

HTML

PROGRAMMIERUNG



Dr. Leander Brandl
BRG Keplerstraße - A-8020 Graz
Universität Graz - Institut für Experimentalphysik
Fachdidaktik der Physik

<http://w4.brgkepler.asn-graz.ac.at>



HTML - PROGRAMMIERUNG

INHALTSVERZEICHNIS - SKRIPTUM

0. Allgemeine Informationen
 1. Was benötigt man zur HTML-Programmierung?
 2. Die erste HTML-Seite; Abspeichern
 3. Die Grundstruktur einer HTML-Seite
 4. Der HEAD
 5. Farben auf einer Web-Seite
 6. Grafiken auf einer Web-Seite
 7. Der BODY
 8. Textausgabe - Zeilenumbruch, zentrieren
 9. Textausgabe - Überschriften
 10. Textausgabe - Sonderzeichen
 11. Textausgabe - Textattribute
 12. Textausgabe - Größe, Farben, Absätze
 13. Textausgabe - Aufzählungen
 14. Textausgabe - Einrücken
 15. Anzeigen von Grafiken - Ränder
 16. Anzeigen von Grafiken - Ausrichtung
 17. Trennlinien
 18. Links - absolut
 19. Links - relativ
 20. Links - neues Fenster - Grafiken
 21. Links - innerhalb einer Seite
 22. Tabellen - Grundstruktur
 23. Tabellen - Ränder, Farben, Größe
 24. Tabellen - Zelleneigenschaften
 25. Tabellen - Verbinden von Zellen
 26. Ausrichtung von beliebigen Objekten
 27. Frames - Grundstruktur
 28. Frames - Die Frameset-Definition
 29. Frames - Ränder von Frames, Scrolling
 30. Frames - Links in Framesets
 31. Frames - Verschachteln von Framesets

Anhang A - Dezimal-Hexadezimal-Tabelle

Anhang B - Übersicht HTML-Befehle



1. Die erste Web-Seite
2. Eine Web-Seite mit Seitentitel
3. Web-Seite mit META-Tags
4. Web-Seite mit Farbeinstellungen
5. Automatischer Zeilenumbruch bei der Ausgabe von Text
6. Beabsichtigte Ausgabe von Text in mehreren Zeilen
7. Ausgabe von zentriertem Text
8. Überschriften auf einer Web-Seite
9. Ausgabe von Sonderzeichen
10. Ausgabe von Text mit Textattributen
11. Ausgabe von Text unterschiedlicher Größe und Farbe
12. Ausgabe von ausgerichtetem Text in Absätzen
13. Ausgabe von nummerierten und nicht nummerierten Aufzählungen
14. Die Verwendung von <blockquote>
15. Die erste Ausgabe einer Grafik
16. Rahmen um eine Grafik
17. Horizontale Ausrichtung einer Grafik
18. Vertikale Ausrichtung einer Grafik am Text
19. Verschiedene Trennlinien
20. Absolute Links
21. Relative Links
22. Anzeigen von Links in neuen Browserfenstern
23. Verwendung einer Grafik als Link
24. Links innerhalb einer Seite
25. Die erste Tabelle
26. Tabellen mit unterschiedlichen Rändern
27. Tabelle mit Farbeinstellungen
28. Ausgerichtete Tabellen unterschiedlicher Breite
29. Eine bunte Tabelle mit unterschiedlicher Ausrichtung in den Zellen
30. Verbinden von Zellen
31. Formatierung mit dem <div>-Tag
32. Erstes Frameset
33. Frameset ohne Rahmen
34. Frameset mit Rahmen und fester Breite
35. Die Funktion von Links in Framesets
36. Verschachteltes Frameset 1
37. Verschachteltes Frameset 2
38. Verschachteltes Frameset 3



HTML - PROGRAMMIERUNG

0. ALLGEMEINE INFORMATIONEN

Was ist HTML?

HTML ist eine sogenannte **Dokumentbeschreibungssprache**. Als damit begonnen wurde, das Internet dazu zu verwenden, um grafisch aufbereitete Informationen und Seiten zu übermitteln, wurde diese Sprache entwickelt, mit der die logischen Strukturen und Inhalte einer Web-Seite beschrieben werden.

Die Abkürzung HTML steht für Hyper Text Markup Language.

Wie werden Web-Seiten übertragen bzw. angezeigt?

Hinter der abgebildeten Web-Seite steht der ausschnittsweise angegebene HTML-Code. Wenn eine Web-Seite über das Internet übertragen wird, dann wird nur der HTML-Code (und evt. Bilder, Grafiken, ...) über das Internet übertragen. Um die Web-Seiten dann zu betrachten, benötigt man ein Programm, das diesen HTML-Code interpretiert und aufgrund der enthaltenen Informationen die Web-Seite „erzeugt“ bzw. darstellt. Solche Programme heißen „Web-Browser“. Die zwei zur Zeit verbreitetsten Browser sind der „**Netscape Navigator**“ und der „**Microsoft Internet Explorer**“.

Der HTML-Code für die neben abgebildete Web-Seite ...

```
<html>
<head>
  <meta name="Author" content="Dr. Leander Brandl">
  <title>startseite</title>
</head>
<body>
<center><font size=+3>HTML - Programmierung</font>
<br>
<hr SIZE=1 NOSHADE WIDTH="100%">
<p><br>
<p><img SRC="einsteins.gif" height=52 width=88></center>
<p><br>
<p>
<hr SIZE=1 NOSHADE WIDTH="100%">
<div align=right><i>
<font size=-1>L.Brandl - J&auml;hner 2001
</font></i></div>
</body>
</html>
```



Was hat es mit dem *http://* auf sich?

Übertragen werden die HTML-Dateien mit dem **HTTP-Übertragungsprotokoll** (Hyper Text Transport Protocol). Deshalb musste man in früheren Zeiten bei Internetbrowsern die Zeichenkette **http://** vor eine DNS-Adresse stellen. Der Browser wusste dann, dass er zum Übertragen das HTTP-Protokoll verwenden musste. Mittlerweile genügt es, wenn man bei einem Browser die DNS-Adresse ohne vorangestelltes **http://** eingibt.

Üblicherweise verwendet man diese Schreibweise aber noch immer, wenn man eine Web-Adresse angibt. Verbindlicherweise sollte man das **http://** verwenden, wenn man Web-Adressen per e-mail versendet - der Leser braucht dann lediglich auf die Adresse zu klicken, ein Web-Browser wird automatisch gestartet und die entsprechende Web-Seite angezeigt.

Beispiel für die exakte Schreibweise einer URL (**Uniform Resource Location**):

http://w4.brgkepler.asn-graz.ac.at/brandl/start.html

„es handelt sich um eine HTML-Datei“

die DNS-Adresse des Internet-Servers

ein Unterverzeichnis auf dem Internet-Server

der Dateiname der HTML-Datei



HTML - PROGRAMMIERUNG

1. WAS BENÖTIGT MAN ZUR HTML-PROGRAMM.

Eigentlich benötigt man zur Erstellung von Web-Seiten nur einen Texteditor zum Schreiben und Bearbeiten des HTML-Code. Weiters sollte man natürlich auch einen Browser zur Verfügung haben, um zu kontrollieren, wie dieser den geschriebenen HTML-Code auswertet.

Mittlerweile gibt es HTML-Editoren mit den unterschiedlichsten zusätzlichen Funktionen wie „Sand am Meer“. In dieser Einführung wird zum Schreiben und Bearbeiten des HTML-Code der **Texteditor von Windows** verwendet. Dieser ist Bestandteil des Betriebssystems „Microsoft Windows“ und steht somit auf jedem PC zur Verfügung. Weiters kann aufgrund der Einfachheit dieses Programms auf die Erklärung programmspezifischer Details der mittlerweile schon sehr komplex gewordenen HTML-Editoren verzichtet werden.

Start des Texteditors von Windows:

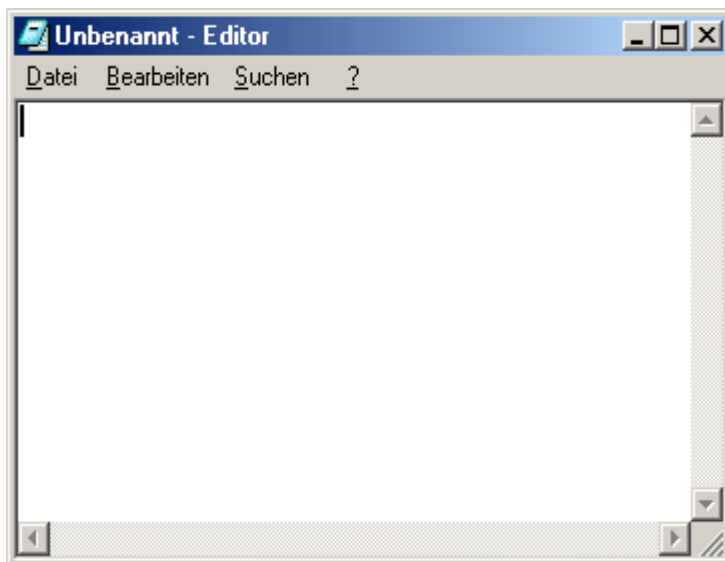
Start --> Programme --> Zubehör --> Editor



Der Texteditor stellt alle bekannten Funktionen zum Eingeben und Bearbeiten von Text zur Verfügung. Im Menü **Datei** findet man die bekannten Einträge **Öffnen**, **Speichern** und **Speichern unter ...** zum Öffnen und Speichern von Text-Dateien.

Und damit wäre auch schon die Natur von HTML-Dateien klar. Es handelt sich dabei um reine (unformatierte) Text-Dateien. Es kommt also nur auf das „WAS“ und nicht das „WIE“ einer HTML-Datei an.

Obwohl an den HTML-Code keine Formatierungs-Forderungen gestellt werden, sollte man sich aus Gründen der Übersichtlichkeit doch an eine gewisse Strukturierung halten - „man wird es sich später selbst danken.“



Die HTML - Sprache:

Weiters benötigt man auch das Wissen über die Sprache HTML. Die reine Auflistung aller HTML-Befehle mit allen deren Eigenschaften (die sogenannte HTML-Referenz) füllt zur Zeit (HTML 4.0, Jänner 2001) ein Buch mit über 700 Seiten. Trotzdem ist es möglich, die Grundzüge dieser Sprache aufgrund ihrer einfachen Struktur in recht kurzer Zeit zu lernen.

Sollte man weiterführende Informationen brauchen oder Genaueres über die Eigenschaften eines HTML-Befehls wissen wollen, wendet man sich am besten an die Entwickler von HTML, dem World Wide Web Consortium. Auf den Web-Seiten dieser Organisation findet man eine komplette HTML-Referenz mit einer recht guten und verständlichen Dokumentation der jeweiligen Befehle und Eigenschaften.

Zu finden ist das W3C unter: <http://www.w3c.org>





HTML - PROGRAMMIERUNG

2. DIE ERSTE HTML-SEITE; ABSPEICHERN

Jede HTML-Datei muss von einem Browser als solche erkannt werden. Dies geschieht durch die beiden HTML-Befehle `<html>` und `</html>`. Alle weiteren Anweisungen, die zu dieser Web-Seite gehören, stehen zwischen diesen beiden Befehlen.

Im Allgemeinen werden HTML-Befehle als sogenannte „**Tags**“ bezeichnet und stehen jeweils in eckigen Klammern `< >`. Somit kann ein Browser genau zwischen einem HTML-Befehl und dem anzuzeigenden Text unterscheiden. Bei den meisten HTML-Tags gibt es zum einleitenden Tag auch einen dazugehörigen beendenden Tag. Diese beiden bilden dann einen sogenannten „**Container**“.

Der grundsätzliche Aufbau einer jeden HTML-Datei:

```
<html>
...
...
...
</html>
```

Wie bereits erwähnt, stehen alle Anweisungen und der auszugebende Text einer HTML-Seite zwischen den beiden Tags `<html>` und `</html>`. Vor dem einleitenden Tag `<html>` und nach dem beendenden Tag `</html>` steht nichts.

Text, der auf einer Web-Seite ausgegeben werden soll, steht unformatiert und ohne einleitenden oder beendenden Tag im Code. Die Formatierungen werden durch HTML-Tags festgelegt.

Abspeichern von Web-Seiten:

Da Web-Seiten auf den verschiedensten Web-Servern zu liegen kommen können, über verschiedenste Computersysteme (Windows, Apple, Unix, Linux, ...) weitergeleitet werden und auch auf unterschiedlichsten Computersystemen in Browsern betrachtet werden können sollen, ist es am besten, wenn man sich bei der Vergabe der Dateinamen an die alte DOS-Konvention hält - dann können sicher keine Probleme auftreten!

Weiters gibt es seit Anfang des WWW eine bis jetzt nicht gelöste Uneinigkeit der Erweiterungen für HTML-Dateien. Von den Systemen **Unix** und **Linux** kommt die Erweiterung `.html`, von **Windows-Systemen** die Erweiterung `.htm`. Mittlerweile dürfte eine gewisse Resignation eingekehrt sein, hier eine weltweite Vereinheitlichung durchzusetzen - es existieren also nach wie vor (und wahrscheinlich „bis in alle Ewigkeit!“) beide Dateierweiterungen nebeneinander. Aus der Erfahrung heraus und in Bezug auf das automatische Aufrufen von Web-Seiten kann nur geraten und empfohlen werden, die Erweiterung `.html` zu verwenden.

Wenn man die folgenden Konventionen für das Vergeben von Dateinamen beachtet, sollte man später keine Probleme mit seinen Web-Projekten haben.

1. Dateinamenlänge max. 8 Zeichen
2. alles klein schreiben
3. keine Leerzeichen oder Sonderzeichen verwenden - ALSO: **nur a - z und 0 - 9**
4. Erweiterung für den Dateinamen: `.html`



*Die Extension ist beim Speichern im Editor jeweils **explizit einzugeben!** (Der Editor vergibt ansonst selbständig die Extension `.txt` und die Seite wird nicht als HTML-Seite „gehandelt“.)*

Aufgabe 1: Die erste Web-Seite





HTML - PROGRAMMIERUNG

3. DIE GRUNDSTRUKTUR EINER HTML-SEITE

Wie bereits beschrieben stehen alle Anweisungen eines HTML-Codes zwischen den beiden Befehlen `<html>` und `</html>`. Nun gibt es eine weitere Unterteilung in zwei Bereiche: den HEAD und den BODY.

Im HEAD sind Informationen über die Seite enthalten wie z. B. der Seitentitel, der Name des Autors oder Schlüsselwörter für den Eintrag der eigenen Seite im Index von Suchmaschinen.

Im BODY steht das, was auf der Seite angezeigt werden soll - Text, Grafiken, Links, Formatierungen, Tabellen, .. - natürlich „verschlüsselt“ in HTML-Tags.

Die Grundstruktur einer HTML-Seite:

```
<html>
  <head>
  ...
  ...
</head>
<body>
  ...
  ...
</body>
</html>
```

Zur Deklaration des HEAD im HTML-Code werden die beiden Tags `<head>` und `</head>` verwendet. Sie bilden den „Container“ für den HEAD-Teil. Für die Festlegung des BODY-Teils dienen die Tags `<body>` und `</body>`.

Wie in der Abbildung gezeigt, sollte man sich bereits von Anfang an angewöhnen, eine gewisse **Strukturierung** in das Schreiben von HTML-Code einfließen zu lassen. Bei längeren und komplizierteren Seiten ist somit das nachträgliche Ändern, wie auch die Fehlersuche um vieles einfacher. Als einfache Grundregel sollte man beim Öffnungstag eines jeden Containers **zwei Zeichen einrücken**, wenn sich der „Inhalt“ dieses Containers über mehrere Zeilen erstreckt. Man erkennt dann einige Zeilen darunte genau, ob der Container richtig (bzw. überhaupt) geschlossen wurde.

Im Folgenden wird die links angegebene **Grundstruktur vorausgesetzt** und die jeweiligen Befehlszeilen, nur mehr aus dem Code herausgenommen, angegeben.

Der Seitentitel:

Mit Seitentitel ist die Zeichenkette gemeint, die in der Titelzeile des Browserfensters angezeigt wird, wenn die Web-Seite in einem Browser geladen wird. In der folgenden Abbildung wurde für die angezeigte Web-Seite der Seitentitel „HTML - Kurs Beispiel 2“ vergeben.



Die Angabe des Seitentitels steht im HTML-Code im HEAD zwischen den Tags `<title>` und `</title>`. Im Code einer Seite würde der Title also wie folgt festgelegt:

```
...
  <head>
    <title>HTML-Kurs Beispiel 2</title>
  </head>
...
```



Aufgabe 2: Eine Web-Seite mit Seitentitel





4. DER HEAD

Als ein Beispiel für eine Anweisung die im HEAD steht, wurde bereits der Seitentitel erklärt. Weiters befinden sich im HEAD weiter Anweisungen, die für das auf der Web-Seite Angezeigte zwar unerheblich sind, für das „Gefunden-werden“ der Seite bei Suchmaschinen aber von größter Wichtigkeit sind.

Diese Informationen werden in sogenannten Meta-Tags abgelegt. Diese sind nicht zu verwechseln mit „normalen“ Tags, die direkten Einfluss auf die Ausgabe der Web-Seite haben. Hier sind 4 Zeilen angegeben, die man unbedingt im HEAD einer Web-Seite plazieren sollte. Diese folgenden angegebenen Code-Zeilen sollen ohne weitere Erklärung bleiben und sollten sich soweit selbst erklären. Die Schreibweise muss natürlich bis auf die selbst zu vergebenden Begriffe exakt eingehalten werden.

META-Tags mit Informationen über die Web-Seite:

```
...
<head>
  ...
  <meta name="author" content="Dr. Leander Brandl">
  <meta name="publisher" content="Universität Graz - Fachdidaktik der Physik">
  <meta name="keywords" content="HTML, HTML-Kurs, HTML-Anleitung">
  <meta name="description" content="Informationen, Beispiele, Übungen, Lösungen">
  ...
</head>
...
```

„**author**“: hier „verewigt“ sich meist der Ersteller der Web-Seite

„**publisher**“: in diesem META - Tag wird der Veröffentlichter der Seite eingetragen

„**keywords**“: diese Schlüsselwörter werden bei der Eingabe von Suchbegriffen bei Suchmaschinen vorrangig durchsucht - wenn sie vorhanden sind!

„**description**“: diese Begriffe werden bei der Ausgabe von Suchergebnissen bei Suchmaschinen zusätzlich angegeben

Aufgabe 3: Web-Seite mit META-Tags





5. FARBEN AUF EINER WEB-SEITE

Farbangaben können unter HTML auf zwei unterschiedliche Varianten durchgeführt werden. Einerseits durch die Angabe eines **expliziten Farbnamens** oder andererseits durch die Angabe eines **RGB-Wertes** (RGB = Rot / Grün / Blau) in hexadezimaler Form.

1. explizite Farbbezeichnungen unter HTML:

Hier sollen die **16 Grundfarben** angegeben werden, die mit Sicherheit auf jedem Computersystem auch so angezeigt werden, wie Sie sie bei sich in der Entwicklungsphase sehen. Nebenan sind auch die RGB-Werte für die unter 2. besprochenen Farbangaben abgedruckt.

Farbe	Farbbezeichnung	Rotwert	Grünwert	Blauwert
Weiß	white	FF	FF	FF
Silber	silver	C0	C0	C0
Grau	gray	80	80	80
Schwarz	black	00	00	00
Rostbraun	maroon	80	00	00
Violett	purple	80	00	80
Rot	red	FF	00	00
Pink	fuchsia	FF	00	FF
Gelb	yellow	FF	FF	00
Grün	green	00	80	00
Hellgrün	lime	00	FF	00
Olivgrün	olive	80	80	00
Blau	blue	00	00	FF
Hellblau	aqua	00	FF	FF
Marineblau	navy	00	00	80
Dunkeltürkis	teal	00	80	80

2. Farbangaben in hexadezimaler Form:

Die Angabe eines Farbwertes in hexadezimaler Form erlaubt ein genaues „Zusammenmischen“ einer Farbe. Allerdings sollte man bedenken, dass die Darstellung der Farben beim Betrachter von dessen Computer abhängt und somit auch stark von dem abweichen kann, was Sie festgelegt haben.

Jede in Computern verarbeitete Farbe besteht aus einer Mischung von Intensitäten der 3 Grundfarben Rot, Blau und Grün. In dieser Reihenfolge geschieht auch die Festlegung durch Zahlenwerte. In HTML wird dies in hexadezimaler Schreibweise getan, wobei voran das Zeichen „#“ gestellt wird. Die Farbangabe für Rot würde lauten: „#FF0000“.

Bei der Angabe des jeweiligen Farbwertes für Rot, Grün oder Blau können diese in 256 verschiedenen Intensitäten dazugemischt werden. Diese dezimalen Werte 0 - 255 entsprechen in hexadezimaler Form 0 - FF. Damit man auch ohne große Umrechnungskennnisse zwischen den beiden Zahlenformaten „farbenprächtig“ mischen kann, finden Sie im Anhang A eine Tabelle, in der sie zu den Dezimalzahlen 0 - 255 die entsprechenden Hexadezimalzahlen finden.

Beispiel: Sie möchten folgende Farbe mischen und probieren mit den folgenden dezimalen Werten:

intensives Rot - also 255, etwas Grün - z.B. 35 und recht viel Blau - z.B. 190

Der hexadezimale Farbcode lautet also: #FF23BE (siehe Tabelle Anhang A)



6. GRAFIKEN AUF EINER WEB-SEITE

1. Grafik - Formate:

Die Grafiken, die man auf Web-Seiten verwenden will, müssen entweder im Grafik-Format **.jpg** oder **.gif** vorliegen. Diese beiden Formate werden heute von allen gängigen Bildverarbeitungsprogrammen unterstützt – man kann also jede Art von Grafik, Bild oder Foto mit einem herkömmlichen Bildbearbeitungsprogramm in eines der beiden Formate bringen. Einfach die gewünschte Grafik laden – im Bildbearbeitungsprogramm auf **Datei --> Speichern unter...** gehen – dort den Dateityp **.jpg** oder **.gif** auswählen und auf **Speichern** klicken. Stellt sich nur die Frage, welches Format man verwenden soll – beide speichern die Bilddaten komprimiert ab, liefern aber je nach Bildmaterial unterschiedliche Ergebnisse in Qualität und Dateigröße:

Das „gif“ – Format: Dieses Format sollte verwendet werden für Bildmaterial das aus wenigen, zum Teil regelmäßigen, gleichfarbigen Bereichen besteht (z. B. Balkendiagramme, Comics,...) und kann 256 Farben darstellen. Weiters stellt dieses Format die Möglichkeit zur Verfügung, eine Farbe als transparent zu deklarieren – bei der Ausgabe dieser Grafik erscheint dann im Bereich der transparenten Farbe der Hintergrund der Web-Seite. Dieses Format wird auch für animierte Grafiken verwendet: Wie bei einem Zeichentrickfilm werden die Einzelbilder einer Animation in einer Bilddatei abgespeichert.

Das „.jpg“ – Format: Dieses Format liefert bessere Ergebnisse als das „gif“-Format bei Bildern mit unregelmäßigen Farbinformationen (z. B. Fotos) und kann 16,7 Millionen Farben darstellen.

Wenn man sich unschlüssig ist, welches Format besser für ein spezielles Bild oder eine spezielle Grafik ist, sollte man ein und dasselbe Bild in beiden Formaten abspeichern, die Dateigrößen vergleichen und auch beide Dateien in einem Browser öffnen, um die Anzeige-Ergebnisse zu vergleichen.

ACHTUNG (Dateigröße): Wenn man auf seinen eigenen Web-Seiten Grafiken, Bilder oder Fotos platziert, sollte man immer daran denken, dass die Größe und Anzahl der Grafiken eine erhebliche Auswirkung auf die Übertragungsgeschwindigkeit einer Web-Seite hat. Je mehr Bilddateien mitübertragen werden müssen, umso „lahmer“ geht der Seitenaufbau vor sich.

2. Hintergrund-Grafiken:

Grafiken, die dazu verwendet werden können, um den Hintergrund einer Web-Seite zu erzeugen, werden als „Texturen“ bezeichnet. Da es viel zu lange dauern würde den gesamten Hintergrund einer Web-Seite als Grafik zu übertragen, hat man sich auf die automatische Erzeugung mit Hilfe von Texturen geeinigt. Diese sind so gestaltet, dass es fließende, nicht zu erkennende Übergänge zwischen dem linken und dem rechten Rand gibt, wenn man diese Textur-Grafik mehrmals nebeneinander anordnet. Dasselbe gilt für den oberen und unteren Rand. Dazu ein Beispiel:



... die Textur



... die Textur - viermal nebeneinander angeordnet

Es braucht bei der Übertragung der Web-Seite über das Internet also nur diese kleine Textur-Grafik mitübertragen werden, der Browser setzt dann daraus den gesamten Hintergrund zusammen, indem er diese Grafik so oft nebeneinander und untereinander anordnet, bis die gesamte Seite bedeckt ist. Diese Grafiken können natürlich wieder in den beiden Formaten **.jpg** oder **.gif** vorliegen.

Im Internet gibt es Sammlungen mit tausenden solcher Texturen. Eine ausgezeichnete Sammlung, in der man sicher alles findet „was das Herz begehrt“ gibt es unter **<http://www.grsites.com>**



7. DER BODY

Im Body einer HTML-Datei steht wie schon erwähnt das, was auf der Web-Seite angezeigt werden soll. Der `<body>` - Tag dient aber nicht nur ausschließlich dazu um den Body einzuleiten. Mit diesem Tag werden auch die Grundeinstellungen für eine Seite wie Schriftfarbe, Hintergrundfarbe, Linkfarbe, usw. festgelegt.

Im Folgenden wird auch die Schreibweise ersichtlich, wie Eigenschaften eines Tags in HTML Werte zugewiesen werden. Die Werte einer Eigenschaft werden der Eigenschaft mit einem „=" zugewiesen und stehen selbst unter Anführungszeichen. Verwendet man mehrere Eigenschaften eines Tags, so trennt man die verschiedenen Eigenschaften, deren Reihenfolge keine Rolle spielt, durch ein Leerzeichen.

Eigenschaften des `<body>` - Tags:

- `bgcolor="red"` ... die Hintergrundfarbe der Web-Seite wird z.B. auf Rot gestellt
- `text="green"` ... mit dieser Eigenschaft wird die globale Textfarbe z.B. mit Grün festgelegt
- `link="blue"` ... die Farbe für die Links
- `alink="green"` ... die Farbe für Links beim Anklicken
- `vlink="purple"` ... die Farbe für Links, die bereits einmal benutzt wurden

Die Farbangaben können natürlich auch in hexadezimaler Form erfolgen (siehe 5.!) Die Festlegung der Hintergrundfarbe würde dann lauten: `bgcolor="#FF0000"`

`background="steine.jpg"` ... die Hintergrundgrafik (dabei sollte es sich möglichst um eine Textur handeln) aus der der Hintergrund der Seite erzeugt wird. Diese Grafik-Datei befindet sich **im gleichen Ordner** wie die HTML-Datei. Zusätzlich sollte man aber auch die Hintergrundfarbe in einem **ähnlichen Farbton** einstellen, damit die Seite bereits in der gewünschten Farbgebung angezeigt wird, auch wenn die Hintergrundgrafik noch nicht vollständig übertragen ist.

Ein Beispiel für die Festlegung aller Farben einer Web-Seite:

```
...
</head>
<body bgcolor="gray" text="blue" link="#00FF00" alink="green" vlink="black">
...
</body>
</html>
```



In diesem Beispiel wird eine Hintergrundgrafik und die Textfarbe festgelegt:

```
...
</head>
<body bgcolor="gray" background="steine.jpg" text="blue">
...
</body>
</html>
```



Aufgabe 4: Web-Seite mit Farbeinstellungen





HTML - PROGRAMMIERUNG

8. TEXTAUSGABE - ZEILENUMBRUCH, ZENTRIEREN

Die erste Textzeile wurde bereits bei den ersten drei Beispielen ausgegeben. Nun verhält sich jeder Browser so, dass der allen Text hintereinander anordnet und am rechten Rand des Browserfensters einen automatischen Zeilenumbruch durchführt.

Aufgabe 5: Automatischer Zeilenumbruch bei der Ausgabe von Text



Erzwingen eines Zeilenumbruches:

Soll eine Textzeile **vor** dem rechten Rand des Browserfensters enden, so muss an der entsprechenden Stelle ein Zeilenumbruch mit dem `
` - Tag im Code festgelegt werden. Die Ausgabe des anschließend im Code stehenden Textes erfolgt dann in der nächsten Zeile. Weiters lassen sich so mit dem `
` - Tag Leerzeilen erzeugen.

```

...
<body>
  ...
  Dieser Text steht in der ersten Zeile -
  <br>
  diese Wörter werden in der nächsten Zeile ausgegeben.
  ...
</body>
...

```



Aufgabe 6: Beabsichtigte Ausgabe von Text in mehreren Zeilen



Zentrieren von Text, Grafiken, Links, Linien, Tabellen, ... auf Web-Seiten:

Zum Zentrieren von Elementen auf Web-Seiten dienen die beiden Tags `<center>` und `</center>`. Alles, was zwischen diesen beiden Tags steht, wird auf der Seite zentriert angezeigt. Zu beachten ist, dass die zentrierte Ausgabe erst endet, wenn der beendende Tag `</center>` gesetzt wird.

```

...
<body>
  ...
  Dieser Text wird linksbündig ausgegeben.
  <br>
  <center>
  ZENTRIERTER TEXT
  ...
  </center>
  ...
</body>
...

```

alles was sich hier im Code „befindet“, wird auf der Seite zentriert ausgegeben



Aufgabe 7: Ausgabe von zentriertem Text in mehreren Zeilen





9. TEXTAUSGABE - ÜBERSCHRIFTEN

Für die Ausgabe von Überschriften sind in HTML 6 Tags vorgesehen, die die Schriftgröße für den eingeschlossenen Text festlegen und einen Zeilenumbruch erzeugen. Weiters gibt es die Möglichkeit, durch die Eigenschaft *align* die Ausrichtung der Überschrift einzustellen.

Die Tags für das Erzeugen von Überschriften:

`<h1> ... </h1>` ... Überschrift 1. Ordnung (größter Text)
`<h2> ... </h2>` ... Überschrift 2. Ordnung
`<h3> ... </h3>` ... Überschrift 3. Ordnung
`<h4> ... </h4>` ... Überschrift 4. Ordnung
`<h5> ... </h5>` ... Überschrift 5. Ordnung
`<h6> ... </h6>` ... Überschrift 6. Ordnung (kleinster Text)

Die Eigenschaft „align“ der Überschriften-Tags:

`align="left"` ... diese Ausrichtung ist standardmäßig eingestellt
`align="right"` ... die Überschrift wird rechtsbündig ausgerichtet
`align="center"` ... die Überschrift wird zentriert ausgerichtet - man erspart sich das Zentrieren durch zusätzliche `<center> ... </center>` Tags.

Ein Beispiel für das Erzeugen von drei unterschiedlichen Überschriften:

```
...  
<body>  
  ...  
  <h1>Überschrift 1. Ordnung (linksbündig)  
  ...  
  <h3 align="center">Überschrift 3. Ordnung (zentriert)</h3>  
  ...  
  <h6 align="right">Überschrift 6. Ordnung (rechtsbündig)</h6>  
  ...  
</body>  
</html>
```



Aufgabe 8: Überschriften auf einer Web-Seite





HTML - PROGRAMMIERUNG

10. TEXTAUSGABE - SONDERZEICHEN

Durch den heterogenen Standard von Computern und Betriebssystemen im Internet wurde bereits in der sehr frühen Phase des Internets ein international genormter Zeichensatz zum Austausch von Zeichen vereinbart. Dieser basierte natürlich auf dem amerikanischen Zeichensatz und beinhaltet deshalb nicht die Sonderzeichen der jeweiligen Sprachen - bei Deutsch die Umlaute, bei Französisch die Akzente, usw.

Allerdings wurde bei der Entwicklung von HTML eine Möglichkeit vorgesehen, die sprachenspezifischen Sonderzeichen auf Web-Seiten darzustellen - auch wenn das Implementieren im HTML-Code etwas umständlich wirkt. An Stelle eines Sonderzeichens muss eine dafür festgelegte Zeichenkette geschrieben werden - man nennt diese Darstellung **Maskierung**. Bei diesen Maskierungen ist genau auf die **Groß- und Kleinschreibung** zu achten!

Die Maskierungen für Sonderzeichen der deutschen Sprache:

Zeichen	Maskierung	Beschreibung
ä	ä	kleines ä
Ä	Ä	großes Ä
ö	ö	kleines ö
Ö	Ö	großes Ö
ü	ü	kleines ü
Ü	Ü	großes Ü
ß	ß	scharfes S
<	<	less than
>	>	greater than
&	&	geschäftliches „und“

Die Maskierungen für weitere nicht im Standardzeichensatz enthaltene Sonderzeichen:

Zeichen	Maskierung	Beschreibung
	 	erzwungenes Leerzeichen
©	©	Copyright - Symbol
®	®	Registered - Symbol
μ	µ	Mikro - Vorsilbe
π	¶	griechischer Buchstabe Pi
·	·	Multiplikationspunkt
±	±	PlusMinus
°	°	Grad
1/4	¼	ein Viertel
1/2	½	ein Halb
3/4	¾	drei Viertel

Ein Beispiel:

Der Satz: „Jürgen übt viel öfter, seit die übrigen Musiker gesagt haben, daß er sonst Ärger bekommt.“ muss also im HTML-Code folgendermaßen geschrieben werden.

```
...
J&uuml;rgen &uuml;bt viel &ouml;fter, seit die &uuml;brigen Musiker
gesagt haben, da&szlig; er sonst &Auml;rger bekommt.
...
```



Aufgabe 9: Ausgabe von Sonderzeichen





11. TEXTAUSGABE - TEXTATTRIBUTE

Textformatierung und die Verwendung verschiedener Schrifttypen ist unter HTML eigentlich immer noch eine heikle Angelegenheit. Gemeint ist dabei, dass die Ausgabe (und somit das Aussehen!) einer Seite abhängig ist von den Einstellungen und installierten Schriften am „Betrachter“ - Computer. Mit der Einführung der **Cascading Style Sheets**, die leider noch immer nicht zufriedenstellend in den beiden Standardbrowsern integriert sind (und zum Teil auch unterschiedlich ausgewertet werden) sollte diesem Übel ein Ende gemacht werden - man wird sehen!?!?!?

Nichts desto trotz lässt sich mit den **traditionellen gestalterischen Mitteln** einiges erreichen. Die in diesem Kurs besprochenen Befehle zur Textausgabe und -formatierung entsprechen dem Standard und **garantieren** eine Ausgabe am „Betrachter“ - Computer, die der von Ihnen festgelegten exakt entspricht.

Unter diesem Punkt sollen Tags zur Steuerung von **bekannten Textattributen** wie fett, kursiv, ... angegeben werden. Die Zeichen, die in dem gewünschten Attribut ausgegeben werden sollen, werden einfach **zwischen** den Start- und End-Tag des jeweiligen Attributs gesetzt. Auch eine Ausgabe eines Textes mit **mehreren Attributen** ist möglich, dazu wird dieser Text einfach von zwei Tag-Containern umgeben.

HTML-Tags für das Erzeugen von Textattributen:

Tag	Beschreibung
<code> ... </code>	fett („bold“)
<code><i> ... </i></code>	kursiv („italic“)
<code><u> ... </u></code>	unterstrichen („underline“)
<code><strike> ... </strike></code>	durchgestrichen
<code><big> ... </big></code>	relativ im Vergleich zur Standardschrift größere Schriftgröße
<code><small> ... </small></code>	relativ im Vergleich zur Standardschrift kleinere Schriftgröße
<code><sup> ... </sup></code>	hochgestellt
<code><sub> ... </sub></code>	tiefgestellt

Ein Beispiel:

Der Satz: „In diesem **Satz** haben einige *Wörter* verschiedene ~~Textattribute~~.“ muss also im HTML-Code folgendermaßen geschrieben werden.

```
...  
In diesem Satz haben einige Wörter verschiedene  
Textattribute.  
...
```



Aufgabe 10: Ausgabe von Text mit verschiedenen Textattributen





12. TEXTAUSGABE - GRÖSSE, FARBEN, ABSÄTZE

Festlegen von Schriftfarbe und Schriftgröße:

Das Festlegen von Schriftfarbe und Schriftgröße geschieht mit dem `` - Tag. Alles, was zwischen den beiden `` ... `` Tags steht, wird in der jeweiligen Farbe und Größe ausgegeben.

Die Angabe der Größe bezieht sich dabei relativ zur eingestellten Standardgröße eines Browsers. Die Größe wird in der Eigenschaft `size` angegeben und kann zwischen 1 und 7 liegen. 3 entspricht der Standardgröße, 1 und 2 stellen kleinere Schriftgrößen dar, 4 - 7 größere. Die Angabe der Schriftfarbe steht in der Eigenschaft `color`. Wie diese Farbwerte angegeben werden, ist unter 5. genau beschrieben.

Eigenschaften des `` - Tags:

`size="6"` ... die Schriftgröße relativ zur Standardgröße, die Werte liegen zw. "1" und "7"

`color="green"` ... mit dieser Eigenschaft wird die Textfarbe auf Grün gesetzt

Ein Beispiel:

Der Satz: „Hallo, ich bin ein größer geschriebener grüner Satz!“ würde unter Verwendung des `` - Tags im HTML-Code folgendermaßen aussehen:

```
...
<font size="6" color="green">
  Hallo, ich bin ein größer geschriebener grüner Satz!
</font>
...
```



Aufgabe 11: Ausgabe von Text unterschiedlicher Größe und Farbe



Festlegen und Ausrichtung von Absätzen:

Absätze müssen unter HTML wie Zeilenumbrüche explizit mit dem `<p>` - Tag definiert werden. Bei dieser Definition kann auch die Ausrichtung des Textes in dem Absatz in der Eigenschaft `align` festgelegt werden.

`align="right"` ... die Ausrichtung des Textes erfolgt rechtsbündig. Die Werte dieser Eigenschaft lauten "left", "right", "center" und "justify". "justify" steht für die Ausrichtung als Blocksatz, wird aber (noch) nicht von allen Browsern unterstützt.

```
...
<p align="right">
  Hier handelt es sich nun um rechtsbündigen Text Text Text Text
  Text Text Text Text Text Text Text Text
  Zu beachten ist, dass es bis zum Ende des Absatzes keinen br-Tag
  gibt. Text Text Text Text Text
</p>
...
```



Aufgabe 12: Ausgabe von ausgerichtetem Text in Absätzen





13. TEXTAUSGABE - AUFZÄHLUNGEN

Das Aufzählen von mehreren Punkten lässt sich in HTML sehr einfach realisieren. Neben geometrischen Symbolen, die vor den Zeilen stehen, kann man auch eine automatische Nummerierung auf verschiedene Arten erzeugen.

Unnummerierte Listen, Aufzählungen:

Gestartet wird eine nicht nummerierte Aufzählung durch den `` - Tag. In diesem Tag wird in der Eigenschaft `type` die Art des Symbols vor den Listeneinträgen festgelegt. Dann folgen die Listeneinträge, die jeweils von `` und `` umgeben sind. Nach dem letzten Listeneintrag steht natürlich ein ``

`type="square"` ... die Symbole vor den Listeneinträgen sind Quadrate, die Werte für diese Eigenschaft sind: `"disc"`, `"square"` und `"circle"`

Ein Beispiel:

Durch den folgenden HTML-Code wird eine unnummerierte Aufzählung mit 4 Einträgen erzeugt. Vor den einzelnen Einträgen befindet sich ein schwarzer (Aufzählungs-) Punkt.

```
...
<ul type="disc">
  <li>erste Zeile</li>
  <li>zweite Zeile</li>
  <li>dritte Zeile</li>
  <li>vierte Zeile</li>
</ul>
...
```



Nummerierte Listen, Aufzählungen:

Gestartet wird eine nicht nummerierte Aufzählung durch den `` - Tag. Dann folgen die Listeneinträge, die jeweils von `` und `` umgeben sind. Im ersten dieser Listeneinträge wird bei der Eigenschaft `type` des `` - Tags die Art der Nummerierung festgelegt. Nach dem letzten Listeneintrag steht natürlich ein ``

`type="A"` ... die Symbole vor den Listeneinträgen sind Quadrate, die Werte für diese Eigenschaft sind:

- `"1"` ... Arabische Ziffern (1, 2, 3, 4, ...)
- `"a"` ... Alphabetische Kleinbuchstaben (a, b, c, d, ...)
- `"A"` ... Alphabetische Großbuchstaben (A, B, C, D, ...)
- `"i"` ... Römische Ziffern in Kleinschreibung (i, ii, iii, iv, ...)
- `"I"` ... Römische Ziffern in Großschreibung (I, II, III, VI, ...)

Ein Beispiel:

Durch den folgenden HTML-Code wird eine nummerierte Aufzählung mit 3 Einträgen erzeugt. Vor den einzelnen Einträgen befindet sich eine fortschreitende Nummerierung mit a., b. und c.

```
...
<ol>
  <li type="a">erste Zeile</li>
  <li>zweite Zeile</li>
  <li>dritte Zeile</li>
</ol>
...
```



Aufgabe 13: Ausgabe von nummerierten und nicht nummerierten Aufzählungen





HTML - PROGRAMMIERUNG

14. TEXTAUSGABE - EINRÜCKEN

Damit der auf einer Web-Seite ausgegebene Text nicht so am linken bzw. rechten Rand „klebt“, gibt es die Möglichkeit eines automatischen Einrückens. Der Textausgabebereich erhält dadurch einen linken und rechten Rand gleicher Breite. Die exakte Breite des Einrückens ist aber in verschiedenen Browsern (und hier sind speziell der Netscape Navigator und der Microsoft Internet Explorer gemeint) unterschiedlich festgelegt und führt deshalb auch zu unterschiedlichen Ergebnissen.

Erzeugt wird dieser begrenzte Ausgabebereich durch die beiden Tags `<blockquote>` und `</blockquote>`. Alles, was nun im HTML-Code zwischen diesen beiden liegt, wird nun in einem schmäleren Bereich ausgegeben. Am besten demonstriert das folgende Beispiel die Funktion von `<blockquote>` und `</blockquote>`.

```
<html>
<head>
  <title>HTML-Kurs Beispiel</title>
</head>
<body>
  <h1 align="center">Darstellung in eingeschränktem Bereich</h1>
  Der Text vor der blockquote-Anweisung wird direkt am linken Rand ausgegeben.
  <blockquote>
    <b>W</b>o?<br>
    <b>W</b>ann?<br>
    <b>W</b>as?<br>
    <b>W</b>arum?<br>
    <br>
    <br>
    <p align="right">
      <b>W</b>o?<br>
      <b>W</b>?<br>
      <b>W</b>as?<br>
      <b>W</b>arum?<br>
    </p>
  </blockquote>
  Der Text nach der blockquote-Anweisung wird wieder direkt am linken Rand ausgegeben.
</body>
</html>
```



Die im Code angegebene Verwendung von `<blockquote>` führt zu folgender Ausgabe:



Aufgabe 14: Die Verwendung von `<blockquote>`





HTML - PROGRAMMIERUNG

15. ANZEIGEN VON GRAFIKEN - RAHMEN

Das Anzeigen von Grafiken auf einer Web-Seite ist im Grunde kein schwieriges Unterfangen. Bevor die Befehlsstruktur erklärt wird, sollte man sich nochmals Kapitel 6 durchlesen. Die Grafiken müssen wieder in den beiden Formaten **.jpg** oder **.gif** vorliegen. Die Einbindung in den HTML-Code erfolgt an der gewünschten Position mit dem **** - Tag. Der Eigenschaft **src** wird der Dateiname der Grafik zugewiesen.

Obwohl es Eigenschaften zum Festlegen der Grafikgröße gibt, kann hier nur geraten werden, dass man Grafiken in einem Bildbearbeitungsprogramm bereits so auf die gewünschte Größe bringt, wie man sie angezeigt bekommen möchte. Nachträgliche Vergrößerungen oder Verkleinerungen durch den **** - Tag haben schlechte Qualität oder unnötig große Grafikdateien zur Folge.

Das Anzeigen der Datei „grafik8.jpg“, die sich im gleichen Verzeichnis wie die html-Datei befindet:

```
...  
<body>  
...  
  
...  
</body>  
...
```



Aufgabe 15: Die erste Ausgabe einer Grafik



Rahmen um eine Grafik:

Damit man einen Rahmen um eine Grafik erzeugen kann, muss man der Eigenschaft **border** des **** - Tags einen *Zahlenwert* - der die *Randbreite in Pixel* angibt - zuweisen. Diese Eigenschaft muss explizit auf 0 gesetzt werden (**border="0"**), wenn bei der Verwendung einer Grafik als Link automatisch ein Rahmen um diese gezeichnet wird.

Im folgenden Beispiel wird ein Rahmen mit der Breite von 3 Pixel um die Grafik gezeichnet:

```
...  
  
...
```



Abstände zwischen Grafiken und umgebenden Objekten:

Da Grafiken auf einer HTML-Seite wie ein Textzeichen behandelt werden, kann es oft nötig sein, die umgebenden Objekte etwas von der Grafik fernzuhalten, damit diese optisch nicht direkt an der Grafik „kleben“. Dazu gibt es zwei Eigenschaften des **** - Tags, die den horizontalen und vertikalen Abstand zwischen einer Grafik und ihrer Umgebung festlegen. Den beiden Eigenschaften **hspace** und **vspace** werden *Zahlenwerte in Pixel* zugewiesen, die den Abstand zu den umgebenden Objekten angeben.

Der Abstand zwischen Grafik und seitlichem Text beträgt im folgenden Beispiel 5 Pixel, zwischen Grafik und oberem bzw. unterem Text 10 Pixel:

```
...  
  
...
```



Aufgabe 16: Rahmen um eine Grafik





16. ANZEIGEN VON GRAFIKEN - AUSRICHTUNG

Das Ausrichten von Grafiken auf einer Web-Seite folgt eigenen Gesetzmäßigkeiten. Am besten, man löst Formatierungsprobleme, in denen Text und Grafik aufeinandertreffen, mit Hilfe von Tabellen - aber dazu später.

Will man die Ausrichtung von Grafiken nur unter Verwendung des `` - Tags durchführen, so ist zwischen horizontaler und vertikaler Ausrichtung zu unterscheiden!

Horizontale Ausrichtung von Grafiken:

linksbündig:

```

```

zentriert:

```
<center>  </center>
```

rechtsbündig:

```

```



Aufgabe 17: Horizontale Ausrichtung einer Grafik



Vertikale Ausrichtung einer Grafik am umgebenden Text:

Da eine Grafik wie ein Textzeichen behandelt wird, meist aber nicht die Höhe des umgebenden Textes haben wird, gibt es eine Möglichkeit einzustellen, wie die Grafik im Vergleich zum seitlichen Text vertikal ausgerichtet werden soll. Dies geschieht wiederum in der Eigenschaft `align` des `` - Tags.

oberer Textrand = oberer Rand der Grafik:

```

```

Text befindet sich auf halber Höhe der Grafik:

```

```



unterer Textrand = unterer Rand der Grafik:

```

```

Aufgabe 18: Vertikale Ausrichtung einer Grafik am Text





17. TRENNLINIEN

Als einfaches Hilfsmittel zur optischen Abgrenzung von Bereichen auf einer Web-Seite sind Trennlinien bekannt. Diese Trennlinien werden mit dem `<hr>` - Tag in den HTML-Code eingefügt und erzeugen zusätzlich einen Zeilenvorschub.

Um diese Trennlinien verschieden zu gestalten, stehen mehrere Möglichkeiten zur Verfügung. Ändern lassen sich die Breite, Höhe, Ausrichtung und die 3D-Darstellung. Bei der Angabe der Breite stehen zwar relative und absolute Breitenangaben zur Verfügung, man sollte aber auf Grund der Skalierbarkeit der Webseiten ausschließlich relative Breitenangaben verwenden.

Eigenschaften des `<hr>` - Tags:

`width="80%"` ... die Angabe der Breite der Trennlinie relativ zur Breite des Browserfensters

`size="4"` ... die Höhe der Trennlinie in Pixel. Die Voreinstellung beträgt 2 Pixel.

`align="left"` ... die Ausrichtung einer Trennlinie. Ein Setzen dieser Eigenschaft zeigt natürlich nur Wirkung, wenn die Trennlinie nicht die volle Länge besitzt. Die möglichen Werte sind `"left"`, `"right"` und `"center"`.

`noshade` ... wird diese Eigenschaft angegeben, dann wird die Linie ohne 3D-Effekt angezeigt.

Der Code für eine zentrierte Trennlinie mit einer Breite von 50% des Browserfensters:

```
...
<body>
  ...
  <hr width="50%" align="center">
  ....
</body>
...
```



Aufgabe 19: Verschiedene Trennlinien





HTML - PROGRAMMIERUNG

18. LINKS - ABSOLUT

Das was „normalen“ Text von HyperText unterscheidet, sind funktionale Elemente im Text, die den angezeigten Text beeinflussen - einfach ausgedrückt: die Links. Diese Verknüpfungen, Verweise oder wie sie im Deutschen auch immer bezeichnet werden, stellen das zentrale Element auf Web-Seiten dar.

Das Erzeugen eines Links im HTML-Code ist eigentlich recht einfach - doch gibt es viele Möglichkeiten zu „verlinken“. Die Angabe eines Links erfolgt durch den `<a>` - Tag, der durch `` abgeschlossen werden muss. Die Eigenschaft, die natürlich unbedingt gesetzt werden muss und das Ziel der Verknüpfung angibt, heißt `href`.

Die grundsätzliche Unterscheidung von Links bezieht sich auf die Ziele der Verknüpfungen. Soll der Link auf eine Seite am eigenen Web-Server verweisen oder soll dieser das Aufrufen einer Web-Seite von einem anderen Web-Server bewirken?!

Der `<a>` - Tag:

```
...
<body>
...
  <a href="http://www.sony.com">Link zu Sony</a>
...
</body>
...
```

der Start-Tag die absolute Adresse der verlinkten Web-Seite der Text, der als Link angezeigt wird der abschließende Tag

Absolute Links:

Will man zu Web-Seiten auf anderen Web-Servern verlinken, so muss bei der Eigenschaft `href` die exakte Adresse dieser Seite angegeben werden - davor ist unbedingt `http://` zu setzen!!! Wird nicht nur zur DNS-Adresse eine Web-Servers sondern direkt zu einer Web-Seite auf einem Web-Server verwiesen so würde dies im Code wie folgt aussehen:

```
<a href="http://w4.brgkepler.asn-graz.ac.at/html/kurs/seite27.html">Kurs - Links</a>
```

e-mail - Links:

Bekannterweise gibt es auf Web-Seiten Links, die das direkte Absenden eines e-mails an die angegebene Person aus dem Standard-e-mail-Programm des Computers ermöglichen. Diese e-mail-Links werden auch mit dem `<a>` - Tag erzeugt. In der Eigenschaft `href` wird die e-mail-Adresse des Empfängers und davor das Schlüsselwort `mailto:` angegeben

```
...
<body>
...
  <a href="mailto:lbrandl@sime.com">Link zu L.Brandl</a>
...
</body>
...
```

Aufgabe 20: Absolute Links





HTML - PROGRAMMIERUNG

19. LINKS - RELATIV

Relative Links:

... beziehen sich auf HTML-Dateien, die zum gleichen Web-Projekt gehören und sich auf demselben Server befinden. Meist wird man bei aufwendigen Projekten eine vernünftige Verzeichnisstruktur für HTML-Dateien, Bilder, ... anlegen. Auf diese Verzeichnisse (und darin enthaltenen Dateien) kann man in der Eigenschaft `href` zugreifen.

Link zu einer Datei, die sich im gleichen Verzeichnis befindet:

Am einfachsten ist es, wenn sich die zu verlinkende Datei im gleichen Verzeichnis befindet. In der Eigenschaft `href` wird dann einfach der Dateiname der Datei, zu der der Link führen soll, angegeben,.

```
...
<body>
  ...
  <a href="seite2.html">Link zur nächsten Seite</a>
  ...
</body>
...
```



Aufgabe 21: Relative Links



Die großen Vorteile, die in diesen relativen Angaben liegen, setzen aber voraus, dass man die Verzeichnisstruktur seines Projekts genau vor Augen hat. Man kann im Prinzip beliebig von einem Verzeichnis in ein anderes verlinken.

Dazu gibt man den Pfad vom aktuellen Verzeichnis zu der zu verlinkenden HTML-Datei an. Will man von einer HTML-Datei, die sich in einem Unterordner befindet, zu einer Datei in einem übergeordneten Ordner verlinken, so geschieht dies durch die Angabe von `../` im Pfad zu der zu verlinkenden Datei.

Angabe des Pfades zu einer HTML-Datei, die sich in einem untergeordneten Ordner befindet:

Die Seite mit den Links befindet sich im Verzeichnis `htmlkurs`. Die Links zu den angegebenen Seiten müssten folgende Pfade enthalten.

```
href="beispiele/beispiele.html"
```

```
href="beispiele/bsp1_10/bsp3.html"
```



Angabe des Pfades zu einer HTML-Datei, die sich in einem übergeordneten Ordner befindet:

Die Seite mit den Links befindet sich im Verzeichnis `bsp11_20`. Die Links zu den angegebenen Seiten müssten folgende Pfade enthalten.

```
href="../../beispiele.html" (diese Datei befindet sich also im Ordner beispiele)
```

```
href="../../../index.html" (diese Datei befindet sich also im Ordner htmlkurs)
```





HTML - PROGRAMMIERUNG

20. LINKS - NEUES FENSTER - GRAFIKEN

Öffnen einer Web-Seite in einem neuen Browserfenster:

Will man die Web-Seite, zu der ein Link verweist, in einem neuen Browserfenster angezeigt bekommen, so muss man als Ziel für den Link ein neues Fenster angeben. Wo eine verlinkte Seite angezeigt werden soll, wird im `<a>` - Tag in der Eigenschaft `target` angegeben. Diese Eigenschaft wird später bei der Arbeit mit Frames eine tragende Rolle spielen.

Damit die verlinkte Seite in einem neuen Browserfenster angezeigt wird, weist man der Eigenschaft `target` den Wert `_blank` zu. Dies hat den Vorteil, dass die andere Seite in einem zweiten Browserfenster geöffnet bleibt und weiterhin zur Verfügung steht. Die bietet sich z. B. für Sammlungen von Links an, die dann nicht jedesmal neu geladen werden muss.

```
...  
<body>  
...  
    <a href="http://www.sony.com" target="_blank">Link zu Sony</a>  
...  
</body>  
...
```



Aufgabe 22: Anzeigen von Seiten in neuen Browserfenstern



Die Verwendung von Grafiken als Links:

Mit dem `<a>` - Tag ist es auch sehr leicht möglich, eine Grafik als Link zu verwenden. Zwischen `<a>` und `` wird statt des Textes einfach der Grafik Tag `` angegeben.

Zu beachten ist, dass bei einer Grafik, die zu einem Link „umfunktioniert“ wird, automatisch ein Rahmen um die Grafik gezogen wird. Will man also keine Rahmen um die Grafik erhalten, so muss der Rand mit `border="0"` explizit unterdrückt werden.

```
...  
<body>  
...  
    <a href="http://www.sony.com"></a>  
...  
</body>  
...
```



Aufgabe 23: Verwendung einer Grafik als Link





21. LINKS - INNERHALB EINER SEITE

Erstellt man sehr lange Webseiten, so sollte man dem Besucher die Möglichkeit geben, innerhalb dieser Seite leicht navigieren zu können - ohne dass dieser ständig mit den Scrollbalken auf- und abfahren muss.

Man könnte z. B. am Seitenanfang die Überschriften der Seite angeben und bei einem Klick darauf springt die Seite automatisch zu der jeweiligen Überschrift. Anders herum wäre es praktisch, wenn man von verschiedenen Stellen auf der Seite durch einen Klick wieder ganz nach oben kommen würde.

Diese Navigation innerhalb einer Seite wird auch mit dem `<a>` - Tag realisiert - nur: in der Eigenschaft `href` steht nicht eine HTML-Datei sondern ein Ziel innerhalb der Seite, das im HTML-Code festgelegt wird.

Erzeugen von Zielen innerhalb einer Seite:

Vor der Überschrift 3 wird auf der Seite ein Ziel „eingrichtet“. Dies geschieht mit dem `<a>` - Tag und der Vergabe eines Namen dieses Ziels:

```
...
<body>
  ...
  ...
  <a name="Abschnitt3">Überschrift 3</a>
  ...
</body>
...
```



Links zu Zielen innerhalb einer Seite:

Um zu einem Ziel zu springen wird einfach ein „gewöhnlicher“ Link erzeugt. In der Eigenschaft `href` wird der Name des Ziels mit einer vorangestellten `#` angegeben.

```
...
<body>
  Willkommen auf meiner Web-Seite
  ...
  <a href="#Abschnitt3">Hier gehts zu Überschrift 3</a>
  ...
</body>
...
```



Aufgabe 24: Links innerhalb einer Seite





22. TABELLEN - GRUNDSTRUKTUR

Tabellen dienen in HTML nicht nur dazu um Messwerte oder ähnliches tabellarisch darzustellen, sondern werden vielfach auch dazu verwendet, um komplexe Formatierungsprobleme zu lösen. Speziell verwendet man Tabellen dann, wenn es darum geht, Seiten trotz vieler verschiedener Objekte trotzdem noch skalierbar zu halten.

Im ersten Abschnitt soll die Grundstruktur des Erzeugens einer Tabelle in HTML klar werden. Im Grund gibt es dazu 3 HTML-Tags. Zuerst wird die Tabelle als solche im Code ausgewiesen - dies geschieht durch `<table>` ... `</table>`. Weiters werden mit `<tr>` ... `</tr>` die Zeilen erzeugt und mit `<td>` ... `</td>` die einzelnen Zellen in einer Zeile.

Beim Schreiben des HTML-Codes sollte man speziell bei Tabellen darauf achten, eine übersichtliche Struktur durch konsequentes Einrücken zu erzeugen. Das folgende Beispiel soll die Tabellen-Tags verständlich machen.

Erzeugen einer einfachen Tabelle mit 3 Zeilen und 2 Spalten:

```
...
<body>
  ...
  <table>
    <tr>
      <td>Zelle 1</td>
      <td>Zelle 2</td>
    </tr>
    <tr>
      <td>Zelle 3</td>
      <td>Zelle 4</td>
    </tr>
    <tr>
      <td>Zelle 5</td>
      <td>Zelle 6</td>
    </tr>
  </table>
  ...
</body>
...
```



Der angegebene HTML-Code würde eine Tabelle in der abgebildeten Form erzeugen. Klar ersichtlich ist im Code auch die Struktur der Tabelle durch eingerücktes Schreiben der Befehle.

Zelle 1	Zelle 2
Zelle 3	Zelle 4
Zelle 4	Zelle 5

Werden beim `<table>` - Tag keine weiteren Angaben gemacht, so wird die Größe der Tabelle in Höhe und Breite automatisch an den Inhalt der Zellen angepasst. Weiters wird die Tabelle am linken Rand des Browserfensters ausgerichtet. Eine erste Möglichkeit, eine Tabelle zu zentrieren, wäre natürlich, die Tabelle zwischen `<center>` ... `</center>` zu setzen.

Aufgabe 25: Die erste Tabelle





23. TABELLEN - RÄNDER - FARBEN - GRÖSSE

Wenn beim `<table>` - Tag nicht explizit ein Rand festgelegt wird, so wird die Tabelle ohne Begrenzungs- und Gitterlinien angezeigt. In HTML gibt es zwei Möglichkeiten, Einstellungen für diese Linien zu treffen. Einerseits kann die Breite des Randes, der die gesamte Tabelle begrenzt festgelegt, werden und andererseits ist es möglich, die Breite der Gitterlinien in der Tabelle zu beeinflussen. Weiters ist es möglich, Einstellungen für die Farbe des Randes wie auch des Hintergrunds zu tätigen.

Natürlich sind auch Eigenschaften vorgesehen, die die Größe einer Tabelle bestimmen. Eigentlich kommt man aber mit der Verwendung der Eigenschaft `width` (Tabellenbreite) aus, wobei wieder einmal betont wird, dass diese Angabe möglichst in %en erfolgen soll, um die Skalierbarkeit der Web-Seite zu gewährleisten.

Eigenschaften des `<table>` - Tags:

<code>border="4"</code>	... die Angabe der Breite des Randes um die gesamte Tabelle in Pixel
<code>cellspacing="1"</code>	... die Breite der Gitterlinien zwischen den Zellen in Pixel
<code>bgcolor="red"</code>	... die Hintergrundfarbe der Tabelle (<i>Farbwerte siehe Abschnitt 5</i>)
<code>bordercolor="#324F0A"</code>	... die Farbe des Tabellenrandes (<i>Farbwerte siehe Abschnitt 5</i>)
<code>width="70%"</code>	... die Breite der Tabelle relativ zur Breite des Browserfensters
<code>align="right"</code>	... Ausrichtung der Tabelle. Die möglichen Werte sind <code>"left"</code> , <code>"right"</code> und <code>"center"</code>

Einstellungen für die Linien einer Tabelle:

```
<table border="2" cellspacing="4">
  ...
</table>
```



Aufgabe 26: Tabellen mit unterschiedlichen Rändern



Einstellungen für die Farben einer Tabelle:

```
<table border="2" bordercolor="green" bgcolor="yellow">
  ...
</table>
```



Aufgabe 27: Tabelle mit Farbeinstellungen



Einstellungen für die Breite und Ausrichtung einer Tabelle:

```
<table border="1" width="80%" align="center">
  ...
</table>
```



Aufgabe 28: Ausgerichtete Tabellen unterschiedlicher Breite





HTML - PROGRAMMIERUNG

24. TABELLEN - ZELLENEIGENSCHAFTEN

Zusätzlich zu den Einstellungen, die für die gesamte Tabelle getroffen werden, besteht auch die Möglichkeit die Ausrichtung des Inhalts, wie auch die Hintergrundfarbe einer Zelle festzulegen. Diese geschieht natürlich im `<td>` - Tag.

Weiters ist es auch möglich, die Breite und Höhe von Zellen relativ (in %) oder absolut (in Pixel) einzustellen. Meist werden diese Einstellungen aber nicht benötigt, da das automatische Anpassen von Zellenhöhe und -breite durch die Browser sehr gute und zufriedenstellende Ergebnisse liefert.

Eigenschaften des `<td>` - Tags:

`bgcolor="red"` ... die Hintergrundfarbe der Tabelle (*Farbwerte siehe Abschnitt 5*)

`align="top"` ... horizontale Ausrichtung des Inhalts in einer Zelle, die möglichen Werte sind `"left"`, `"right"`, `"center"` und `"justify"`.

`valign="right"` ... vertikale Ausrichtung des Inhalts in einer Zelle, die möglichen Werte sind `"top"`, `"middle"` und `"bottom"`.

In der folgenden Tabelle würde das Wort „Hallo!“ in der Zelle links oben (deren Hintergrund rot ist) genau in der Mitte angezeigt werden:

```
<table border="1">
  <tr>
    <td bgcolor="red" valign="middle" align="center">Hallo!</td>
    ...
  </tr>
</table>
```



Aufgabe 29: Eine bunte Tabelle mit unterschiedlicher Ausrichtung in den Zellen





25. TABELLEN - VERBINDEN VON ZELLEN

Für das Verbinden von Zellen stehen im `<td>` - Tag die beiden Eigenschaften `colspan` und `rowspan` zur Verfügung. Damit lassen sich komplexe Tabellen mit unterschiedlichen Zellen wie in den Abbildungen erzeugen. Der Code für die beiden abgebildeten Tabellen erklärt das „Wirken“ dieser beiden Eigenschaften.

Die Zahl, die den beiden Eigenschaften zugewiesen wird, bezieht sich auf die Zeilen (rows) bzw. Spalten (Cols), über die sich die Zelle erstrecken soll. Natürlich muss dann darauf geachtet werden, dass in der nächsten Zeile bzw. Spalte nicht zu viele Zellen angelegt werden.

2 Beispiele für die Verwendung von verbundenen Zellen:

...		
...
...

...
...
...

```
...  
<table border="1">  
  <tr>  
    <td colspan="3">...</td>  
  </tr>  
  <tr>  
    <td>...</td>  
    <td>...</td>  
    <td>...</td>  
  </tr>  
  <tr>  
    <td>...</td>  
    <td>...</td>  
    <td>...</td>  
  </tr>  
</table>  
...
```



```
...  
<table border="1">  
  <tr>  
    <td rowspan="2">...</td>  
    <td>...</td>  
    <td>...</td>  
  </tr>  
  <tr>  
    <td>...</td>  
    <td>...</td>  
  </tr>  
  <tr>  
    <td>...</td>  
    <td>...</td>  
    <td>...</td>  
  </tr>  
</table>  
...
```

Hier ist zu beachten, dass in der ersten Zeile nur eine Zelle angelegt wird, die sich über drei Spalten erstreckt.

Dieses Beispiel ist schon etwas komplizierter. In der ersten Zeile werden drei Zellen angelegt, wobei sich die erste über zwei Zeilen erstreckt. Deshalb dürfen in der zweiten Zeile nur zwei Zellen angelegt werden. Die dritte Zeile ist nicht betroffen und es werden drei Zellen angelegt.

Aufgabe 30: Verbinden von Zellen





26. AUSRICHTUNG VON BELIEBIGEN OBJEKTEN

Da nun (fast) alle Objekte, die sich auf einer Web-Seite befinden können, ausführlich behandelt worden sind, soll an dieser Stellen noch der wichtigste Tag für das horizontale Ausrichten von Objekten auf einer Web-Seite behandelt werden.

Die meisten Objekte bieten die Eigenschaft `align` an, mit der die Ausrichtung festgelegt werden kann. Der `<p>` - Tag dient dazu, um einen Textabsatz zu definieren und auszurichten. Nun gibt es aber auch die Möglichkeit, die Ausrichtung unterschiedlicher Objekte festzulegen. Dies geschieht durch den `<div>` - Tag. Alle auszugebenden Objekte einer Web-Seite, die zwischen `<div>` und `</div>` stehen, werden so ausgerichtet, wie es im `<div>` - Tag festgelegt wurde.

Verwendung des `<div>` - Tags:

Der `<div>` - Tag besitzt ausschließlich die Eigenschaft `align`. Die möglichen Werte sind `"left"`, `"right"`, `"center"` und `"justify"`.

```
...
<body>
  ...
  <div align="right">
    ...
    
    <a href="http://www.sony.com">Link</a>
    ...
    <table border="1">
      <tr>
        <td>...</td>
        <td>...</td>
      </tr>
      <tr>
        <td>...</td>
        <td>...</td>
      </tr>
    </table>
    Text Text Text
  </div>
...
```



In diesem Beispiel werden also alle angeführten Elemente rechtsbündig ausgegeben - egal ob es sich um eine Grafik, einen Link, eine Tabelle oder normalen Text handelt.

Aufgabe 31: Formatierung mit dem `<div>` - Tag





HTML - PROGRAMMIERUNG

27. FRAMES - GRUNDSTRUKTUR

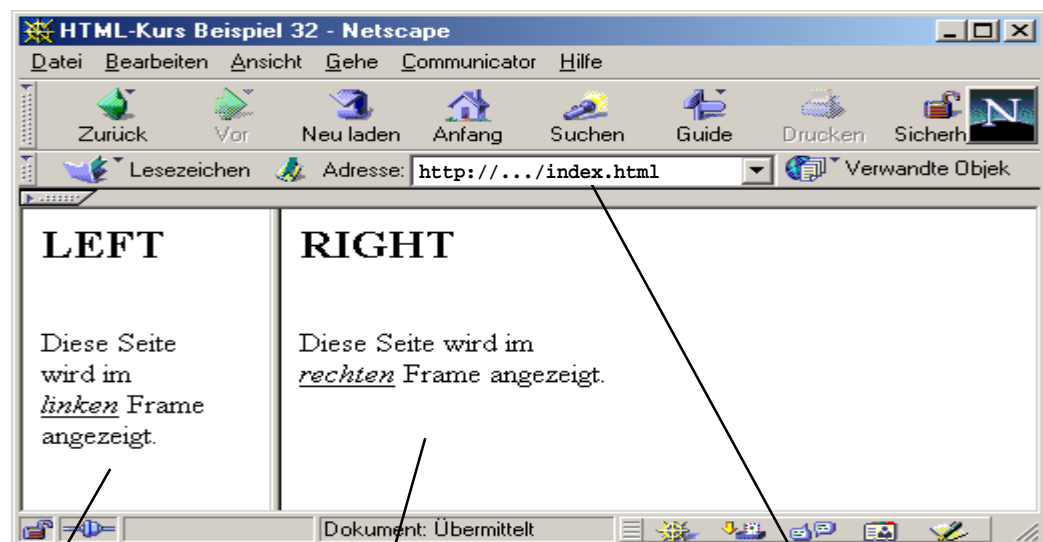
Mittlerweile ist die Darstellung mit Hilfe der Frame-Technik schon so selbstverständlich, dass diese für die Besucher von Web-Seiten eigentlich nichts besonderes mehr darstellt. Für den Ersteller von umfangreichen Web-Projekten ist die Verwendung von Frames fast unumgänglich, um eine gut strukturierte Präsentation seiner Inhalte anbieten zu können.

Was versteht man unter Frames?

Frames sind nichts anderes als **Rahmen**, die in HTML geschriebene Seiten darstellen können. Wird der Anzeigebereich einer Browsers in zwei Frames (also zwei Rahmen) unterteilt, so können in diesen unabhängig voneinander zwei Web-Seiten angezeigt werden.

Zur Unterteilung des Anzeigebereiches eines Browsers muss in der HTML-Datei, die vom Browser geladen wird, ein sogenanntes **Frameset** definiert sein. Diese Datei enthält nichts anderes als die Informationen zur Unterteilung des Anzeigebereiches des Browsers. Die „tatsächlichen“ Web-Seiten befinden sich wiederum in eigenen HTML-Dateien.

Welche HTML-Dateien werden benötigt und wie sehen diese aus?



```
<html>
<head>
...
</head>
<body>
...
LEFT
...
</body>
</html>
```

leftpage.html

```
<html>
<head>
...
</head>
<body>
...
RIGHT
...
</body>
</html>
```

mainpage.html

```
<html>
<head>
...
</head>
<frameset>
...
</frameset>
<noframes>
...
</noframes>
</html>
```

index.html

Für dieses Beispiel werden also **3 HTML-Dateien** benötigt:

die Steuerdatei **index.html** (enthält die Informationen zur Aufteilung in Frames)

die Web-Seite **leftpage.html** (wird im linken Frame angezeigt)

die Web-Seite **mainpage.html** (wird im rechten Frame angezeigt).



28. FRAMES - DIE FRAMESET-DEFINITION

Wie bereits erwähnt, muss die Datei, die von einem Browser geladen wird, die Informationen über die Unterteilung des Anzeigebereiches in Frames enthalten. Dies geschieht in der sogenannten Frameset-Definition.

Da mittlerweile (fast) alle Web-Server die Datei `index.html` als Startseite an einen Browser weiterleiten, wenn nur die DNS-Adressen mit dem jeweiligen Verzeichnis eingegeben wird, sind die Frameset-Definitionen in diesem Kurs immer in einer HTML-Datei mit diesem Namen abgelegt.

Die Struktur einer Frameset-Definition:

- Die Definition erfolgt durch die beiden Tags `<frameset>` und `</frameset>`.
- Dazwischen werden die einzelnen Frames mit dem Tag `<frame>` definiert.
- Die HTML-Datei mit der Frameset-Definition hat keinen BODY.
- Für ältere Browser (die keine Frames unterstützen und die es fast nicht mehr gibt) wird zwischen den Tags `<noframes>` und `</noframes>` eine entsprechende Meldung angegeben.

Die Steuerungsdatei mit der Frameset-Definition für das Beispiel auf der vorigen Seite:

```
<html>
<head>
  <title>Erste Frameset-Definition</title>
</head>
<frameset cols="25%,*">
  <frame name="leftframe" src="leftpage.html">
  <frame name="mainframe" src="mainpage.html">
</frameset>
<noframes>
  Sorry, Ihr Browser unterstützt leider keine Frames!
</noframes>
</html>
```



Die Eigenschaften des `<frameset>` - Tags:

Die Unterteilung in Frames kann entweder in Spalten oder Zeilen erfolgen. Dazu dienen die Eigenschaften `cols` und `rows`. Die Anzahl der Spalten bzw. Zeilen wird durch die Anzahl der zugewiesenen Zahlenwerte, die durch eine Beistrich getrennt sind, festgelegt. Die Angaben können absolut (in Pixel) oder relativ (in Prozent) erfolgen.

`cols="40%,*"` ... Unterteilung in zwei Spalten, der linke Frame hat eine Breite von 40% der Breite des Browserfensters. Die Breite des zweiten Frames wird automatisch ermittelt.

`rows="100,*,40"` ... Unterteilung in 3 Zeilen, die Höhen des oberen und unteren Frames sind fix in Pixel gegeben. Die Höhe des mittleren Frames wird automatisch ermittelt.

Die Eigenschaften des `<frame>` - Tags:

`name="leftframe"` ... hier wird der Name des Frames festgelegt, über den er später angesprochen werden kann.

`src="seite.html"` ... dies ist die HTML-Datei, die in diesem Frame angezeigt wird. Die Angabe kann dabei absolut oder relativ erfolgen (*siehe Abschnitte zu den Links*).

Aufgabe 32: Erstes Frameset





29. FRAMES - RÄNDER VON FRAMES, SCROLLING

Anzeigen eines Randes zwischen den Frames:

Wird bei der Frameset-Definition keine weitere Angabe gemacht, wird zwischen den Frames ein Rand angezeigt, den der Besucher verschieben und somit die Größe der Frames verändern kann. Will man aus gestalterischen Gründen, dass zwischen den Frames kein Rand angezeigt wird und somit ein fließender Übergang zwischen gleichen Hintergrundfarben oder Texturen zu sehen ist, so gibt es die Möglichkeit, das Anzeigen des Randes zu unterdrücken, indem man die Eigenschaft `frameborder="0"` setzt.

```
...
<frameset cols="25%,*" frameborder="0">
  <frame name="leftframe" src="leftpage.html">
  <frame name="mainframe" src="mainpage.html">
</frameset>
...
```



Aufgabe 33: Frameset ohne Rand



Mögliches Ändern der Framegröße unterdrücken:

Will man den Rand zwischen den Frames angezeigt haben, dem Besucher aber nicht ermöglichen, diesen Rand zu verschieben, so kann dies durch Setzen der Eigenschaft `noresize` bei der Definition eines Frames im `<frame>`-Tag geschehen. Im folgenden Beispiel genügt es, wenn dies beim linken Frame gemacht wird, wenn man z. B. eine Unterteilung in drei Frames hat, dann setzt man diese Eigenschaft bei der Definition des mittleren Frames.

```
...
<frameset cols="25%,*">
  <frame name="leftframe" src="leftpage.html" noresize>
  <frame name="mainframe" src="mainpage.html">
</frameset>
...
```



Aufgabe 34: Frameset mit Rahmen und fester Breite



Scrollbars in Frames:

Tätigt man bei der Festlegung der einzelnen Frames keine Einstellung, so entscheidet der Browser selbst, ob er gemäß des angezeigten Inhalts Scrollbars in einem Frame anzeigt oder nicht. Will man die Anzeige von Scrollbars unterdrücken, setzt man die Eigenschaft `scrolling="no"` im `<frame>`-Tag, will man die Ausgabe von Scrollbars erzwingen, so setzt man im `<frame>`-Tag die Eigenschaft `scrolling="yes"`.

Im folgenden Beispiel wird also die Anzeige von Scrollbars im linken Frame unterdrückt, egal, wie groß die darin dargestellte Seite ist. Der Besucher hat allerdings keine Möglichkeit, die verdeckten Inhalte zu sehen.

```
...
<frameset cols="25%,*">
  <frame name="leftframe" src="leftpage.html" scrolling="no">
  <frame name="mainframe" src="mainpage.html">
</frameset>
...
```





HTML - PROGRAMMIERUNG

30. FRAMES - LINKS IN FRAMESETS

Meist werden Framesets dazu verwendet, um Inhalte in einer übersichtlichen und strukturierten Weise zu präsentieren. Dies setzt natürlich voraus, dass die Frames miteinander interagieren können. Wenn man also auf einen Link im linken Frame klickt, soll z. B. im rechten Frame eine neue Seite angezeigt werden.

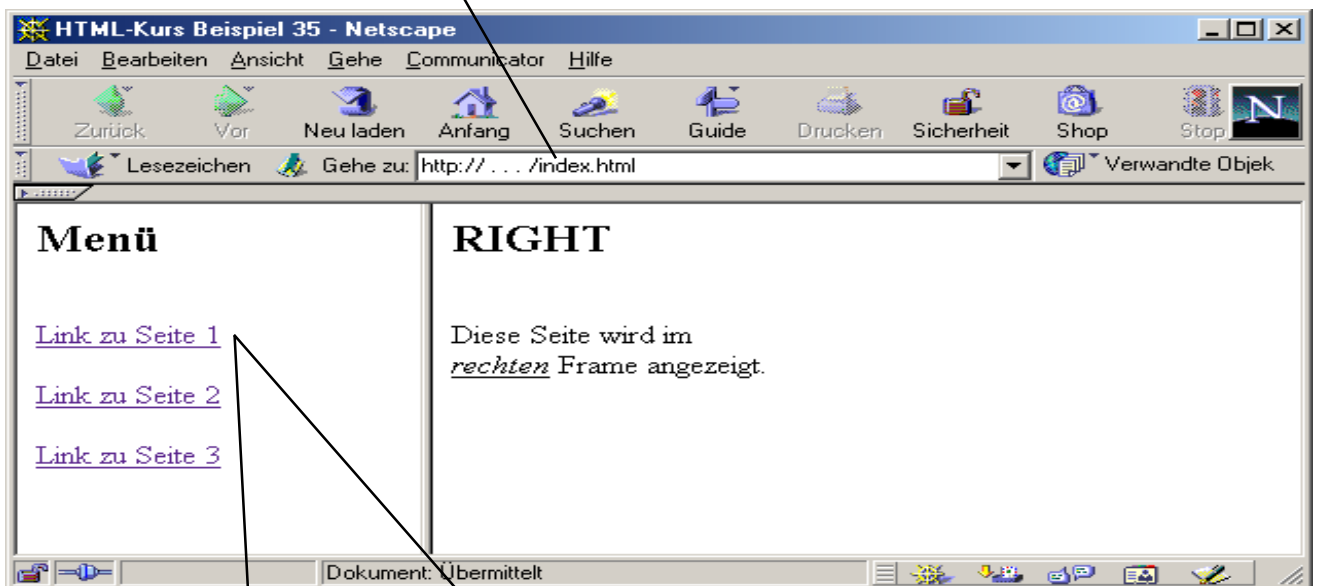
Um dies zu verstehen, ist bereits alles bekannt, was man dazu benötigt. In Abschnitt 20 wurde bereits die Eigenschaft `target` des Tags zur Erzeugung eines Links erklärt und dort verwendet, um Web-Seiten in einem neuen Browser-Fenster anzeigen zu lassen.

Will man nun die verlinkte Seite nicht in einem neuen Fenster, sondern in einem anderen Frame anzeigen lassen, so dient dazu wieder diese Eigenschaft `target` des `<a>`-Tags. Dieser wird einfach der Name des Frames zugewiesen, in dem die Seite angezeigt werden soll. Zur Erinnerung: Der Name des Frames wurde in der Frameset-Definition im `<frame>`-Tag festgelegt.

Ein Beispiel:

Im Folgenden ist der Teil der Frameset-Definition aus der Steuerungsdatei für das Frameset angegeben. Zu beachten sind die Namen, die für die Frames festgelegt werden.

```
...  
<frameset cols="25%,*" frameborder="0">  
  <frame name="leftframe" src="leftpage.html">  
  <frame name="mainframe" src="mainpage.html">  
</frameset>  
...
```



```
...  
<a href="seite1.html" target="mainframe">Link zu Seite 1</a>  
<br>  
<a href="seite2.html" target="mainframe">Link zu Seite 2</a>  
<br>  
<a href="seite3.html" target="mainframe">Link zu Seite 3</a>  
...
```

Nebenan ist ein Ausschnitt aus der HTML-Datei angegeben, die im linken Frame angezeigt wird. Man sieht deutlich, dass bei der Festlegung der Links der Anzeigebereich für die verlinkte Seite in der Eigenschaft `target` festgelegt wird.

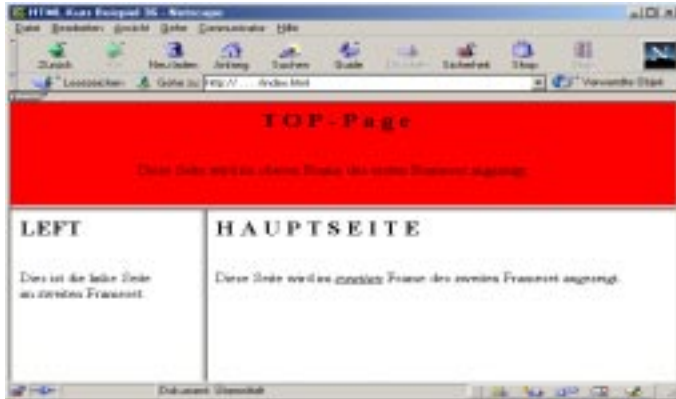
Aufgabe 34: Die Funktion von Links in Framesets





31. FRAMES - VERSCHACHTELN VON FRAMESETS

Bis jetzt wurde nur gezeigt, wie man den Anzeigebereich eines Browsers in Spalten oder Zeilen unterteilen möchte. Was ist aber zu tun, wenn man wie in der ersten Abbildung oben einen Frame für die Angabe des Firmennamens, der Anschrift und des Logos hätte, sich links ein Menü-Frame befinden soll und rechts in einem Frame die Web-Seiten mit den zu präsentierenden Inhalten angezeigt werden sollen?



Die Lösung dieser Aufgabe gestaltet sich im Prinzip recht einfach - nur: der Code für die Frameset-Definition wird zunehmend komplizierter.

Im Grunde macht man nichts anderes, als anstatt der Festlegung eines Frames eine weitere Frameset-Definition „einzubauen“.

D.h.: Für die Erzeugung der Unterteilung der neigen Abbildung wird zuerst eine Unterteilung in zwei Zeilen erzeugt. Weiters wird anstatt des zweiten Frames in diesem Frameset ein weiteres Frameset mit zwei Spalten definiert.

„Der Code sagt mehr als tausend Worte“:

Anstelle der Definition des zweiten Frames im ersten Frameset wird ein weiteres Frameset festgelegt.

Zusätzlich soll hier wieder einmal darauf hingewiesen werden, dass konsequentes Einrückchen beim Schreiben des HTML-Code sehr hilfreich ist und spätere Fehlersuche erleichtert bzw. überflüssig macht.

```
...
<frameset rows="80,*">
  <frame name="topframe" src="toppage.html">
  <frameset cols="100,*">
    <frame name="leftframe" src="leftpage.html">
    <frame name="mainframe" src="mainpage.html">
  </frameset>
</frameset>
...
```

Aufgabe 36: Verschachtelte Framesets 1



Weitere Möglichkeiten:

Der Fantasie sind beim Verschachteln von Framesets keine Grenzen gesetzt - vorausgesetzt, man schafft es noch, im HTML-Code den Überblick zu behalten. Natürlich kann man in der Definition des zweiten Framesets wiederum ein weiteres Frameset einbauen.

Die wichtigste Vorgabe bei der Erstellung ausgefeilter Framesets sollte aber immer die Übersichtlichkeit für leichtere Navigation für einen Besucher bleiben. Sollte diese durch zu starken programmierischen Einsatz und Enthusiasmus verfehlt werden, so hat man sicher über das Ziel hinausgeschossen.

Aufgabe 37: Verschachtelte Framesets 2



Aufgabe 38: Verschachtelte Framesets 3

