordern eine Erweiterung auf dem Server, um Ton flüssig, also 'streaming' zu überführen. Im allgemeinen sind die Plug-Ins kostenlos erhältlich, während die Produzenten ihr Geld an den Entwicklungsprogrammen oder den Servererweiterungen verdienen.

Was Plug-Ins angeht, habe ich folgende Ratschläge:

- Benutze sie nur mit gutem Grund. Besteht tatsächlich ein vernünftiger Grund, Information in einem Plug-In-Format zu präsentieren, macht sich die Zielgruppe vermutlich auch die Mühe, das entsprechende Plug-In downzuloaden und zu installieren.
- Setze einen Link zu dem Ort, wo das Plug-In erhältlich ist.
- Benutze die üblichsten Plug-In-Formate und vermeide ungewöhnliche oder seltene Formate, die vielleicht nur für eine Plattform existieren.

Weiteres über Plug-Ins erfährst Du im Online-Supplement dieses Heftes.

Java-Applets

Applets sind kleine Programme in der Programmiersprache *Java*, die in eine Webseite integriert sind. Die Programmiersprache Java wurde speziell für Programme entwickelt, die über ein Netzwerk heruntergeladen werden. Solche Programme zeichnen sich durch ihre Plattformunabhängigkeit aus: ein und dasselbe Java-Programm kann unmodifiziert unter Windows, Mac, UNIX, OS/2, Amiga und so weiter laufen.

Java hat mehrere Sicherheitsmaßnahmen integriert, dank derer ein Applet keinen Virus plazieren, keine Informationen aus dem Computer beziehen oder andere gefährliche Dinge tun kann, es sei denn, Du läßt das selber ausdrücklich zu. Java wurde von Sun Microsystems entwickelt.

Applets werden vor allem für kleinere Dinge wie Animationen, kleine Spiele und dergleichen benutzt – oder aber als Benutzeroberfläche für ein Programm auf einem Server. Java läßt sich aber auch für eigentliche Programme verwenden – so setzt man zum Beispiel augenblicklich die Textverarbeitung Wordperfect in Java um, das neue StarOffice wurde ebenfalls tlw. im Java-Code geschrieben.

Willst Du Java-Applets erstellen, erfordert das Programmierer-Kenntnisse. Die Plazierung eines fertigen Java-Applets auf einer Webseite ist dagegen keine besonders schwierige Sache.

Java Javas offizielle Website: `java.sun.com`
Tutorial Suns Java-Lehrbuch: `java.sun.com/docs/books/tutorial`
JavaWorld Magazin für die Java-Gemeinde: `www.javaworld.com`

ActiveX-Komponenten

ActiveX-Komponenten sind kleine Programme, die ebenso wie Java-Applets in Webseiten integriert werden können. Eine ActiveX-Komponente muß allerdings speziell für die jeweilige Computerplattform wie etwa Win95 geschrieben werden, im Gegensatz zu den plattformunabhängigen Java-Applets. Dafür werden ActiveX-Komponenten direkter ins System integriert, was erweiterte Möglichkeiten – und ein entsprechend größeres Sicherheitsrisiko bedeutet.

Die ActiveX-Technologie ist im Internet nicht allzu vielseitig verwendbar, da das System nur unter Win95/NT funktioniert – und selbst dann nur, wenn man den MSIE 3 benutzt. In kleineren internen Netzwerken, etwa in einem Unternehmen, wo jedermann einen Win95-Computer benutzt, können sie aber ein leistungsfähiges Hilfsmittel sein.

ActiveX wurde von Microsoft entwickelt – im Rahmen der Strategie der Firma für die Erschaffung eines Internets, das „unter Win95/NT am besten aussieht".

Mehr über ActiveX erfährst Du im Online-Supplement des Heftes.

JavaScript

JavaScript ist eine simple Programmiersprache, die in HTML-Dokumenten benutzt werden kann. Mit ihrer Hilfe kannst Du Deine Webseiten interaktiver gestalten. Du kannst zum Beispiel

- kontrollieren, ob ein Formular ausgefüllt wurde, bevor es gesendet wird
- in der Statuszeile Lauftexte oder Mitteilungen zeigen, zum Beispiel wenn die Maus über einen Link geführt wird
- ein Bild bei Mausberührung auswechseln
- komplizierte Framesets kontrollieren, in denen Du zum Beispiel mehrere Frames gleichzeitig wechseln willst
- neue Fenster mit spezifischer Größe und gegebenenfalls ohne Menü- und Werkzeugleiste öffnen.

Außerdem kann JavaScript Mitteilungen von und zu Objekten in der Seite wie etwa Java-Applets, Plug-Ins oder dergleichen senden und empfangen.

Trotz des Namens ist JavaScript nicht mit Java verwandt. Beide Programmiersprachen wurden speziell für das Web entwickelt – in ihrer Funktion sind sie aber ziemlich unterschiedlich. Java-Applets sind Programmdateien eigenen Formats. Sie können zwar in Webseiten erscheinen, haben ansonsten aber nichts mit HTML zu tun und können auch ohne einen Webbrowser laufen. JavaScript dagegen ist Code, der unmittelbar in einer HTML-Datei erscheint und die unterschiedlichen Elemente einer Webseite manipuliert.

JavaScript ist außerdem einfacher und weniger kompliziert als Java und leichter zu erlernen.

Unmittelbar ist JavaScript eine vielversprechende Technologie – sie hat aber gewisse Probleme. Ursprünglich wurde sie von Netscape entwickelt und kontrolliert und hat sich nicht zu einem eigentlichen Standard entwickelt. Microsoft hat eine Kopie von Java namens JScript entwickelt. Der „kleinste gemeinsame Nenner", also der Teil der Sprache, der sowohl auf dem MSIE 3 als auch auf dem Navigator funktioniert, ist aber ziemlich begrenzt, und es gibt Kompatibilitätsprobleme zwischen den beiden Versionen.

Mehr über JavaScript erfährst Du im Online-Supplement des Heftes.

JavaScript 1.0 Das vollständigste Material zu JavaScript findest Du bei Netscape:
`www.netscape.com/eng/mozilla/Gold/handbook/javascript`
JavaScript 1.1 ebenfalls bei Netscape:
`www.netscape.com/eng/mozilla/3.0/handbook/javascript`
Lehrbuch über JavaScript: `www.webconn.com/java/javascript/intro/`
Inkompatiblitäten: ein wichtiger Überblick über die Unterschiede zwischen Netscapes JavaScript und Microsofts JScript: `www.webcoder.com/browsersupport`

CGI & Formulare

CGI ermöglicht ein Zusammenspiel zwischen einer Webseite und einem Programm auf dem Server. Dank dieser Technik kann der Leser einer Webseite unmittelbar Informationen an den Server senden, der seinerseits wiederum auf der Basis dieser Informationen eine neue Webseite generieren kann. CGI wird zum Beispiel für Suchen und Anmeldungen benutzt oder für Webseiten, die im Bedarfsfall generiert werden.

CGI unterscheidet sich grundsätzlich von ‘mobilem Code’ wie etwa Java, JavaScript und ActiveX, der an den Computer des Benutzers geschickt und dort ausgeführt wird. Ein CGI-Programm wird auf dem Server betrieben, und das Resultat wird an den Benutzer geschickt. Im Gegensatz zu Java, ActiveX und JavaScript benötigt der Benutzer also nicht eine ganz bestimmte Browserversion oder dergleichen. Außerdem kann ein CGI-Programm mit anderen Programmen auf dem Server, zum Beispiel Datenbanken, kombiniert werden.

Ein CGI-Programm wird über einen Link aktiviert, oder aber, indem man einen Schaltknopf in einem Formular anklickt. Das Programm generiert auf dem Server eine neue Webseite, die an den Benutzer geschickt wird. Der Benutzer selbst erlebt diesen Vorgang wie einen normalen Link.

Ein Beispiel: Du suchst etwas bei *Yahoo*, schreibst also einige Worte ins Suchfeld und drückst dann den Schaltknopf 'search'. Deine Suchworte werden an das CGI-Script übermittelt, das seine Datenbank durchsieht und eine HTML-Seite mit den gefundenen Informationen generiert, die wiederum Dir übermittelt wird.

Meist wird das eigentliche CGI-Programm von einem Programmierer erstellt, während der Webdesigner für die Überführung der Information von HTML in CGI und umgekehrt sorgt. Leider erlauben nicht alle Web-Hotels die Möglichkeit, mit CGI-Programmen zu arbeiten. Die CGI-Technik kann ein Risiko für den Server bedeuten und belastet ihn mehr als normale Webseiten. Manche Provider testen CGI-Programme auf ihre Sicherheit hin, bevor sie freigegeben werden.

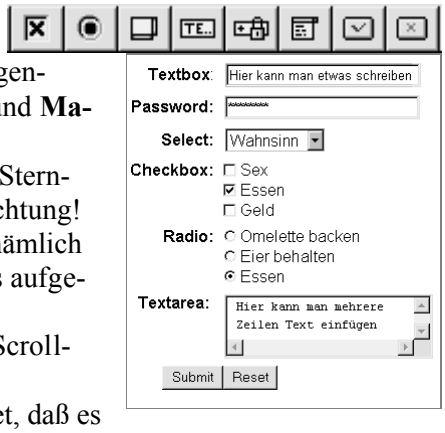
Formulare

Der Benutzer sendet seine Informationen über ein *Formular,* englisch *form,* an das CGI-Programm. Ein solches Formular besteht aus mehreren Feldern, also Textfeldern, Feldern zum Ankreuzen oder dergleichen, und einem *Submit*-Schaltknopf. Wird dieser angeklickt, werden die in diese Felder eingetragenen Informationen an das CGI-Programm übermittelt.

Alle Felder haben die Eigenschaft **Name**. Der Name eines Feldes ist wichtig – er erlaubt dem CGI-Programm die Identifikation der zugesandten Werte.

Schaltknöpfe zum Einfügen von Formularfeldern	
PageMill	Knöpfe auf der unteren Symbolleiste.
Composer	unterstützt Formulare nicht
FrontPage	View|Forms Toolbar (Ansicht|Formular-Symbolleiste)
Claris	Window|Show Forms Palette

Es gibt folgende Formularfelder:



Textbox – Texteingabefeld, das eine einzelne Zeile erlaubt. Seine Eigenschaften sind **Size**, das heißt Breite des Feldes in Anzahl Zeichen, und **Maxlength**, maximale einsetzbare Zeichenanzahl.

Password – siehe **Textbox**, nur werden die eingesetzten Zeichen als Sternchen dargestellt. Wird für die Eingabe eines Passwortes benutzt. Achtung! Möglichst nicht für wichtige Passworte benutzen – das Wort wird nämlich als Klartext verschickt, kann also mit etwas bösem Willen durchaus aufgefangen werden.

Textarea – Texteingabefeld, das mehrere Zeilen enthalten und über Scrollbars, also Laufleisten, gesteuert werden kann.

Checkbox – Feld zum Ankreuzen. Die Eigenschaft **Checked** bedeutet, daß es unmittelbar als angekreuzt dargestellt wird.

Radio – ein runder Schaltknopf, der ebenso wie ein Ankreuzfeld markiert wird. Gehört ein Radiobutton zu einer Gruppe, kann jeweils nur einer dieser Knöpfe markiert werden. Wird ein anderer gedrückt, wird der erste deaktiviert – genau wie bei einem Transistorradio, wo auch jeweils nur ein Schalter eingedrückt ist. Radiobuttons, die derselben Gruppe angehören, haben den gleichen Namen. **Checked** gibt an, ob der Button unmittelbar aktiviert wird.
Select – eine Aufstellung von wählbaren Elementen. Es gibt zwei Typen – beim einen ist nur jeweils ein Element wählbar, der andere erlaubt mehrere Möglichkeiten. Für jede Möglichkeit der Aufstellung müssen die Eigenschaften **Name** und **Value** angegeben werden.
Hidden – ein Feld, das für den Benutzer unsichtbar ist, aber dennoch Informationen enthält, die dem Server übermittelt werden. In *PageMill* wird ein solches verstecktes Feld über Edit|Insert Invisible|Hidden Field eingefügt.
Submit – der Schaltknopf, der das Formular versendet. Muß in jedem Formular erscheinen.
Reset-Schaltknopf – löscht sämtliche Eingaben im Formular.

Ein Formular als Ganzes hat zwei Eigenschaften: einmal **Action**, also den URL des CGI-Programms, an das die Information geschickt wird, zum anderen **Method**, das heißt die Methode, nach der die Information übermittelt wird – und hier handelt es sich um **Get** oder **Post**. Diese Werte werden beide vom Programmierer festgelegt.

Eigenschaften eines Formulars	
PageMill	im Inspector, Registerblatt Form
Composer	unterstützt Formulare nicht
FrontPage	Form Properties (Formulareigenschaften), Schaltknopf Settings (Einstellungen)
Claris	Edit|Document Options|Advanced

PageMill und *Claris Home Page* gehen der Einfachheit halber davon aus, daß jede Seite höchstens ein Formular enthält, obwohl die HTML-Spezifikationen eigentlich mehrere erlauben.

E-Mail-Formulare

Hast Du keinen Zugang zu CGI-Programmen, kannst Du ein Formular erstellen, dessen Inhalt unverändert an eine E-Mail-Adresse übermittelt wird, statt an ein CGI-Programm. Dazu schreibst Du `mailto:ich@meine.adresse` ins **Action**-Feld. Die **Method** ist in diesem Falle **post**. Das funktioniert allerdings nur im *Netscape Navigator*, nicht aber im *MSIE*.

GGI-Programmierung

CGI steht für *Common Gateway Interface*. Eigentlich handelt es sich hier um Angaben dazu, wie das angesprochene Programm seinen Input erhält und wie der Output aussehen soll.

Das eigentliche Programm auf dem Server kann in jeder beliebigen Programmiersprache verfaßt sein – zum Beispiel C/C++, Perl, Java oder Basic. Außerdem gibt es Systeme, über die Du auf dem Server HTML-Seiten mit JavaScript oder ähnlichen Sprachen kombinieren und dort verarbeiten kannst, bevor das Ergebnis, also die modifizierte HTML-Seite, an den Benutzer geschickt wird. Über die Scripts erhältst Du außerdem Zugang zu Standardobjekten auf dem Server wie Datenbanken. Netscapes LiveWire, Microsofts Active Server Pages und NeXTs Webobjects sind solche Systeme.

Vierter Teil: Planung und Design für eine Website

Einfacher Zugriff auf Informationen ist nicht dasselbe wie Wissen.

Terry Pratchet

Online-Hypertext ist nicht gerade ein ideales Medium. Auf einem Computerbildschirm ist ein Text nicht so leicht lesbar wie auf Papier – und je mehr man sich konzentrieren muß, um etwas zu lesen, desto weniger bringt es einem. Auf dem Bildschirm fehlt außerdem das Gefühl dafür, 'wo man eigentlich ist und wieviel noch fehlt', das man hat, wenn man ein Buch oder eine Zeitschrift direkt in der Hand hält. Und die dezentrale, chaotische Struktur im Web mit Hyperlinks in allen Richtungen macht die Sache auch nicht leichter.

Für eine Website sind daher Lesbarkeit, strukturelle Deutlichkeit und klare Kommunikation wichtiger als smartes Design, *Corporate Identity* und so weiter – anders als im gedruckten Medium, wo ein weniger lesefreundliches Layout durchaus akzeptabel ist, wenn es die Sache spannender macht.

Das heißt natürlich keinesfalls, daß man für eine Website auf flottes Layout und spannende Typographie verzichten sollte – nur, daß das weniger wichtig ist als der Inhalt, der vermittelt werden soll, es sei denn, das *Design* ist Deine Botschaft. Für die Eingangsseite in Deine Website ist das spannende Design und/oder die deutliche *Corporate Identity* natürlich ein wichtiger Bestandteil der 'Botschaft'.

Das Medium hat also seine Grenzen. Um so wichtiger ist die gründliche Planung, bevor Du Dich an die eigentlichen Seiten machst. Fängst Du mit ihnen an, verfängst Du Dich leicht in technischen Möglichkeiten und Begrenzungen und verlierst den klaren Blick auf Deine Absichten in der Website.

Arbeitsplan

Es gibt einen natürlichen Arbeitsweg für die Entwicklung einer Website:

1) Definiere Zweck, Zielgruppe und Inhalt.
2) Plane Struktur und Organisation des Inhalts und definiere eine einfache Art zu navigieren.
3) Erstelle Vorlagen für Seitenlayout und Typographie.
4) Erstelle die einzelnen Webseiten im Webeditor.
5) Teste Deine Website.

Nach meiner Erfahrung ist es *nicht* empfehlenswert, die einzelnen Schritte miteinander zu mischen. Liegt die Hauptstruktur noch nicht fest, ist es unpraktisch, mit dem Layout anzufangen, und bist Du Dir nicht über den Zweck Deiner Site im klaren, solltest Du nicht mit der Arbeit an den einzelnen Seiten im Editor anfangen.

Hältst Du Dich an die übergeordnete Struktur, kannst Du laufend bestehende Seiten modifizieren und neue hinzufügen. Willst Du aber das Grunddesign oder den Zweck Deiner Site ändern, solltest Du das gründlich planen und alle drei Punkte durchdenken, bevor Du Dich an die eigentlichen Änderungen in den Seiten machst. Je besser Deine Vorbereitungen sind, desto einfacher wird letztlich die Arbeit an den eigentlichen Seiten und die Aktualisierung der gesamten Site. Glaub's mir – ich spreche aus Erfahrung.

Und bevor Du den Start Deiner Website ausposaunst, solltest Du an Hofstadters Gesetz denken: *Es dauert immer länger als man glaubt – auch wenn man mit Hofstadters Gesetz rechnet.* (Douglas R. Hofstadter)

Während des Planens mußt Du alles über die technischen Finessen in Deiner Website vergessen und nur an eines denken: Was willst Du mit Deiner Site?

Immer mit der Ruhe

Hast Du noch keine Website gemacht, braucht Dich das nicht in Panik zu versetzen. Viele Leute und besonders Unternehmen glauben, daß Informationstechnologie wie ein Zug ist, den man nicht verpassen darf – nur schnell einsteigen, bevor er losfährt! In Wirklichkeit ist Informationstechnologie natürlich ein Produkt, mit dem viele Leute Geld verdienen. So lange jemand also daran interessiert ist, wird es auch erhältlich sein.

Im Grunde ist das Internet ein Ort, wo es wenig bringt, ganz vorne dabei zu sein. Die Dinge ändern sich hier so schnell, daß es nach einem Jahr gar keine Rolle spielt, ob Du dabei warst – alles ist ohnehin ganz anders.

Definiere Deinen Zweck

Du mußt wissen, was Du eigentlich mit Deiner Website willst. Es kann Dir zum Beispiel um folgende Dinge gehen:

- Image – Freunden oder Geschäftspartnern die Existenz der Website mitteilen, den URL auf die Visitenkarte schreiben und so weiter.
- Online-Visitenkarte – über die Website finden die Leute schnell Deinen Namen oder den Deiner Firma, die Adresse, Telefonnummer, E-Mail-Adresse und so weiter.
- Feedback – über die Website ist es einfach, mit Dir Verbindung aufzunehmen und Dir etwas mitzuteilen.

- Kommunikation – eine Gruppe von Leuten kann Deine Website als Forum für den Austausch über ein Thema benutzen.
- Information – Du hast Informationen, die Du vermitteln willst und die für andere interessant sind. Das kann zum Beispiel ein Newsletter für Deine Organisation sein oder eine wissenschaftliche Arbeit, die für andere interessant ist. Diese Form von Websites ist vielleicht die meistverbreitete.
- Reklame – Verkauf eines Produkts. Dieser Typ von Websites ist wohl der schwierigste, was das Funktionieren angeht.
- Spaß an der Freud' – es macht Dir einfach Spaß, eine Website herzustellen.

Die Vorschläge in den folgenden Abschnitten gehen davon aus, daß es Dir bei Deiner Website um Kommunikation geht.

Zielgruppe

Welche Leute werden sich für Deine Website interessieren? Und was kannst Du tun, damit die richtigen Leute kommen? Vergiß nicht, daß es 'den Leuten' im allgemeinen ziemlich egal ist, was Du ihnen erzählen oder verkaufen oder von ihnen haben willst – es interessiert sie viel mehr, ob Du etwas hast, das sie selber gerne hätten. Kannst Du ihnen nichts bieten, lohnt sich das Web nicht. Reklamebroschüren, Zeitungsannoncen, Fernsehreklamen und so weiter sind lauter Möglichkeiten, den Leuten von Dingen zu erzählen, von denen sie 'eigentlich nichts wissen wollen' – das Web dagegen erfordert, daß die Leute sich bewußt dafür entscheiden, Deine Seite zu besuchen.

Je genauer und spezieller Deine Zielgruppe ist, desto größer Deine Möglichkeit, sie anzusprechen. Baust Du eine Website auf, die alle und jeden ansprechen soll, kämpfst Du gegen Disney, MTV, Louvre, Playboy und so weiter. Richtest Du sie an die Fabrikanten von Dosenöffnern oder Sammler von zweifarbigen Fünf-Pfennig-Briefmarken und bietest ihnen etwas an, das sie brauchen können, ist Dein Erfolg sicher.

Information kontra Informationen

Denke daran, daß wenig Information mehr bringt als viele Informationen. Je mehr Informationen, desto längere Zeit vergeht, bis der Leser sie sortiert und die wichtigen Punkte gefunden hat. Je besser Du Deine Informationen im voraus filterst, desto wertvoller sind sie für den Benutzer.

Online lesen oder drucken?

Denke darüber nach, ob Deine Leser die angebotenen Informationen eigentlich online lesen. Größere zusammenhängende Texte wie etwa ein Lehrbuch oder einen Essay werden die meisten Leute downloaden und offline lesen oder ausdrucken. Vielleicht wäre RTF oder PDF ein besseres Format für Deine Arbeit. Auch in diesem Fall solltest Du allerdings Dein Material online beschreiben – immerhin soll das Interesse des Lesers geweckt werden, und nicht jedermann downloadet einen größeren Text. Dafür liest man aber einen Offline-Text oder ein gedrucktes Exemplar intensiver und mit größerem Gewinn.

Das Grundbild

Der Inhalt in Deiner Site muß eine Struktur haben. Dank Hypertext kann man zwar Texte und Dokumente in einem großen Spinngewebe kreuz und quer miteinander verbinden – das heißt aber keineswegs, daß man jetzt nicht mehr an Gliederung und Reihenfolge der Informationen zu denken braucht. Ganz im Gegenteil war es noch nie so wichtig, Informationen klar, logisch und funktionell zu strukturieren.

Der Benutzer muß sich in Deiner Website orientieren und das finden können, wonach er sucht. Die Site muß aufgrund eines klaren Fundaments oder einer Idee aufgebaut werden, die der Benutzer unmittelbar begreift – muß er zuerst minutenlang überlegen, wie die Site eigentlich aufgebaut ist, geht sein Interesse schnell flöten. Das einfachste ist, die Website nach einem allgemein bekannten Grundbild zu strukturieren. Ein paar Beispiele:

Zeitung: Die Site besteht aus einer Reihe von Artikeln, die nicht in bestimmter Reihenfolge gelesen werden müssen – einmal abgesehen davon, daß Artikel über ein bestimmtes Thema sich teilweise auf frühere Artikel zum selben Thema stützen. Die Artikel werden nach Wichtigkeit und Aktualität angeordnet – neuere Artikel im oberen Teil – größere Überschriften weiter vorne.

Lexikon: Die Site enthält Faktenwissen in kleineren Bruchstücken. Die Artikel sind nicht in bestimmter Reihenfolge miteinander verbunden, sondern verweisen gegenseitig aufeinander.

Buch: Die Site ist ein Buch, dessen Seiten der Reihe nach hintereinander gelesen werden sollen. Gegebenenfalls können ein Inhaltsverzeichnis und ein Index den Einstieg an verschiedenen Stellen des Textes ermöglichen.

Wie Du siehst, stützen sich diese drei Grundbilder auf die fundamentale Vorstellung, daß *Webseiten wie Papierseiten funktionieren.* Diese Vorstellung durchdringt unsere gesamte Auffassung vom aktuellen Web –

weswegen sie kein schlechter Ausgangspunkt ist. Vor diesem Hintergrund werden Begriffe wie 'letzte Seite', 'nächste Seite' und 'Homepage' unmittelbar einleuchten. Diese Grundvorstellung eignet sich vor allem dafür, Text zu organisieren. Je mehr Interaktivität und Kommunikation Deine Website enthält, desto schiefer wird sie aber.

Die Grundvorstellung vom **Raum** ist ebenso umfassend wie die der Papierseiten. Die Website kann wie ein Haus mit mehreren Räumen sein, durch die man sich bewegt, ein Geschäft, an dessen Regalen man entlangschlendert, eine Stadt, deren Orte und Häuser man besucht.

Du kannst auch an eine **Baumstruktur** denken – hier bewegt man sich durch mehrere Ebenen von Menüs und Untermenüs und zielt immer genauer, bis man endlich findet, was man sucht. Die Baumstruktur ist unter Computerbenutzern ziemlich beliebt, macht sie es doch möglich, große Informationsmengen zu strukturieren und zu überblicken, wenn man nur systematisch denkt.

Dabei besteht aber die Gefahr, daß der Benutzer die Struktur des Inhalts nicht unbedingt genau so auffaßt oder begreift wie der Designer. So ist es zum Beispiel riskant, eine Website an der internen Struktur eines Unternehmens zu orientieren oder ein Lehrbuch nach der inneren Logik des Themas zu strukturieren – einer Logik, die die Zielgruppe des Lehrbuchs vermutlich noch nicht überschauen kann.

Je mehr Überblick der Leser zu einem Thema hat, desto weniger linear sucht er sich seine Informationen. Ein Anfänger liest ein Lehrbuch von Anfang bis Ende, der Experte dagegen bohrt in größeren Informationsmengen nach genau definierten Informationen.

In einem ungenauen Material oder einem Material, zu dem der Überblick fehlt, findet man vielfach die gesuchten Informationen in einer **spinnwebhaften** Struktur – jeder Abschnitt enthält zahlreiche assoziative Links zu verwandten Abschnitten. Hier sucht der Benutzer seine Informationen eher 'intuitiv'.

Andere Strukturen

Allmählich verbreiten sich im Internet neue Strukturen für die Organisation von Informationen. In einer *FAQ* werden Informationen in der Form von Frage und Antwort geboten, was sich gut für Bereiche eignet, wo Kommunikation und Meinungsaustausch herrschen.

Threading eignet sich sehr gut für Diskussionen – die Mitteilungen ordnen sich nach einer flachen verzweigten Struktur, nach ihrem inneren Zusammenhang. Das ist eine sehr lebendige, aber auch ziemlich chaotische Organisationsform für Informationen. Richtig nützlich wird sie, wenn es außerdem eine Übersicht dazu gibt, welche Themen wo angesprochen werden.

Chronologisch/'what's new' – viele Sites enthalten eine Liste über Änderungen und Ergänzungen zu einer Website in umgekehrter chronologischer Reihenfolge. So stellt der Besucher schnell fest, was hier neu ist.

Beispiele

Man kann mehrere Grundbilder miteinander kombinieren. Microsofts *Sitebuilder Workshop* (`www.microsoft.com/workshop`) benutzt das grundlegende Bild eines Planeten für jede der sechs Hauptgruppen. Das verdeutlicht, daß es hier um verschiedene Welten geht, die nicht in bestimmter Reihenfolge zueinander gehören. Der einzelne 'Planet' baut sich dann wie ein Magazin auf, mit Artikeln, Interviews, Nachrichten und so weiter. Die Kombination von mehreren Grundbildern oder Strukturen erlaubt weitere Organisierungsmöglichkeiten – andererseits ist sie für Außenstehende nicht so unmittelbar einleuchtend.

Ein Grundbild kann sehr subtil und dennoch wirkungsvoll sein. Yahoo (`www.yahoo.com`) stellt sich unmittelbar schlicht als eine Reihe von Seiten mit Links dar. Jede Seite gibt aber zuoberst den Suchpfad zur aktuellen Seite an, zum Beispiel 'Top|Computers and Internet|World Wide Web|Page Design and Layout'. Jede Überschrift ist ein Link auf die entsprechende Seite. So wird ein Weg mit allen seinen Phasen angedeutet. Das @-Zeichen verdeutlicht in Links die Tatsache, daß man von hier aus eine Gruppe in einer anderen Verzweigung der Baumstruktur anspricht.

Du wirst feststellen, daß die Bilder 'Baum' und 'Suchpfad' nicht ausdrücklich dargestellt werden, die Struktur veranlaßt mich aber dennoch, eine Site nach diesem Grundbild zu verstehen, und dadurch fällt mir die Orientierung leichter.

Grundbilder und ihre Grenzen

Ein Grundbild ist nicht nur ein Haken, an dem man eine Struktur aufhängt – es ist auch ein Werkzeug, mit dem man das Material aus neuem Blickwinkel sieht. Stell Dir Dein Material aus dem Blickwinkel verschiedener Grundbilder vor – Dir kommen sicher viele neue Ideen.

Vermeide Grundbilder, die sich nicht mit allgemein anerkannten Bildern vertragen. Zum Beispiel wäre es in einem Buch ziemlich logisch, 'abwärts' in den nächsten Artikel zu gehen und 'nach oben' zum vorigen, man hat aber allgemein akzeptiert, daß der nächste Artikel

'vorwärts' steht, also nach rechts, und der vorige 'rückwärts', also nach links.

Ein Grundbild darf natürlich nicht so bindend werden, daß es die Orientierung und Navigierung in der Website erschwert. Baut sich die Site zum Beispiel aus 'Räumen' in einem 'Haus' auf, müßte man logischerweise durch einen weiteren Raum, den Korridor, gehen, um von einem Raum zum anderen zu kommen. Das ist aber komplizierter als nötig. Statt dessen könnte man ein Menü einrichten, mit dem man von Raum zu Raum springt. So eine 'Fernbedienung' wäre 'Zauberei', weil sie im Verhältnis zum Grundbild unrealistisch erscheint.

Entsprechend ist auch eine Suchfunktion in einem Buch 'reine Magie', wenn man sie am Grundbild mißt – es sei denn, das Buch ist ein Lexikon.

Vielfach sind es jedoch gerade diese 'magischen Effekte', die die neuen Seiten im Webmedium nutzen. Da sie aber nicht unmittelbar einleuchten, wenn man sie am Grundbild mißt, erfordern sie meist eine Erläuterung. Eine Website, die auf dem Illustrierten-Grundbild aufbaut und ganz auf magische Effekte verzichtet, könnte genauso gut als richtige Illustrierte herausgegeben werden.

Suche

Ein Suchmechanismus, der einzelne Worte in der Website nachschlagen kann, setzt den Zugang zu einem CGI-Programm voraus. Eine Alternative wäre ein umständlich selbst hergestellter Index oder ein Index, der von einem Hilfsprogramm generiert wird.

Inhaltsverzeichnis

Ein Inhaltsverzeichnis ist unbedingt notwendig. Bei kleineren Websites kann es auf der Eingangsseite erscheinen, andernfalls ist es eine selbständige Seite, deren Hauptüberschriften auf der Eingangsseite wiederholt werden.

Eine kurze Beschreibung des Inhalts einer Seite unter ihrer Überschrift kann sehr hilfreich sein.

Das Inhaltsverzeichnis muß nicht unbedingt textlich aufgebaut werden – es kann auch grafisch als Karte oder Baumstruktur dargestellt werden, wenn das einen besseren Überblick über die Site vermittelt.

Kommunikation

Kommunikation in mehreren Richtungen gehört zu den spannendsten Möglichkeiten im Internet – im Gegensatz zu der einfachen Mitteilung, die eine normale Website ist. Kommunikation läßt sich aufteilen in *Echtzeit-Kommunikation* (Chat, Telefonie, Videokonferenzen) und *verspätete Kommunikation* (E-Mail, Newsgroups).

Echtzeitkommunikation (Real-Time-Communication) ist ziemlich kompliziert und stellt große Anforderungen an Server und Client, weswegen ich nicht näher darauf eingehen will. 'Verspätete Kommunikation' dagegen liegt durchaus im Möglichen für die meisten Websites und ist außerdem vielfach wesentlich interessanter.

Die einfachste Art der Diskussion ist hier, daß die Leser E-Mails an die Website schicken, die der Webmaster empfängt, sortiert und auf die Seiten legt. Für den Webmaster ist das ziemlich viel Arbeit, besonders wenn viele Briefe kommen, dafür erfordert es aber keinen CGI-Zugang.

Andererseits gibt es CGI-Programme, die E-Mail automatisch sortieren und die einzelnen Briefe in Webseiten integrieren. Manche Websites haben eine Seite mit einem praktischen Formular, in das man seinen Schriebs einfügt. Die meisten ziehen aber vermutlich normale E-Mail vor, da man die ohnehin kennt.

Ein gutes Beispiel für ein lebendiges Diskussionsmilieu im Web ist Hotwireds Threads: `threads.hotwired.com/threads/`

Browserwatch/news ist ein Beispiel für eine andere Weise, eine kommunikationsorientierte Website zu betreiben. Hier bearbeitet und kommentiert der Webredakteur alle Beiträge. `browser-watch.internet.com/news.html`

Die Länge einer Seite

Wie lang darf eine Seite sein? Nun, sie muß eine 'Informationseinheit' enthalten, also so viel Information, wie der Leser Deiner Meinung nach verdauen kann, ohne abzuspringen. Hast Du einen durch und durch linearen Text von 86 Seiten, kann der durchaus auf einer Webseite stehen – nur wäre es vielleicht besser, ihn als RTF zu formatieren, so daß der Leser ihn downloaden kann.

Artikel, Rezensionen und dergleichen sind vielfach Informationseinheiten – man liest den einzelnen Artikel linear, es steht aber nicht fest, in welcher Reihenfolge die Artikel zu lesen sind.

Navigation und Oberfläche

Eine Webseite muß dem Benutzer klarmachen, wo er sich befindet, sie muß aber auch Schaltknöpfe oder Links enthalten, die ihn weiterbringen. Diese 'Benutzeroberfläche' ist nicht etwa nur eine 'Schale' um die Website – sie sollte eine unmittelbare Funktion ihrer Struktur sein und sich an demselben Grundbild ausrichten wie sie.

Gebundene Links

Gebundene Links sind Links zu bestimmten festen Seiten – zum Beispiel zur Eingangsseite, zu einer Suchseite oder zu einer Feedback-Seite. Jede Seite sollte einen Link zur Startseite enthalten, falls der Benutzer mitten in die Struktur hereinplatzt, etwa durch einen Sprung von einem Suchmechanismus.

Relative Links

Hier geht es um Links mit einem Titel, der sich aus der Position der Seite in der Struktur versteht, zum Beispiel *vorwärts*, *rückwärts* oder *aufwärts*. Das ist praktisch, weil es sich an ein wohlbekanntes Bild hält, sollte aber niemals für sich stehen. Folgende Punkte sind unumgänglich:

- die Angabe der relativen Position. Beim Schaltknopf 'aufwärts' sollte man wissen, wie tief man in die Struktur eingedrungen ist.
- die Angabe des eigentlichen Zielorts, zum Beispiel „letzte Seite: *Ezra Pound*- Verbindung | nächste Seite: Kommentare zu *The Waste Land*"

Relative Links solltest Du nur benutzen, wenn es tatsächlich einen eindeutigen Verlauf oder eine Reihenfolge im Inhalt der einzelnen Seiten gibt. In einer Artikelsammlung oder einem Nachschlagewerk können Ausdrücke wie Vor und Zurück nur verwirren.

Parallelnavigation

Vielfach baut sich eine Website als eine Reihe von nebengeordneten Seiten oder Hauptgruppen auf. In solchen Fällen stellt man Links zu den einzelnen Hauptpunkten als waagerechte Leiste zuoberst auf der Seite, als Registerblätter in einer Kartei oder als 'Sidebar' am Rand des Bildschirms dar. Achte darauf, daß der Leser immer weiß, in welcher Gruppe oder Seite er steht – und daß er weiß, ob er auf dem markierten Punkt steht oder auf einem Unterpunkt zu diesem.

Gleichgeordnete Punkte müssen nicht unbedingt reihenweise nebeneinander stehen. `www.naestvednet.dk`, die Site der dänischen Stadt Næstved, benutzt zum Beispiel eine kleine Zeichnung der Stadt aus der Vogelperspektive, auf der wichtige Gebäude als Links zur entsprechenden Hauptgruppe in der Website dienen.

Kontextbestimmte Links

In einer eher ungenauen nonlinearen Struktur wie zum Beispiel einem Nachrichtendienst enthält jeder Artikel im allgemeinen einige Links zu Artikeln über dasselbe Thema oder verwandte Themen. Solche Seiten können eine im Verhältnis zur Artikellänge große Anzahl von Links enthalten. Auf solchen Sites wird man auch viel Zeit auf das Suchen und Klicken verwenden – wie man in einer Zeitung ja auch viel Zeit damit verbringt, die Überschriften zu durchfliegen.

`www.news.com` ist ein Beispiel dafür. Zu jedem Artikel gehören drei Navigationsfelder: eins nach rechts mit allen festen Rubriken des Nachrichtendienstes, eins nach links zu den neuesten Überschriften und eins nach unten zu Artikeln, die sich mit diesem Thema befassen. Außerdem können Links zu relevanten Artikeln im eigentlichen Artikel als symbolhaftes Bild mit einer Überschrift und dem Text *Related Story* auftauchen.

Symbole

Ein Symbol sollte grundsätzlich mit einem erklärenden Text versehen werden. Meist verrät das Motiv eines Symbols wenig von seiner Funktion – erst wenn man seine Bedeutung kennt, kann man es leichter 'lesen' als Text.

Sei vorsichtig mit undeutlichen oder obskuren Symbolen oder Namen. Es gibt oft Symbole mit Titeln wie 'Pavillon, 'Zentrum', 'Extra' oder andere undurchsichtige Begriffe, die für den Designer vielleicht sinnvoll sind, dem Benutzer aber nichts verraten.

Navigationswerkzeuge müssen in der gesamten Website gleich funktionieren. Verweist ein Symbol auf eine bestimmte Seite, muß es natürlich auch im Kopf dieser Seite erscheinen.

Beispiele

`www.adobe.com` ist ein Beispiel für eine Navigationsleiste mit viel zu viel Design, bei der man den Überblick verliert: Zu jedem Produkt gibt es Menüpunkte wie *Overview*, *Details*, *At work*, *Getting help* – aber was heißt *At work* eigentlich in diesem Zusammenhang? Und was ist der Unterschied zwischen *Support* und *Getting help*? Oder der zwischen *Solutions*, *Services* und *Products*? Wohin führt *Up*, wenn die Leiste den Eindruck einer flachen Struktur vermittelt? Und was bedeutet *Studio*?

Außerdem deutet die Leiste nirgendwo an, wo man sich eigentlich befindet.

Natürlich ist diese Struktur durchaus durchdacht und logisch, nur vermittelt die Navigationsleiste das nicht. Außerdem hat man problematischerweise eine sehr strenge Struktur gewählt, in die alle Produkte eingepaßt werden müssen.

Sei vorsichtig mit allzu strengen Strukturen! Belaste eine Vorlage nicht stärker, als sie es tragen kann.

Hypertext

Links, die unmittelbar im Text erscheinen, müssen als Leseweg verständlich sein. Der Leser muß den Link verfolgen und an der neuen Stelle weiterlesen und -blättern können. Der Link darf also nicht mitten in einen Zusammenhang springen, der nicht zu verstehen ist, ohne daß man in der neuen Struktur rückwärts geht. Natürlich sollte man auch nicht etwa einen Link zu Informationen erstellen, die nicht verständlich sind, wenn man nicht im Ursprungstext weiterliest. Vergiß nicht, daß jeder Link den Besucher so weit entführen kann, daß er vielleicht nie auf Deine Site zurückkehrt.

Links müssen in den natürlichen Verlauf des Textes integriert sein und sollten keine Sonderbehandlung genießen – sie werden ohnehin ziemlich hervorgehoben. Schreibe also nicht 'Hier klikken, um dorthin zu kommen…'. Verweise niemals auf einen Link mit 'unten auf der Seite gibt es einen Link zu…' oder dergleichen.

Hinweise

Links, die relevant sind, aber nicht unmittelbar in den Text integriert werden können, solltest Du am unteren Ende der Seite oder wie eine Literaturliste auf einer eigenen Seite anbringen.

Unter jedem Link solltest Du kurz beschreiben, wohin er führt.

Listen von 'interessanten Orten im Web' sind nur sinnvoll, wenn sie das Ergebnis einer längeren kritischen Sortierung sind und wenn Du angibst, *warum* der jeweilige Link interessant ist und wozu er nutzt.

Textbearbeitung

Die wichtigsten Navigationswerkzeuge in größeren Texten – ob das nun Hypertext ist oder nicht – sind nicht etwa Sidebars oder Symbole, sondern Überschriften.

Definiere die Anzahl der Überschriftsebenen, mit denen Du arbeiten willst. Selten brauchst Du mehr als zwei oder drei. Manche Texte verlangen natürlich eine 'tiefe' Struktur mit vielen Ebenen. In einem Hypertext verwirrt das aber eher, als daß es die Sache verdeutlicht, weil man ohnehin Sprünge vornimmt.

Je inhaltsreicher eine Überschrift ist, desto schneller findet der Leser, was er sucht. *Kapitel 2* oder *§4.2.1* sind keine besonders guten Überschriften. 'Unterhaltsame' Überschriften, die eher danach streben, das Interesse des Lesers zu fangen, statt den Inhalt eines Abschnitts zu verdeutlichen, nutzen leider wenig, wenn man den Text nicht linear liest.

Sei ziemlich konsequent in der Länge Deiner Abschnitte und der Menge der Abschnitte unter jeder Überschrift. Unterrubriken, das heißt kurze Beschreibungen des Inhalts eines Artikels unter seiner Überschrift, sind aus Zeitungen und Illustrierten vertraut und können sehr hilfreich sein.

Wo es den Text übersichtlicher macht, sollest Du Listen und Punktaufstellungen benutzen.

Sage nichts über das Medium

In einem Zeitungsartikel wirst Du kaum lesen: 'Dies ist ein Zeitungsartikel'. Genauso überflüssig ist es, in einer Website zu sagen, daß es sich um eine Website handelt. Das kann unter Umständen sogar ziemlich irritieren.

Schreibe also nicht *Narnias Website*, *Willkommen bei Narnia Online!* oder so ähnlich. Eine Überschrift á la *Narnia* und eventuell ein *Willkommen*! reichen völlig aus – der Benutzer weiß ohnehin, daß er online ist.

Schreibe auch nicht: diese Website enthält – begnüge Dich mit dem Inhalt. Statt 'Links zu relevanten Websites' reicht ein 'siehe auch…'.

Rede nicht von der Technik, die hinter den Seiten steckt – das interessiert nur Webdesigner, während die Benutzer eher am Inhalt interessiert sind. Mußt Du ausdrücklich erklären, wie man technisch durch Deine Site navigiert oder sie zum Laufen bringt, etwa mit einem „Bildschirmeinstellung auf mindestens die und die Breite setzen", stimmt offensichtlich etwas nicht an Deinem Design.

Unter Umständen kommst Du natürlich nicht daran vorbei, von der Technik zu reden – zum Beispiel wenn Du Plug-Ins benutzt und angibst, welches denn nun nötig ist und wo man es findet. Damit versperrst Du aber auch vielen Leuten den Zugang zu Deiner Website. Nur wenige Benutzer im Web verstehen sich darauf, Software downzuloaden oder sie gar zu installieren. Vergiß nicht, daß das Publikum im Web ziemlich unterschiedliche Voraussetzungen hat.

Sprache

Achte darauf, schnell und deutlich klarzumachen, was man eigentlich mit Deiner Website anfangen kann. Im

allgemeinen haben Webbenutzer wenig Geduld mit Websites, deren Zweck nicht unmittelbar einleuchtet.

Hast Du erst Deine Zielgruppe 'eingefangen', darf der Inhalt gerne kompliziert werden. Die meisten Webbenutzer wenden gerne viel Geduld und Energie darauf, ein Thema gründlich und tiefgehend zu verstehen. Oft benutzt man das Web für Informationen, die so gründlich oder spezialisiert nicht in der Stadtbücherei zu finden sind.

Bei Websites muß Deine Korrekturlesung ebenso gründlich und sorgfältig sein wie bei einer gedruckten Ausgabe.

Sei vorsichtig mit englischen Redewendungen. Da Englisch im Internet so sehr dominiert, rutscht man leicht aus mit einer 'anglifizierten' Sprache.

Seitenlayout

Der Sinn des Seitenlayouts ist es, eine Seite durchschaubarer zu machen, indem man sie in Blöcke oder Bereiche aufteilt. Große Informationsmengen werden übersichtlicher, wenn sie durch das Layout in Gruppen und Untergruppen aufgeteilt werden. Als Beispiel solltest Du Dir `www.news.com` ansehen. Ein Text ohne Überschriften oder Einteilungen oder ein Text mit doppeltem Zeilenabstand nach jeder zweiten Zeile wird leicht zu einer unübersichtlichen grauen Masse.

Betrachtest Du eine Seite aus einem gewissen Abstand, stellst Du fest, daß jedes Feld seine eigene Farbe und Schwere hat. Der Text ist grau. Überschriften wirken schwerer als Text, was dem Auge hilft, die Seite in Überschrift und Textkörper aufzuteilen. Senkrechter Abstand zwischen Textblöcken teilt die gesamte Textmenge deutlicher auf, und mit dem richtigen Abstand ergeben sich Textabsätze.

Blöcke, die außerhalb der Texthierarchie stehen, wie etwa Kopfzeile, Navigationsleiste, Links zu anderen Seiten, können durch ihre Farbe oder Schwere vom Textkörper und voneinander getrennt werden.

Der Abstand ist das wichtigste Werkzeug bei der Aufteilung einer Seite. Je mehr Raum um ein Element, desto deutlicher steht es da und desto lesbarer wird der Inhalt.

Will das Auge eine Seite überblicken, bewegt es sich von oben links nach unten rechts schräg über den Bildschirm. Die wichtigsten Informationen solltest Du auf dieser Linie anbringen – zum Beispiel die Überschrift der Seite in der oberen linken Ecke und darunter die Unterrubrik/das Inhaltsverzeichnis. Setzt Du die Überschrift oder das *Logo* der *Site* links und die Überschrift der *Seite* rechts, dauert es etwas, bis sich der Leser orientiert.

Sei konsequent beim Seitenlayout! Alle Seiten sollten möglichst denselben Aufbau und in etwa dieselben Navigationswerkzeuge an denselben Stellen haben. Das hilft dem Leser.

Es lohnt sich, eine Skizze vom Seitenlayout auf Papier aufzuzeichnen, bevor Du es auf dem Bildschirm erstellst. Deine Skizze sollte die einzelnen Felder in verschiedenen Grautönen darstellen, um ein gutes Gleichgewicht zu finden. Mache Deine Skizze so abstrakt wie möglich, mit wenigen Strichen oder Formen. Ergeben sich klare Linien oder Formen in der Komposition, die das Auge sieht?

Funktionelle Blöcke

Die meisten Websites bauen auf funktionellen Blöcken auf: oben steht eine Kopfzeile mit einer Identifikation der Site, also Logo und Namen, und der Überschrift/dem Thema der Seite. Dazu kommen einige feste Schaltknöpfe namens 'Home', 'Suche', 'Feedback', 'Inhalt' und so weiter, eine Navigationsleiste mit den Hauptgruppen in der Site oder im Thema und vielleicht 'vorwärts' und 'zurück'.

Die Navigationsknöpfe sitzen am besten oben rechts, weil man sie meist erst dann benötigt, wenn man einen Überblick über den Inhalt der Seite hat.

Längere Seiten erfordern außerdem eine Fußzeile, die die wichtigsten Navigationshilfen wiederholt – zumindest einen Link zum Seitenkopf. Außerdem finden sich in der Fußzeile meist ein E-Mail-Link zum Webmaster sowie eventuelle Meta-Informationen wie Copyright, Datum, URL, falls der Leser die Seite ausdrucken will. Der 'natürliche' Navigationsknopf – je nach der Struktur der Site vorwärts zur nächsten Seite oder zurück zur Eingangsseite – sollte unten rechts wiederholt werden.

Design am sichtbaren Bereich ausrichten

Nur die wenigsten Webseiten werden je gescrollt. Man sieht sich an, was sichtbar ist, und springt dann weiter. Darum solltest Du oben auf jeder Seite mitteilen, was sie enthält. Deckt die Überschrift nicht den ganzen Inhalt der Seite ab, solltest Du oben ein kleines Inhaltsverzeichnis mit allen Unterpunkten erstellen.

640 x 480 ist die kleinste allgemein benutzte Bildschirmauflösung. Füllt der *Navigator* den gesamten Bildschirm bei aktivierten Werkzeugleisten, der Statusleiste und der Win95-Leiste, enthält das Browserfenster 620 x 290 Pixel. Deaktiviert man die überflüssigen Leisten, steigt die Höhe auf 350 Pixel – das tun aber

nicht alle Leute, vielleicht, weil sie gar nicht wissen, wie.

Kopfzeile, Inhaltsangabe der Seite und die hauptsächlichen Navigationsknöpfe sollten alle innerhalb dieses 'Minimalbildschirms' liegen. Bilder sollten ebenfalls diesen Bereich nicht übersteigen.

Auf einem Macintosh füllt das Browserfenster im allgemeinen nicht die gesamte Breite des Bildschirms – rechts bleibt ein Streifen frei, in dem der Desktop erscheint. In diesem Fall füllt das Browserfenster 480 Pixel in der Breite.

Nichts übertreiben

Je mehr Schaltknöpfe, Symbole, Leisten und Sonderspalten es auf einer Seite gibt, desto unübersichtlicher und weniger lesefreundlich wird sie. Begnüge Dich also mit dem notwendigsten.

Bilder, die keine selbständige Funktion haben, solltest Du vermeiden. 'Bullets', das heißt Sternchen, Kugeln und andere grafische Elemente vor Listenpunkten, machen weniger auf den eigentlichen Inhalt als auf sich selber aufmerksam – ebenso Rahmen und Ränder um Tabellen und dergleichen. Bunte oder blinkende Striche zwischen Textabsätzen überschreien den Text.

Typographie

Unter Typographie verstehen wir – zumindest im folgenden – die Kombination von Zeichensatz, Schriftschnitt, Schriftgröße und Farbe, die dem Leser die Bedeutung eines Textabsatzes verdeutlicht. Damit solltest Du vorsichtig umgehen! Viele verschiedene Typographien wirken verwirrend. Ich würde eine Typographie für den Textkörper und höchstens zwei weitere für Überschriften empfehlen. Je weniger Text es gibt, desto geringer seine Anforderungen an Lesefreundlichkeit. Der Textkörper muß so lesbar wie möglich gestaltet werden, die Typographie für den Titel der Eingangsseite dagegen soll vor allem gut aussehen.

Schriftgröße

Halte Dich an eine Schriftgröße für den Textkörper. Allzu große Überschriften stören beim Lesen. Markierst Du eine Überschrift mit fetter oder kursiver Schrift oder dergleichen, braucht sie nicht größer als der Textkörper zu sein, und außerdem ist das oft lesefreundlicher.

Schriftschnitt

GROSSBUCHSTABEN UND KAPITÄLCHEN sind nicht besonderes lesefreundlich, weil hier alle Buchstaben die gleiche Höhe haben. Du solltest sie nur für Titel und ähnliche einmalig erscheinende Dinge benutzen.

Kursiv ist auf einem Bildschirm ebenfalls wenig lesefreundlich, weil die schrägen Linien gegen die Pixel auf dem Bildschirm stehen. *Kursiv* sollte also nur für einzelne Wörter benutzt werden, nicht aber für komplette Absätze oder Überschriften. Willst Du einen Absatz hervorheben, zum Beispiel als Zitat, machst Du das mit einem Einzug. **Fett** liest sich leichter als *kursiv*, stört aber andererseits das Schriftbild mehr – es eignet sich also vor allem für Text, der nur durchflogen und nicht gründlich gelesen wird, wie Wörter in einer Liste.

Im allgemeinen empfiehlt es sich nicht, **fett** und *kursiv* im selben Textkörper zu benutzen – das ergibt ein ziemlich unsauberes Schriftbild.

Unterstreichung darf niemals zur Hervorhebung eines Textes benutzt werden – der Leser faßt das als Link auf.

Zeichensatz

Auf einer Seite solltest Du nur zwei Zeichensätze benutzen – den einen für Überschriften, den anderen für Textkörper. Benutzt Du mehr, wirkt das verwirrend und chaotisch. Manche Zeichensätze eignen sich besonders für den Bildschirm, so zum Beispiel Times, Arial/Helvetica und Courier. Andere Schriften sind hier weniger lesbar – die meisten Zeichensätze sind nun einmal für den Druck auf Papier bestimmt.

Farben

Schwarzer Text ist für die Augen unmittelbar das angenehmste. Weißer Hintergrund ergibt hier den größten Kontrast zum Text, auf vielen Bildschirmen wird das aber zu leuchtkräftig und so allzu anstrengend für die Augen. Eine diskrete helle Hintergrundfarbe ist angenehmer.

Mit Farben kannst Du eine Typographie hervorheben – zum Beispiel ergibt eine Sonderfarbe für Überschriften mehr Leben und Kontrast im Verhältnis zum Textkörper. Wie bei Zeichensätzen würde ich auch hier höchstens zwei empfehlen: eine für Überschriften und eine für den Textkörper.

Abstand

Unmittelbar setzt ein Browser eine leere Zeile zwischen zwei Absätze. Das ist aber nicht besonders lesefreundlich. Ein Abstand zwischen Textblöcken sollte eigentlich Zusammenhänge und Einteilungen im Text verdeutlichen. Also sollte es möglichst keinen Abstand zwischen Überschrift und Textkörper oder zwischen einzelnen Absätzen im Textkörper geben. Der sollte nur über Überschriften oder für größere Pausen im Text benutzt werden.

Balken und Schnörkel

In Situationen, wo eine Überschrift fehl am Platze wäre, kannst Du eine Vignette oder ein Symbol benutzen. Das könnte zum Beispiel für eine Pause gelten, einen Bruch oder eine Stimmungsänderung in einer Novelle oder einem Essay, der nicht durch Überschriften unterteilt wird.

HTML verfügt für diesen Zweck über ein integriertes Symbol: den waagerechten Balken. Allerdings ist so ein Balken nicht besonders schön – auch dann nicht, wenn er regenbogenbunt ist oder blinkt oder was viele Websites sich nun einfallen lassen. Meist erfüllen ein Sternchen oder eine Grafik mit einem einfachen Schnörkel diesen Zweck viel besser.

Seitenränder und Zeilenlänge

Je mehr Freiraum es um den Text gibt, desto einfacher ist er zu lesen. Die optimale Zeilenlänge liegt laut Experten bei 40-60 Zeichen. Was das in Pixel bedeutet, ist nicht einfach zu sagen – es hängt vom Zeichensatz und der Textgröße ab. 12 Punkt Times New Roman, *Netscape Navigators* normale Einstellung, ergibt bei einer Spaltenbreite von 400 Pixel 50-60 Zeichen. Auf einem Macintosh, dessen Bildschirmschriften etwas kleiner sind, ergibt das etwas mehr Zeichen.

Sei vorsichtig, wenn es um die Justierung von Überschriften oder Bildern geht – eine Zentrierung bricht mit der lesefreundlicheren Linksjustierung.

Farben für Links

Normalerweise werden Links blau dargestellt und bereits besuchte Links violett. Experten im Interface-Design machten frühzeitig darauf aufmerksam, daß das nicht gerade klug ist: Links müssen natürlich auffallen und sollten daher die auffälligste Farbe haben, zum Beispiel rot. Bereits besuchte Links dagegen sollten in den Hintergrund treten und also in einer etwas dezenteren Farbe erscheinen, etwa blau.

Das hielten etliche Webdesigner für vernünftig, weswegen sie sich daran hielten. Und allmählich wurde die Bedeutung der Farben ziemlich unverständlich, weil jeder seine eigenen Regeln benutzte.

Darum würde ich Dir, was Farben angeht, folgendes raten: Halte Dich an den vorgegebenen Standard. Willst Du ihn unbedingt ändern, mußt Du eindeutig festlegen, was jede Farbe bedeutet. Gib neuen Links eine klare, reine Farbe, etwa rot, blau oder grün, und bereits besuchten Links eine weniger klare, dunklere oder grautönigere Version derselben Farbe. Enthält Dein Textkörper zahlreiche Links, sollte die Link-Farbe allerdings nicht zu kräftig ausfallen, weil das das Lesen erschwert.

Eingangsseite

Das Design der Startseite für Deine Website ist natürlich ziemlich wichtig. Folgende Informationen solltest Du im oberen Teil dieser Seite anbringen, also in dem Teil, der ohne Scrollen sichtbar ist:

- WAS: welchen Zweck hat diese Website, was enthält sie und wozu dient sie?
- WER: wer zeichnet für die Site verantwortlich? Die meisten Websurfer entwickeln rasch einen gesunden kritischen Sinn gegenüber ihren Quellen und haben wenig Vertrauen zu Sites, deren Urheber nicht erkennbar sind.

Außerdem sollte die Eingangsseite möglichst einen E-Mail-Link zum Webmaster enthalten, so kann ein Besucher nämlich auf eventuelle Fehler aufmerksam machen.

Hat die Organisation hinter der Site eine Adresse und eine Telefonnummer in der 'Nicht-Cyber-Welt', sollten diese auf der Eingangsseite stehen – entweder als solche oder aber mit einem eindeutigen Link.

Literatur

Ein paar empfehlenswerte Websites zum Thema Webdesign vom Designgesichtspunkt her:

Web Pages That Suck „Learn Good Web Page Design by Looking at Bad Web Pages"
`www.webpagesthatsuck.com`

Yale Web Style Guide Gründliche, seriöse Behandlung aller Aspekte im Webdesign
`info.med.yale.edu/caim/manual/`

Web Page Design for Designers – Hauptgewicht auf Layout und Typographie
`ds.dial.pipex.com/pixelp/ wpdesign/`

Guide to Web Style Suns Führer für Webdesign, Hauptgewicht auf Funktionalität und Navigation
`www.sun.com/styleguide/`

Testen

Am besten setzt Du eine Person, die selber noch keine Website gemacht hat, an Deinen Computer und bittest sie, die Site zu testen. Sie soll Dir alles mitteilen, was ihr auffällt – wie sieht die Sache als Ganzes aus, wie funktioniert die Navigation und so weiter. Antworte nicht auf ihre Fragen, sondern notiere ihre Reaktionen. Hinterher kannst Du sie zwar fragen, ob die Site wie geplant funktioniert – aber meistens sagen die unmittelbaren Reaktionen mehr über die Schwächen einer Site: 'Hoppla, wo bin ich denn jetzt gelandet? … wie komme ich zurück? …wo finde ich wohl…?'

Wartung einer Website

Es muß jemanden geben, der für die Website verantwortlich ist – einen Webmaster. Seine Aufgabe ist, die notwendigen Berichtigungen und Aktualisierungen in der Site zu erledigen. Der Webmaster ist nicht unbedingt mit dem Webdesigner identisch. Viele Unternehmen lassen ihre Website von einem externen Büro entwerfen und sie dann von einem internen Webmaster warten. Der Webdesigner ist sozusagen der Architekt und der Webmaster der Hausmeister.

Der Webmaster sollte eine Kopie der Website auf seinem eigenen PC installieren. Hier kann er Berichtigungen und Änderungen vornehmen und sie testen, bevor die Seiten auf den Server überführt werden – auch dann, wenn der Server ans Lokalnetz angeschlossen ist.

Es gibt einige Grundvorgänge, die unbedingt in die Organisation eingearbeitet werden müssen:

- Die Website muß in den 'Informationsfluß' des Unternehmens integriert werden – welche Informationen sollen laufend im Web veröffentlicht werden: Pressemitteilungen, Kalender, wichtige Begebenheiten, Produkte und so weiter.
- Jeder, der Beiträge liefert, muß wissen, welche Formate der Webmaster vorzieht – RTF oder HTML, TIFF oder GIF und so weiter.
- Alle fertiggestellten Webseiten, egal von wem, werden an den Webmaster gegeben – nur er hat das Recht zum Uploaden.
- Das Webmaterial muß einer ebenso sorgfältigen Korrektur unterzogen werden wie gedrucktes Material. Merkwürdigerweise sind Schreibfehler in einem Browser sehr viel deutlicher als in einer Textverarbeitung.
- E-Mail an die Website muß gelesen und gegebenenfalls beantwortet werden. Vielleicht entdeckt jemand einen Fehler, den Du selber nicht bemerkt hast, und eine 'dumme' Frage zeigt Dir vielleicht, daß Deine Website doch nicht so logisch und einleuchtend aufgebaut ist, wie Du Dir das vorstellst.
- Du mußt unbedingt darauf achten, daß Deine Website keine veralteten Informationen enthält. Halte sie auf dem aktuellen Stand und lösche veraltete Seiten.

Wie viele Leute haben Deine Site besucht?

Willst Du wissen, wie viele Leute Deine Site besuchen, analysiere das Logbuch des Servers. Besitzt Du keinen eigenen Server, kannst Du das natürlich nicht ohne weiteres, also mußt Du mit Deinem Provider reden. Auf die sogenannten 'Counters', also Zähler, die es auf vielen Websites gibt, ist nicht unbedingt Verlaß.

Webdesign für ein Intranet

Ein Intranet ist ein internes Netzwerk in einem Unternehmen, das dieselben Komponenten benutzt wie das Internet: Browser, HTTP, HTML, E-Mail. Früher benutzten die Unternehmen viele verschiedene, zueinander mehr oder weniger inkompatible Netzwerktypen, allmählich erkennt man aber, daß es einfacher und billiger ist, sich an die Internetstandards zu halten.

Webdesign für ein Intranet ist nicht wesentlich anders als Webdesign fürs Internet. Die drei Hauptunterschiede sind:

1. Normalerweise hat man ein äußeres Web, das für alle zugänglich ist, und ein inneres Web, das nur den Betriebsangehörigen offensteht. Die beiden Webs können mehr oder weniger integriert sein.

2. Man hat wirklichen Überblick und Kontrolle über die Software der Clients: Betriebssysteme, Browserversionen und so weiter. Das erleichtert es, die Funktionalität auf Plug-Ins, JavaScript, ActiveX und dergleichen aufzubauen, weil man sichern kann, daß alle Benutzer dasselbe Programm haben.

3. Die Bandbreite in einem Intranet kann das Vielfache der Bandbreite im Internet betragen. Das ermöglicht eine ausgedehntere Anwendung von Multimedia und anderen spannenden, aber platzintensiven Effekten.

So veröffentlichst Du Deine Website

Wie findet man Deine Adresse?

Zunächst einmal kannst Du Leuten davon erzählen oder die Adresse auf Deine Karte setzen und dergleichen mehr. Schreibe den URL so kurz wie möglich – kannst Du Dich mit
`www.ozymandias.de`
begnügen, brauchst Du nicht
`http://www.ozymandias.de/ index.html`
zu schreiben.

Wie kommt Deine Website in Yahoo, Dino, AltaVista etc.?

Du meldest Deine Website an. Viele Leute in Deutschland benutzen DINO, um deutsche Websites zu finden. Du solltest Dich also vor allem hier anmelden. Du aktivierst die Eingangsseite von DINO über `www.dino-online.de`, klickst auf den `DINO-Katalog` und dann auf 'Seite anmelden'. Anschließend kannst Du die Kategorie Deiner Seite bestimmen und die Seite anmelden.

Was Dein internationales Publikum betrifft, gibt es Websites, die Deine Seiten auf einen Schlag bei vielen internationalen Indizes und Katalogen wie Yahoo und AltaVista anmelden können. Versuche es zum Beispiel mit
`www.liquidimaging.com/liqimg/ submit/`

Vergiß nicht, **meta**-Tags zu erstellen, damit Deine Seiten nach Wunsch indexiert werden.

Achte auch auf sinnvolle Titel, die selbständig verständlich sind.

Die sicherste Weise, entdeckt zu werden, ist aber, andere Sites dazu zu bringen, daß sie einen Link auf Deine Site einrichten. Du kannst E-Mails an artverwandte Sites schicken und Deine Ankunft im Web melden – und eventuell schreiben, daß Du zu ihnen linkst, wenn sie zu Dir linken. Ist Deine Website spannend, tun sie das sicher. Das gilt allerdings nicht für kommerzielle Sites. Es gibt auch Newsgroups, in denen Du neue, relevante Sites mitteilen kannst. Allerdings solltest Du zuerst den FAQ der Newsgroup lesen – nicht überall wird die Annoncierung von Websites akzeptiert, vor allem wenn sie kommerziell sind. Über Newsgroups erfährst Du mehr in *Wie startet man ins Internet* im Abschnitt *USENET*.

Urheberrecht im Internet?

Gerade hier gibt es viele Mißverständnisse. Doch es gelten dieselben Regeln wie sonst auch:

- Der Urheber hat automatisch das Urheberrecht (oder *Copyright*) für alles originale 'geistige Eigentum', also Originaltext, Grafik, Musik, Programmcode und so weiter. Es ist *nicht* notwendig, ausdrücklich „©" oder „Copyright by…" im Material anzuführen oder das Material irgendwo zu registrieren, um das Copyright zu beanspruchen.
- Es ist verboten, Material anzuwenden oder zu kopieren, dessen Urheberrecht von jemandem beansprucht wird – es sei denn man hat eine ausdrückliche Genehmigung. Man darf allerdings „in angemessenem Maß" zitieren oder dergleichen.
- Das Copyright entfällt automatisch nach einer gewissen Zeit. Bei Literatur sind das 70 Jahre nach dem Tod des Urhebers. Danach kann das Material frei benutzt werden.

Das Web erhebt auch neue Fragen zum Urheberrecht, die noch nicht ganz geklärt sind. Die Regeln scheinen etwa so auszusehen:

- Es ist keine Genehmigung nötig, um einen Link zum Material anderer zu erstellen.
- Es ist nicht erlaubt, Material auf anderen Servern so „einzukapseln", daß es wie eigenes Material aussieht – etwa durch Einsetzen eines Bildes, dessen Quelldatei auf einem anderen Server liegt, oder durch das „Framen" eines Dokuments auf einem anderen Server, um es als Teil des eigenen Materials erscheinen zu lassen.

Mehr über Urheberrechts-Regeln erfährst Du unter
10 Big Myths about copyright explained:
`www.clarinet.com/brad/ copymyths.html`

Die Zukunft im Web

HTML entwickelt sich ständig weiter. Zur Zeit bereitet W3C die endgültigen Spezifikationen für HTML 4.0 vor, die einerseits Frames, Scripting und andere allgemein akzeptierte Punkte enthalten, andererseits HTML den Umgang mit mehreren Schriftsprachen wie z.B. Chinesisch oder Japanisch, mit wechselnden Schriftrichtungen und anderem mehr ermöglichen. Die vorläufigen Spezifikationen findest Du unter
`http://www.w3.org/TR/WD-html40/`.

Es gibt noch weitere spannende Entwicklungen wie *Push, PNG, XML, Desktop integration, Dynamic HTML, Document Object Model, CORBA/IIOP*. Mehr zu diesen Punkten findest Du im Online-Supplement.

Davon solltest Du die Finger lassen!

Natürlich macht Dich das Lesen eines solchen Heftes nicht zum Profi – hier folgen aber einige beachtenswerte Punkte, wenn Du nicht den Eindruck eines Amateurs erwecken willst:

- Keine Hintergrundbilder.
- Halte Dich an die vorausdefinierten Farben für Links und besuchte Links. Mußt Du sie ändern, mache deutlich, wofür sie stehen.
- Verwende keine unnötigen und irritierenden Spezialeffekte wie animierte GIFs und JavaScript-Lauftexte.
- Halte Dich an wenige verschiedene Typographien auf ein und derselben Seite – zwei, höchstens drei Kombinationen von Zeichensatz, Schriftgröße und -farbe.
- Sei konsequent bei Deiner Schriftwahl.Mache keine inhaltslosen Seiten, etwa Seiten, die nur ein Logo oder einen Titel oder zwei bis drei Links zu Unter-Seiten enthalten.
- Benutze keine Tabellen mit dreidimensionalem Rahmen. Vermeide dreidimensionale waagerechte Striche. Das macht einen schlechten Eindruck.
- Beziehe Dich nicht auf das Medium. Schreibe nicht ‘diese Website …’. Keine Links mit dem Text ”…hier klicken…” oder ”…dieser Link führt zu…”. Schreibe lieber den Inhalt des Links. Verweise nicht auf Links: ’unten findest Du einen Link zu …’.
- Stelle keine Seiten her, die nur von einem einzelnen Browser richtig dargestellt werden. Es ist gar nicht schwer, Seiten zu erstellen, mit denen alle Browser umgehen können, außerdem wird sich sicher niemand einen Browser downloaden, nur um Deine Seite zu sehen. Schreibe nicht ’Diese Seite erscheint am besten in dem und dem Browser’.
- Vermeide Reklame für einen Browser auf Deiner Seite, das macht einen schlechten Eindruck. Andere tun das oft, weil sie Schmiergeld erhalten. Microsoft bietet Webdesignern, die das MSIE-Logo auf ihre Webseiten setzen, kostenlose Software, Pizzas und anderes mehr an.
- Schreibe nicht „Im Bau” bzw. „Hier gibt es bald mehr Inhalt”. Webseiten sind grundsätzlich im Bau. Nimm nur fertige Seiten.
- Benutze keine Plug-Ins ohne guten Grund.
- Setze Deinen Textkörper nicht größer als normal – und gib ihm keine Textfarbe.
- Vermeide Ausdrücke wie ‘Cyber-’, ‘Cyberspace’ oder ‘Infoautobahn’.
- Sorge für die Aktualität. Aktualisiere Deine Seiten oder lösche sie, wenn sie veraltet sind.
- Achte auf gute Sprache. Sei vorsichtig mit der ‘Anglifizierung’ der Sprache. Ist Dein Publikum deutsch, schreibe auf Deutsch.

ActiveX Microsofts Antwort auf Java, Plug-Ins usw., *Windows*-zentriert.
Alta Vista eine der stärksten Suchmaschinen; www.altavista.digital.com
Animation eine Sequenz zusammenhängender Bildern, etwa ein Zeichentrickfilm.
Anker Punkt in einem HTML-Dokument, der eine Link-Verknüpfung dorthin ermöglicht.
Anti-alias ein grafischer Trick, der durch Farbabstufung rauhe Kanten ausgleicht.
Applet in Webseite integriertes Java-Programm.
ASCII Zeichensatz, der die üblichsten Zeichen enthält, unter anderem das englische Alphabet.
Attribut Erweiterung zu einem Tag.
Bandbreite Kapazität der Datenüberführung einer Verbindung. Je größer desto schneller.
Browser Client-Programm, das Webseiten darstellt und die Navigierung durch sie ermöglicht.
Browserkrieg malerischer Ausdruck für den Wettstreit zwischen den führenden Browserproduzenten Netscape und Microsoft.
Cache beim Browser die kürzlich eingelesenen Dateien, die noch im Speicher liegen.
CGI *Common Gateway Interface* System, das über eine Webseite ein Server-Programm aktiviert.
Client Programm auf der Benutzerseite im Netzwerk, z.B. Webbrowser oder Mailprogramm.
cookie Informations-'Fetzen', den ein Server im Browser speichert, so daß er diesen (damit Dich) beim nächsten Besuch wiedererkennt.
CSS *Cascading Style Sheets* typographische Sprache, die in Style Sheets benutzt wird.
Cyber- siehe Cyberspace. Stammt eigentlich von dem Begriff *Cybernetics*.
Cyberspace Begriff stammt aus dem Science-Fiction-Roman *Neuromancer* (1984) von *William Gibson*, bez. ein Virtual-Reality-Computernetzwerk. C. und Cyber- werden kritiklos für alles benutzt, was Computer, Internet, Informationstechnologie und andere coole Dinge angeht.
dither Simulation einer Farbe in einem Bild durch Mischung von Punkten anderer Farben.
Domänen-Name Teil eines Servernamens
Download eine Datei aus dem Internet holen, um sie auf dem eigenen Computer zu speichern.
Editor Programm, in dem man eine Datei bearbeitet, wie etwa ein Webeditor für Webseiten oder ein Texteditor für Textdateien.
FAQ *Frequently Asked Questions* – Liste üblicher Fragen und Antworten zu einem Thema.
fließende Bilder Bilder auf Webseite, die rechts- oder linksjustiert und vom Text umflossen werden.
Formular Ansammlung von Feldern, in die Informationen eingefügt werden, die an ein CGI-Programm weitergegeben werden.
Frame Bestandteil eines Framesets.
Frameset Dokument, das das Browserfenster in mehrere getrennte Dokumente unterteilt.
Freeware kostenlose Software.
FTP *File Transfer Protocol* – Protokoll, durch das im Internet Dateien überführt und die Dateistrukturen auf einem Server manipuliert werden – Ordner erstellen, Dateien löschen, ändern usw.
FTP-Programm Programm zur Übertragung und Manipulation von Dateien über das Internet.
GIF im Web vielbenutztes Grafikformat, das sich vor allem für einfache Grafiken eignet.
GIF-Animation Variation des GIF-Formates, die Animationen enthält.
Header Meta-Information zu einem Dokument, z.B. MIME-Type, der bei einer HTTP-Überführung vor dem eigentlichen Dokument übermittelt wird.
Homepage *1.* die Eingangsseite, Startseite einer Website. *2.* die Seite, mit der der Browser startet. *3. umgangssprachlich* für eine Website.
HTML *HyperText Markup Language*. Kodierungssprache für Webseiten. Ein HTML-Dokument enthält Text, Bilder, Links zu anderen Web-Objekten.
HTTP *HyperText Transfer Protocol*. Wird benutzt, um Webseiten von einem Server zu übermitteln.
Hyperlink aktiver Verweis in Hypertext o. -medium.
Hypermedia siehe Hypertext, kann desweiteren Bilder, Ton und dergleichen enthalten.
Hypertext Text, der vielfache Verweise enthält, die man unmittelbar verfolgen kann, was zu einem nonlinearen Lesen des Textes führt.
Imagemap Bild, dessen einzelne Bereiche Links zu verschiedenen Stellen sind.
Infoautobahn (nach dem englischen *the information superhighway*) (*umg.*) die informationstechnische Infrastruktur, hierunter das Internet.
Interaktivität wenn ein Programm auf einen Input seitens des Benutzers reagiert. Die einfachste Form von Interaktivität im Web sind Links.
Internetanbieter Firma, die Dienste wie Internetanschluß, Server, Webhotel etc. anbietet.
Intranet internes Netzwerk in einem Unternehmen, das mit dem Internet kompatibel ist.
ISDN spezielle digitale Telefonverbindung mit wesentlich größerer Bandbreite als eine normale Modemverbindung.
ISO-8859-1 Zeichensatz, Erweiterung von ASCII mit allen Sonderzeichen der westeuropäischen Sprachen wie *ä*, *ö*, *ü*.
Java Programmiersprache, speziell für den Netzgebrauch entwickelt, plattformunabhängig, entwickelt von *Sun Microsystems*
JavaScript rudimentäre Programmiersprache, die in Verbindung mit HTML Webseiten interaktiver machen kann, entwickelt von Netscape.
JPEG Web-Grafikformat, gut für Photos.
JScript Microsofts Ausgabe von JavaScript.
Komprimierung Methode, dank derer eine Datei weniger Raum einnimmt als ursprünglich.
Link Hyperlink.
Microsoft weltweit führender Softwareproduzent, auch im Internet *(MSIE, FrontPage)*

MIME *Multipurpose Internet Mail Extension* Standard, nach dem Server im Internet das Format für versendete Dateien angeben.
MSIE *Microsoft Internet Explorer* populärer Browser, Nr. 2 nach *Netscape Navigator*.
Multimedia meist Kombinationen von interaktiven, nicht-textbasierten Medien wie Ton oder Film auf einem Computer, oft auf CD-ROM.
Netscape Softwarefirma; lange führend in Internetsoftware (*Netscape Navigator*) und Servern, sehr wichtig für die Entwicklung des Web.
Newsgroups Diskussionsgruppen im Internet.
Objekt *1.* in HTML: externe Datei, die als Feld in eine Webseite integriert wird, (Grafik, Applet, Plug-In.) *2.* beim Programmieren: Ansammlung von Funktionen und Daten; ein Element mit Eigenschaften und Funktionen; ein Programmmodul, das mit anderen Modulen kommuniziert.
objektorientiert beim Programmieren: auf Objekten aufgebaut.
offen bei Datenformaten, Standards, Protokollen verwendet, Bezeichnung für 'im allgemeinen 'nicht von einem kommerziellen Unternehmen kontrolliert, Spezifikationen zugänglich für alle, kann von jedermann benutzt werden'.
Palette die Farben, die zu einem Bild gehören.
PDF *Portable Document Format* Format für die Überführung von fertiggestalteten Dokumenten, die ausgedruckt werden sollen.
Pixel Bildpunkt auf einem Bildschirm.
Plattform Kombination von Hardware und Betriebssystem.
plattformunabhängig Software oder Daten, die von Computern verschiedener Hardware- und Betriebssysteme verarbeitet werden können.
Plug-In Erweiterungsmodul für einen Browser, das diesen zum Umgang mit einem neuen Datenformat befähigt, etwa Film, Ton, VRML usw.
Protokoll Regelsatz für die Kommunikation über ein Netzwerk, z.B. HTTP für die Überführung von Webseiten und SMTP für E-Mail.
relativer URL Adresse, die im Verhältnis zur Adresse des Dokuments interpretiert wird, in dem er auftaucht.
RFC *Request For Comments* Dokumente, die die im Internet benutzten Standards beschreiben. Der Name besagt, daß solche Standards grundsätzlich veränderlich sind und diskutiert werden.
RTF *Rich Text Format* plattformunabhängiges Dateiformat für den Austausch von Textdokumenten zwischen verschiedenen Programmen.
Server Programm bzw. Computer, der über ein Netzwerk Dienstleistungen an Clients bietet. Ein Webserver liefert z.B. Webseiten und andere Dokumente an einen Client mit einem Webbrowser.
SGML *Standardized General Markup Language* 'Metasprache' zur Entwicklung von Datenformaten. HTML entstand auf Basis von SGML.
Shareware Programme, die man vor dem Kauf (oft 30 Tage) kostenlos ausprobieren kann.
Site kurz für Website
Style Sheets Vorschrift, die HTML mit erweiterten Layout- und Typographiebeschreibungen versieht.
Tabelle Rahmenstruktur auf einer Webseite, die für Tabellenaufstellungen und Layoutzwecke dient.
Tag Markierung o. Formatierungscode in HTML.
Texteditor Editor für die Bearbeitung von reinem ASCII- oder ISO-8859-1-Text, etwa *Wordpad* unter *Windows* oder *Simple Text* auf dem *Mac*.
Upload eine Datei vom eigenen Computer auf einen anderen Computer im Internet überführen.
URL *Uniform Resource Locator* Adresse für ein Objekt im Internet, z.B. eine Webseite.
VRML *Virtual Reality Markup Language* Standardformat zur Beschreibung von 3D-Modellen oder -Welten. Kontrolliert von W3C.
W3C Kürzel für World Wide Web Consortium
Web (2) umgangssprachl. Bez. des World Wide Web
Web integrierte Struktur von Hypertext- oder Hypermediadokumenten.
Webdesign die hohe Kunst, Webseiten und Websites zu erstellen.
Webeditor Programm zum Erstellen von Webseiten
Webhotel Mietraum auf dem Server eines Internetanbieters, der die eigene Website aufnimmt.
Webseite Dokument in HTML, das über HTTP aus dem Internet überführt wird.
Website Ort bzw. Stelle im Web; zusammenhängende Gruppe von Webseiten im Internet.
World Wide Web Consortium unabhängige Organisation, die Richtlinien und Empfehlungen für Standards im Web erstellt, z.B. HTML und HTTP.
World Wide Web gigantische Ansammlung von Hypermediadokumenten im Internet, basiert auf der Dokumentbeschreibungssprache HTML und dem Übertragungs-Protokoll HTTP. Das Besondere sind die Links, mit deren Hilfe man weltweit von einem Dokument zum anderen „springt".
WWW Kürzel für World Wide Web
WYSIWYG *What You See Is What You Get* beschreibt unter anderem Webeditoren, die ein einigermaßen realistisches Bild vom endgültigen Aussehen einer Webseite vermitteln – im Gegensatz zu älteren Editoren, die nur den Formatierungscode zeigten.
Yahoo unentbehrlicher umfassender Katalog über Websites – www.yahoo.com

www.KnowWare.de