



Werkzeuge der Informatik

Übung 1

Aufgabe 1: HTML Grundlagen*

Aufgabe 2: Einführung in Cascading Stylesheets (CSS)*

Wintersemester 2009/ 2010

Institut für Informatik

*) <http://de.selfhtml.org/>

Übersicht Aufgabe 1: HTML Grundlagen

▪ **Aufgabenstellung**

- 0: Einführung in HTML
- 1: Allgemeine Regeln für HTML
- 2: Dateiweite Einstellungen
- 3: Elemente für die Textstrukturierung
- 4: Tabellen
- 5: Verweise
- 6: Grafiken
- 7: Formulare

Aufgabenstellung: HTML Feedback-Formular

- In dieser Aufgabe erstellen Sie ein Feedback-Formular in HTML, das folgendes Aussehen hat (Die Daten in den Eingabefeldern dienen lediglich als Beispiel):

Feedbackformular - Windows Internet Explorer

http://wirf.in.tu-clausthal.de/foalem/ws0809/feedback.html

Vielen Dank, dass Sie uns ein Feedback senden wollen!

Vorname	Olivier	
Name	Foalem	
Strasse	Julius-Albert-Str. 4	
PLZ und Ort	38678	Clausthal-Zellerfeld
E-Mail	olivier.foalem@tu-clausthal.de	
Ihre Nachricht an uns	<p>Hallo!</p> <p>Dies ist eine Testnachricht.</p> <p>Viele Grüße, Olivier Foalem</p>	
Bitte beurteilen Sie den aktuellen Aufgabenzettel	<p><input type="radio"/> Ich finde den Aufgabenzettel zu schwer</p> <p><input type="radio"/> Ich finde den Aufgabenzettel angemessen</p> <p><input type="radio"/> Mir sind die Aufgaben zu leicht</p>	

Senden

- Verwenden Sie bitte KEINEN grafischen HTML-Editor! Diese Abgaben werden NICHT akzeptiert.**

Aufgabenstellung: Details (1)

- **Arbeiten Sie zunächst dieses Skript durch, um die grundlegenden Prinzipien von HTML zu erlernen (S. 8 ff.). Mit Hilfe dieses Skripts erhalten Sie einen Überblick über alle notwendigen HTML-Schlüsselwörter, die zur Bearbeitung des Aufgabenzettels notwendig sind!**
- **Betrachten Sie erst anschließend das „HTML-Feedbackformular“ genauer:**
 - Es besteht aus einer zentrierten Überschrift und einem (grau hinterlegten) Formularbereich.
 - Das Formular ist als Tabelle bestehend aus acht Zeilen aufgebaut:
 - Die ersten drei Zeilen enthalten zwei Spalten
 - Die vierte Zeile enthält drei Spalten
 - Die Zeilen fünf bis sieben bestehen wieder aus zwei Spalten und
 - die letzte Zeile enthält nur eine Spalte!
 - Das Formular enthält folgende Elemente:
 - Die Textfelder „vorname“, „nachname“, „strasse“, „plz“, „ort“ und „email“
 - Den Textbereich „nachricht“
 - Drei Optionsschalter (Radio-Buttons) mit Namen „auswahl“
 - Den Button „schalter“

Aufgabenstellung: Details (2)

- Das Formular selbst trägt folgende Attribute (bitte auf korrekte Schreibweise achten!):
 - name="feedback"
 - method="post"
 - action="http://winf.in.tu-clausthal.de/fofoalem/ws0910/wdi-skript.php"
- Für die Anzeigelänge und die interne Feldlänge der Textfelder definieren Sie bitte jeweils einheitlich:
 - „vorname“: 40
 - „nachname“: 40
 - „strasse“: 60
 - „plz“: 20
 - „ort“: 50
 - „email“: 40
- Für den Textbereich gilt: Spalten = 80 und Zeilen = 15.
- Für die drei Optionsschalter definieren Sie bitte folgende Werte:
 - 1. value = „schwer“
 - 2. value = „angemessen“
 - 3. value = „leicht“
- Der Button „Senden“ ist vom Typ „submit“.

Aufgabenstellung: Ergebnis

- Wenn Sie denken, dass Sie alle Eingaben richtig gemacht haben, dann sollten Sie nach dem Ausfüllen aller Eingabefelder und Drücken des Buttons „Absenden“ eine HTML-Seite „Auswertung Ihrer Eingaben“ sehen, in dem alle Eingaben zusammengefaßt auftauchen, die Sie zuvor gemacht haben:

Auswertung Ihrer Eingaben - Windows Internet Explorer

http://winf.in.tu-clausthal.de/foalem/ws0809/wdi-skript.php

Datei Bearbeiten Ansicht Favoriten Extras ?

Auswertung Ihrer Eingaben

Vielen Dank für Ihr Feedback!

Sie haben folgenden Eingaben gemacht:

Ihr Name: Olivier Foalem
Ihre Anschrift: Julius-Albert-Str. 4 in 38678 Clausthal-Zellerfeld

Sie haben uns folgenden Text hinterlassen:
Hallo! Dies ist eine Testnachricht. Viele Grüße, Olivier Foalem

Sie finden dieser Aufgabenzettel hat den Schwierigkeitsgrad: angemessen

Diese Angaben wurden an uns per E-Mail versendet.
Sie haben eine Kopie dieser E-Mail an folgende E-Mail-Adresse erhalten: olivier.foalem@tu-clausthal.de

Hier dürfen keine Eingaben fehlen!

Fehleranalyse

- Das Serverskript (wdi-skript.php), welches Sie aus Ihrem Formular aufrufen, erwartet genau die verlangten Attributnamen, um diese auszuwerten!
- Das Skript wird Ihnen
 - das Bestätigungsfenster erstellen
- Sollte das Bestätigungsfenster trotz korrektem Ausfüllen aller Eingabefelder nicht alle Daten enthalten (so wie Sie es eigentlich erwartet haben), so überprüfen Sie bitte in jedem Fall, ob Sie in Ihrem HTML-Dokument die korrekten Attributnamen für die Formularelemente verwendet haben.

Übersicht: HTML Grundlagen

- Aufgabenstellung
- **0: Einführung in HTML**
- 1: Allgemeine Regeln für HTML
- 2: Dateiweite Einstellungen
- 3: Elemente für die Textstrukturierung
- 4: Tabellen
- 5: Verweise
- 6: Grafiken
- 7: Formulare

0.1 HTML als „lingua franca“ des Web

- Unter einer "lingua franca" versteht man eine "Brot- und Buttersprache", eine Sprache, die jeder kennt, jeder spricht, jeder leicht erlernen kann und jeder braucht. HTML ist eine solche Sprache. Vom Web-Gründer Tim Berners-Lee entwickelt, wurde HTML im Zuge des Web-Booms zum erfolgreichsten und verbreitetsten Dateiformat der Welt. Immer wieder rümpfen Entwickler, die gerne alles, was einfach sein könnte, durch tausend Filter und Übersetzer jagen, um es kompliziert zu machen und ihren technischen Durst zu stillen, über HTML die Nase - eben weil es so einfach ist.
- **Aber erstens** ist es bei genauerem Hinsehen gar nicht so einfach - fast kein namhaftes Web-Angebot - egal ob Suchmaschine, Online-Magazin oder Firmenauftritt - ist bis heute in der Lage, fehlerfreies und standardkonformes HTML auf seinen Web-Seiten zu realisieren.
- **Und zweitens** reicht HTML für die Mehrzahl der Inhalte, die heute im Web angeboten werden, vollkommen aus. Denn HTML ist eine Sprache zur Strukturierung von Texten, wobei aber auch die Möglichkeit besteht, Grafiken und multimediale Inhalte in Form einer Referenz einzubinden und in den Text zu integrieren.
- **Mit HTML können Sie Überschriften, Textabsätze, Listen und Tabellen erzeugen. Sie können anklickbare Verweise auf beliebige andere Web-Seiten oder Datenquellen im Internet erzeugen. Nicht-textuelle Inhalte können Sie wie bereits erwähnt referenzieren. Sie können Formulare in den Text integrieren. Und last but not least bietet HTML Schnittstellen für Erweiterungssprachen wie Stylesheets oder JavaScript an, mit deren Hilfe Sie HTML-Elemente nach Wunsch gestalten und formatieren oder Interaktion mit dem Anwender realisieren können.**

0.2 HTML als software-unabhängiges Klartextformat

- HTML ist ein so genanntes Klartextformat. HTML-Dateien können Sie mit jedem beliebigen Texteditor bearbeiten, der Daten als reine Textdateien abspeichern kann. Es gibt also keine bestimmte Software, die man zum Erstellen von HTML-Dateien benötigt. Zwar gibt es längst mächtige Programme, die auf das Editieren von HTML spezialisiert sind, doch das ändert nichts an der entscheidenden Eigenschaft: **HTML ist nicht an irgendein bestimmtes, kommerzielles Software-Produkt gebunden.** Diese wichtige, vielleicht sogar wichtigste Eigenschaft von HTML sollten Sie immer im Auge behalten, wenn man Ihnen erzählen will, dass Web Publishing nur mit bestimmten Software-Produkten möglich sei.
- Da HTML ein Klartextformat ist, lässt es sich auch hervorragend mit Hilfe von Programmen generieren. Von dieser Möglichkeit machen beispielsweise CGI-Scripts Gebrauch. Wenn Sie im Web zum Beispiel einen Suchdienst benutzen und nach einer Suchanfrage die Ergebnisse präsentiert bekommen, dann ist das, was Sie am Bildschirm sehen, HTML-Code, der von einem Script generiert wurde.

0.3 HTML als Auszeichnungssprache

- HTML bedeutet **HyperText Markup Language**. **HTML ist eine so genannte Auszeichnungssprache (Markup Language). Sie hat die Aufgabe, die logischen Bestandteile eines textorientierten Dokuments zu beschreiben.** Als Auszeichnungssprache bietet HTML daher die Möglichkeit an, typische Elemente eines textorientierten Dokuments, wie Überschriften, Textabsätze, Listen, Tabellen oder Grafikreferenzen, als solche auszuzeichnen.
- Das Auszeichnungsschema von HTML geht von einer hierarchischen Gliederung aus. HTML zeichnet Inhalte von Dokumenten aus. Dokumente haben globale Eigenschaften wie zum Beispiel Kopfdaten. Der eigentliche Inhalt besteht aus Elementen, zum Beispiel einer Überschrift 1. Ordnung, Textabsätzen, Tabellen und Grafiken. Einige dieser Elemente haben wiederum Unterelemente. So enthält ein Textabsatz zum Beispiel eine als betont oder fett markierte Textstelle, eine Aufzählungsliste besteht aus einzelnen Listenpunkten, und eine Tabelle gliedert sich in einzelne Tabellenzellen.
- **Die meisten dieser Elemente haben einen fest definierbaren Erstreckungsraum.** So geht eine Überschrift vom ersten bis zum letzten Zeichen, eine Aufzählungsliste vom ersten bis zum letzten Listenpunkt, oder eine Tabelle von der ersten bis zur letzten Zelle. Auszeichnungen markieren Anfang und Ende von Elementen. Um etwa eine Überschrift auszuzeichnen, lautet das Schema:
 - **[Überschrift] Text der Überschrift [Ende Überschrift]**
- Bei einem Element, das wiederum Unterelemente besitzt, etwa einer Aufzählungsliste, lässt sich das gleiche Schema anwenden:
 - **[Liste]**
[Listenpunkt] Text des Listenpunkts [Ende Listenpunkt]
[Listenpunkt] Text des Listenpunkts [Ende Listenpunkt]
[Ende Liste]
- Web-Browser, die HTML-Dateien am Bildschirm anzeigen, lösen die Auszeichnungsmarkierungen auf und stellen die Elemente dann in optisch gut erkennbarer Form am Bildschirm dar.

Übersicht: HTML Grundlagen

- Aufgabenstellung
- 0: Einführung in HTML
- **1: Allgemeine Regeln für HTML**
- 2: Dateiweite Einstellungen
- 3: Elemente für die Textstrukturierung
- 4: Tabellen
- 5: Verweise
- 6: Grafiken
- 7: Formulare

1.1 Elemente und Tags in HTML

- HTML-Dateien bestehen aus Text. Zur Textauszeichnung gibt es bestimmte Zeichen aus dem normalen Zeichenvorrat.
- Der Inhalt von HTML-Dateien steht in **HTML-Elementen**. HTML-Elemente werden durch so genannte **Tags** markiert. Fast alle HTML-Elemente werden durch ein einleitendes und ein abschließendes Tag markiert. Der Inhalt dazwischen ist der "Gültigkeitsbereich" des entsprechenden Elements. Tags werden in spitzen Klammern notiert.
- Beispiel: `<h1>HTML - die Sprache des Web</h1>`
 - Das Beispiel zeigt eine Überschrift 1. Ordnung.
 - Das einleitende Tag `<h1>` signalisiert, dass eine Überschrift 1. Ordnung folgt (*h = heading = Überschrift*).
 - Das abschließende Tag `</h1>` signalisiert das Ende der Überschrift. Ein abschließendes Tag beginnt mit einer öffnenden spitzen Klammer und einem Schrägstrich `</`.

1.2 Verschachtelung von Elementen

- Elemente können ineinander verschachtelt werden.
- Auf diese Weise entsteht eine hierarchische Struktur. Komplexere HTML-Dateien enthalten sehr viele Verschachtelungen. Deshalb sprechen Fachleute auch von **strukturiertem Markup**.
- Beispiel: `<h1><i>HTML</i> - die Sprache des Web</h1>`
 - Das `i`-Element steht für *italic* (= *kursiver Text*). Der Text zwischen `<i>` und `</i>` wird als kursiv interpretiert, abhängig von der eingestellten Schriftart und Schriftgröße für die Überschrift 1. Ordnung.

1.3 Attribute in Tags

- Einleitende Tags und Standalone-Tags können zusätzliche Angaben enthalten.
- Beispiel: `<h1 align="center">HTML - die Sprache des Web</h1>`
 - Durch `align="center"` wird bewirkt, dass der Text zentriert ausgerichtet wird (*align = Ausrichtung, center = zentriert*).

1.4 Grundgerüst einer HTML-Datei

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>Beschreibung der Seite</title>
  </head>
  <body>
  </body>
</html>
```

1.4.1 Dokumenttyp-Deklaration

- HTML ist innerhalb der Familie der Auszeichnungssprachen nur eine von vielen, wenn auch die prominenteste. HTML selbst hat außerdem bereits eine mehrjährige Geschichte und ist in verschiedenen, recht unterschiedlichen Versionen normiert worden. Mit der Dokumenttyp-Deklaration bestimmen Sie, welche Auszeichnungssprache in welcher Version Sie verwenden. Eine auslesende Software, etwa ein Web-Browser, kann sich an dieser Angabe orientieren.
- Die Regeln für HTML sind mit Hilfe von SGML formuliert. Nach den Regeln einer SGML-basierten Auszeichnungssprache ist eine HTML-Datei erst dann eine **gültige (valide)** Datei, wenn sie einen bestimmten Dokumenttyp angibt und sich dann innerhalb des restlichen Quelltextes genau an die Regeln hält, die für diesen Dokumenttyp definiert sind.
- Denn hinter jeder Dokumenttyp-Deklaration stecken so genannte **Dokumenttyp-Definitionen (DTD)**. Auch für HTML gibt es solche Dokumenttyp-Definitionen. Dort ist geregelt, welche Elemente ein Dokument vom Typ HTML enthalten darf, welche Elemente innerhalb von welchen anderen vorkommen dürfen, welche Attribute zu einem Element gehören, ob die Angabe dieser Attribute Pflicht ist oder freiwillig usw.
- **Als HTML-Anfänger mag Ihnen der Aufwand, der da betrieben wird, vielleicht nicht ganz einleuchten. Doch genau diese Dokumenttypen, mit deren Hilfe sich Regeln für Sprachen wie HTML genau definieren lassen, sind ein riesiger Fortschritt in der EDV. Denn nur so lässt sich das Konzept der software-unabhängigen, aber regelgerechten Dateiformate konsequent durchsetzen.**
- Ohne den Bezug auf die offiziellen Regeln wären Sprachen wie HTML unverbindliche Konventionen, die schnell in Dialekten verwässern würden.
- Das ist genauso wie bei natürlichen Sprachen: ohne eine gewisse Regelung der "Rechtschreibung" würde eine Schriftsprache im Laufe der Zeit so stark zersetzt, dass am Ende kaum jemand mehr verstehen kann, was der andere mit dem, was er geschrieben hat, meint.
- Da Software außerdem noch viel dümmer ist, als Menschen, und viel genauere Vorgaben benötigt, um zu "verstehen", was man ihr mitteilt, ist das Beziehen auf Regeln dort sogar noch wesentlich wichtiger.

1.4.1 Dokumenttyp-Deklaration (2)

- Beispiel einer Dokumenttyp-Deklaration:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/loose.dtd">
```

- Notieren Sie die Dokumenttyp-Deklaration am Anfang der HTML-Datei vor dem einleitenden <html>-Tag in Großbuchstaben so, wie angegeben.
- Hinter der startenden spitzen Klammer folgt ein Ausrufezeichen. Dahinter folgt die Angabe DOCTYPE HTML PUBLIC. Das bedeutet, dass Sie sich auf die öffentlich verfügbare HTML-DTD beziehen. Die folgende Angabe, die in Anführungszeichen steht, ist wie folgt zu verstehen:
 - **W3C** ist der Herausgeber der DTD, also das W3-Konsortium.
 - Eine Angabe wie **DTD HTML 4.01 Transitional** bedeutet, dass Sie in der Datei den Dokumenttyp "HTML" verwenden, und zwar in der Sprachversion 4.01 und deren Variante Transitional.
 - Das **EN** ist ein Sprachenkürzel und steht für die Sprache, in diesem Fall Englisch. Die Angabe bezieht sich darauf, in welcher natürlichen Sprache die Element- und Attributnamen der Tagsprache definiert wurden, nicht auf den Inhalt Ihrer Datei. Benutzen Sie also immer EN, da die Namen von HTML-Elementen und -Attributen auf der englischen Sprache basieren.
 - Die Schrägstriche notieren Sie so wie im obigen Beispiel.
 - Ferner enthält die Dokumenttyp-Deklaration die **Web-Adresse der Dokumenttyp-Definition** (DTD). Die Angabe dieser Adresse ist nicht zwingend nötig. Über die angegebene Web-Adresse kann eine auslesende Software die Dokumenttyp-Definitionen aufrufen und in den darin notierten Regeln "nachgucken". Die meisten heutigen Browser tun das bei HTML allerdings nicht, weil die verbreiteten Dokumenttypen bereits implementiert sind.

1.5 Regeln bei der Texteingabe

- Wenn Sie einen Texteditor oder einen quelltextorientierten HTML-Editor zur Erstellung Ihrer HTML-Dateien verwenden, sollten Sie folgende Regeln kennen und beachten:
 - Notieren Sie in einer neuen Datei immer zuerst das Grundgerüst einer HTML-Datei (vgl. Abschnitt 1.4).
 - Beachten Sie bei der Texteingabe die Maskierungsvorschriften für Zeichenvorrat, Sonderzeichen und HTML-eigene Zeichen.
 - Setzen Sie Zeilenumbrüche und Leerzeilen so, dass Sie im Quelltext eine optimale Übersicht behalten. Beachten Sie aber auch, dass Zeilenumbrüche und Absatzschaltungen im Web-Browser nicht so angezeigt werden, wie sie im Quelltext eingegeben wurden.
 - Für Zeilenumbrüche und Absatzschaltungen, die im WWW-Browser wirksam sein sollen, müssen Sie die entsprechenden HTML-Elemente verwenden, zum Beispiel diejenigen für Textabsätze (Absatzschaltungen) oder Zeilenumbrüche.
- Beachten Sie, dass es in HTML keine Tabulatoren gibt. Ein im Quelltext eingegebener Tabulator wird bei der Anzeige im Web-Browser in ein Leerzeichen umgewandelt.
- Zeilenumbruch-Zeichen, Tabulator-Zeichen und einfache Leerzeichen bilden in HTML die Klasse der so genannten **White-Space-Zeichen** (*white space* = "weißer Raum"). Die Browser setzen in der Regel ein Tabulator-Zeichen oder Zeilenumbruch-Zeichen im Editor als Leerzeichen im HTML-Text um. Mehrere solcher White-Space-Zeichen hintereinander werden ignoriert und zu einem einzigen Leerzeichen zusammengefasst.
- Um mehrere Leerzeichen hintereinander zu erzwingen, können Sie anstelle der normalen Leerzeicheneingabe die Zeichenfolge „ “ (geschütztes Leerzeichen) eingeben, und zwar so oft hintereinander wie gewünscht.

1.6 HTML-Kopfdaten

- Jede HTML-Datei **muss** einen Titel erhalten. Das ist aus folgenden Gründen besonders wichtig:
 - Der Titel der Datei wird bei der Anzeige im Web-Browser in der Titelzeile des Anzeigefensters angezeigt.
 - Der Titel der Datei wird bei der Anzeige im Web-Browser in Karteireitern (tabs) angezeigt.
 - Der Titel der Datei wird vom Web-Browser beim Setzen von Lesezeichen (Bookmarks, Favoriten) auf die Datei verwendet.
 - Der Titel der Datei wird im Web-Browser in der Liste der bereits besuchten Seiten angezeigt.
 - Der Titel der Datei dient im Web vielen automatischen Suchprogrammen als wichtiger Input. Wenn die Datei zu den Suchtreffern einer Suche gehört, bieten viele Suchmaschinen den Titel der Datei als anklickbaren Verweis an.

```
<head>
    <title>Ausblick vom Hamburger Michel</title>
    <!-- ... andere Angaben im Dateikopf ... -->
</head>
```

- Innerhalb des Grundgerüsts einer HTML-Datei wird der Titel im Dateikopf notiert.
- `<title>` leitet die Angabe des Titels ein. Dahinter folgt der Text des Titels. Mit `</title>` wird die Titelangabe abgeschlossen (*title = Titel*).

1.7 Meta-Angaben zum Inhalt

- In Meta-Angaben können Sie verschiedene nützliche Anweisungen für Web-Server, Web-Browser und automatische Suchprogramme im Internet ("Robots") notieren.
- Meta-Angaben können Angaben zum Autor und zum Inhalt der Datei enthalten.
- Sie können aber auch HTTP-Befehle absetzen, zum Beispiel zum automatischen Weiterleiten des Web-Browsers zu einer anderen Adresse.
- **Die Meta-Angaben: Beschreibung, Autor, Stichwörter**
 - Die hier beschriebenen Meta-Angaben sind am verbreitetsten, sie wurden einst von fast allen großen Suchmaschinen im Web ausgelesen.
 - Die heutigen populären Suchmaschinen berücksichtigen sie teilweise immer noch. Insbesondere für seiteninterne Suchprogramme, die sich auf die Korrektheit der Metadaten verlassen können, sind die Angaben nach wie vor wichtig.
 - Sie können Ihre Autorenschaft, eine Kurzbeschreibung des Inhalts, charakteristische Stichwörter und das Publikationsdatum notieren. Insbesondere die Kurzbeschreibung wird in manchen Suchmaschinen angezeigt, wenn ein Anwender nach etwas sucht und diese Datei zu den Treffern gehört. Es ist also sinnvoll, diese Meta-Angaben in allen inhaltlich relevanten HTML-Dateien zu notieren und dabei redaktionell genauso sorgfältig zu sein wie bei dem Text, der im Browser-Fenster angezeigt wird.

```
<head>
  <meta name="description" content="Homepage von Olivier Foalem">
  <meta name="author" content="Olivier Foalem">
  <meta name="keywords" content="„ Olivier, Foalem, Clausthal-Zellerfeld">

  <!-- ... andere Angaben im Dateikopf ... -->
</head>
```

1.7 Meta-Angaben zum Inhalt (2)

- Mit `<meta name="description" content="Beschreibungstext" >` bestimmen Sie einen Beschreibungstext (*meta = "über", name = Name, description = Beschreibung, content = Inhalt*).
- Mit `<meta name="author" content="Autorenname" >` teilen Sie den Namen des Autors (d.h.: des inhaltlich Verantwortlichen) für die HTML-Datei mit (*author = Autor*).
- Mit `<meta name="keywords" content="[Wortliste]" >` bestimmen Sie Stichwörter für ein Suchprogramm. Ein Anwender, der in der Suchdatenbank des Suchprogramms nach einem dieser Stichwörter sucht, soll die aktuelle HTML-Datei bevorzugt als Suchtreffer angezeigt bekommen, da das Stichwort in dieser Datei ein zentrales Thema ist (*keywords = Schlüsselwörter*). Trennen Sie die einzelnen Schlüsselwörter durch Kommata. Ein Schlüsselwort kann durchaus aus zwei oder mehreren Einzelwörtern bestehen. Halbe oder ganze Sätze gehören hier allerdings nicht hin, denn dafür gibt es den Beschreibungstext, den Sie bei `name="description"` notieren können.

1.8 Meta-Angaben zur Zeichenkodierung

- Sie können mit Hilfe einer Meta-Angabe angeben, welche Zeichenkodierung die HTML-Datei verwendet.
- Diese Angabe ist für den Web-Browser besonders wichtig, denn sie teilt ihm mit, nach welchem Kodierung die Bytes der Datei in Zeichen umgewandelt werden müssen.

```
<head>
<meta http-equiv="content-type" content="text/html; charset=ISO-8859-1">
<!-- ... andere Angaben im Dateikopf ... -->
</head>
```

- Die Angabe der verwendeten Zeichenkodierung wird mit **http-equiv="content-type"** definiert. Dies weist darauf hin, dass eine Angabe gemacht wird, die normalerweise vom Web-Server in dem HTTP-Header Content-Type an den Web-Browser gesendet wird.
- Bei **content** geben Sie zunächst den MIME-Typ an, der für HTML-Dateien immer **text/html** lautet.
- Darauf folgt, durch einen Strichpunkt getrennt, die Angabe der Kodierung nach dem Schema **charset** zusammen mit dem eindeutigen Namen der Kodierung.
- Im obigen Beispiel wird die Kodierung **ISO-8859-1** angegeben, die für westeuropäische Sprachen, darunter Deutsch genutzt wird.

Übersicht: HTML Grundlagen

- Aufgabenstellung
- 0: Einführung in HTML
- 1: Allgemeine Regeln für HTML
- **2: Dateiweite Einstellungen**
- 3: Elemente für die Textstrukturierung
- 4: Tabellen
- 5: Verweise
- 6: Grafiken
- 7: Formulare

2.1 Farbe für den Hintergrund

- Sie können eine Farbe für den Hintergrund des Anzeigefensters bestimmen. Die gesamte HTML-Datei wird auf dieser Hintergrundfarbe angezeigt.

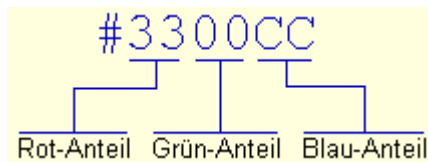
```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
  "http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <title>Dateiweite Farbe für den Hintergrund</title>
  </head>

  <body bgcolor="#CCFFFF">
    <h1>Dies ist eine Überschrift</h1>
  </body>
</html>
```

- Die Angabe zur Hintergrundfarbe erfolgt im einleitenden `<body>`-Tag der HTML-Datei.
- Mit dem Attribut `bgcolor` bestimmen Sie die Farbe für den Bildschirmhintergrund (*bgcolor = background color = Hintergrundfarbe*).
- Die gesamte HTML-Datei wird auf der hier definierten Hintergrundfarbe angezeigt.

2.1.1 Einschub: Farben in HTML

- Grundsätzlich gibt es zwei Möglichkeiten, Farben in HTML zu definieren:
 - durch Angabe der RGB-Werte der gewünschten Farbe in Hexadezimalform (RGB = Rot/Grün/Blau-Wert der Farbe)
 - durch Angabe eines Farbnamens
- Wenn Sie hexadezimale Werte angeben, arbeiten Sie Browser-unabhängig, und Sie haben die volle Freiheit zwischen 16,7 Millionen Farben.
- Eine hexadezimale Ziffer kann also 16 Zustände haben. Für jeden Farbwert (Rot, Grün, Blau) stehen 2 Ziffern zur Verfügung. Das macht $16 \times 16 (= 256)$ mögliche Zustände pro Farbwert.
- **Beachten Sie:** Es gibt 16 Grundfarben, die von jedem VGA-kompatiblen Bildschirm angezeigt werden können:



black	#000000	gray	#808080
maroon	#800000	red	#FF0000
green	#008000	lime	#00FF00
olive	#808000	yellow	#FFFF00
navy	#000080	blue	#0000FF
purple	#800080	fuchsia	#FF00FF
teal	#008080	aqua	#00FFFF
silver	#C0C0C0	white	#FFFFFF

2.2 Farben für Text und Verweise

- Sie können dateiweite Farben definieren:
 - für Text (gültig für alle Elemente wie Überschriften, normalen Fließtext, Listen usw.),
 - für Verweise zu noch nicht besuchten Stellen,
 - für Verweise zu bereits besuchten Stellen,
 - für Verweise, während sie angeklickt werden.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>Farbe f&uuml;r Text und Verweise</title>
</head>
<body bgcolor="#663333" text="#FFCC99" link="#FF9966" vlink="#FF9900" alink="#FFFFFF">
  <h1>Text</h1>
  <a href="http://de.yahoo.com/">Verweis zu Yahoo</a>
</body>
</html>
```

- Die Angaben erfolgen im einleitenden `<body>`-Tag der HTML-Datei. Für die Farbauswahl gelten die Regeln zum Definieren von Farben in HTML.
 - Mit **text** definieren Sie eine Farbe für den Text.
 - Mit **link** definieren Sie eine Farbe für Verweise zu noch nicht besuchten Dateien (*link = Verweis*). Mit **vlink** definieren Sie eine Farbe für Verweise zu bereits besuchten Dateien (*vlink = visited link = besuchter Verweis*).
 - Mit **alink** definieren Sie eine Farbe für Verweise, die der Anwender gerade anklickt (*alink = activated link = aktivierter Verweis*).

2.3 Hintergrundbild einbinden

- Sie können für die Anzeige einer HTML-Datei ein Hintergrundbild bestimmen. Dabei wird das Bild über das ganze Anzeigefenster hinweg immer wiederholt, so dass ein Tapeteneffekt (Wallpaper) entsteht.
- Besonders geeignet für Wallpaper-Effekte sind relativ kleine Grafiken, die irgendein abstraktes Muster darstellen.
- Die Hintergrundgrafik sollte als Grafikdatei im GIF-Format oder JPEG-Format vorliegen.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/loose.dtd">  
<html>  
<head>  
<title>Hintergrundbild einbinden</title>  
</head>  
  
<body background="background.jpg" text="#990000"> <h1>Text</h1>  
</body>  
</html>
```



Übersicht: HTML Grundlagen

- Aufgabenstellung
- 0: Einführung in HTML
- 1: Allgemeine Regeln für HTML
- 2: Dateiweite Einstellungen
- **3: Elemente für die Textstrukturierung**
- 4: Tabellen
- 5: Verweise
- 6: Grafiken
- 7: Formulare

3.1 Überschriften definieren

- HTML unterscheidet 6 Überschriftenebenen, um Hierarchieverhältnisse in Dokumenten abzubilden.
- `<h[1-6]>` (*h = heading = Überschrift*) leitet eine Überschrift ein.
 - Die Nummer steht für die Überschriftenebene.
 - 1 ist die höchste Ebene, 6 die niedrigste.
 - Dahinter folgt der Text der Überschrift.
 - `</h[1-6]>` beendet die Überschrift und steht am Ende des Überschriftentextes.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
    <title>&Uuml;berschriften definieren</title>
  </head>
  <body>
    <h1>&Uuml;berschrift 1. Ordnung</h1>
    <h2>&Uuml;berschrift 2. Ordnung</h2>
    <h3>&Uuml;berschrift 3. Ordnung</h3>
    <h4>&Uuml;berschrift 4. Ordnung</h4>
    <h5>&Uuml;berschrift 5. Ordnung</h5>
    <h6>&Uuml;berschrift 6. Ordnung</h6>
  </body>
</html>
```

Überschrift 1. Ordnung

Überschrift 2. Ordnung

Überschrift 3. Ordnung

Überschrift 4. Ordnung

Überschrift 5. Ordnung

Überschrift 6. Ordnung

3.2 Textabsätze definieren

- Absätze dienen der optischen Gliederung eines Textes. Beim Erstellen von HTML-Dateien genügt es nicht, im Editor einen harten Umbruch einzufügen. WWW-Browser ignorieren solche Umbrüche!

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
    <title>Textabsätze definieren</title>
  </head>
  <body>
    <h1>Textabsätze definieren</h1>
    <p>Hier beginnt ein Absatz, und hier ist er zu Ende.</p>
    <p>Hier beginnt ein neuer Absatz, und hier ist er zu Ende.</p>
  </body>
</html>
```

- Absätze dienen der optischen Gliederung eines Textes. Beim Erstellen von HTML-Dateien genügt es nicht, im Editor einen harten Umbruch einzufügen. WWW-Browser ignorieren solche Umbrüche!

3.3 Zeilenumbruch erzwingen

- Text innerhalb von normalen Absätzen, Listen, sowie in Überschriften oder Tabellenzellen wird vom Web-Browser bei der Anzeige automatisch umbrochen. Sie können jedoch an einer gewünschten Stelle einen Zeilenumbruch erzwingen.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<title>Zeilenumbruch erzwingen</title>
</head>
<body>
<h1>Ottos Brief an die Mutter</h1>
<p>Liebe Mama!</p>
<p>Vor drei Wochen habe ich erfahren,<br>
    dass Du krank bist.<br>
    Mach' Dir keine Sorgen.<br>
    Mir geht es gut.</p>
</body>
</html>
```

- `
` (*br = break = Umbruch*) fügt an der gewünschten Stelle einen Zeilenumbruch ein.
- Dabei ist es egal, ob das alleinstehende Tag am Ende der vorherigen Zeile steht (wie im Beispiel), oder in einer eigenen Zeile, oder am Anfang der folgenden Zeile.

3.4 Sonderzeichen

- Wenn Sie Kodierungen **wie ISO 8859-1** (vgl. Folie 22) verwenden, können Sie die deutschen Umlaute direkt in Ihrem Editor eintippen. Sofern sie die besagte Angabe zur Zeichenkodierung im Kopf der HTML-Datei notieren, können sie davon ausgehen, dass aktuelle und auch ältere Browser die Datei korrekt verarbeiten.
- Dennoch gibt es einige Sonderfälle, in denen es sinnvoll ist, sich auf die **ASCII-Zeichen** zu beschränken, um mögliche Probleme bei der Verarbeitung zu vermeiden.
- In diesem Fall können Sie deutsche Umlaute sowie das scharfe S durch **benannte Zeichen** umschreiben. Das gilt für den gesamten Inhalt einer HTML-Datei (vgl. Tabelle rechts).

Zeichen	Ersetzung
ä	ä
Ä	Ä
ö	ö
Ö	Ö
ü	ü
Ü	Ü
ß	ß

3.5 Listen (1)

- **Aufzählungslisten** sind z.B. von Bedeutung, um Produkteigenschaften oder Argumente für eine These übersichtlich darzustellen. Bei einer Aufzählungsliste werden alle Listeneinträge mit einem Aufzählungszeichen (Bullet) versehen.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<title>Aufzählungslisten definieren</title>
</head>
<body>
<h1>Aufzählungsliste</h1>

<ul>
  <li>1. Punkt </li>
  <li>2. Punkt</li>
  <li>3. Punkt </li>
</ul>

</body>
</html>
```

Aufzählungsliste

- ◆ 1. Punkt
- ◆ 2. Punkt
- ◆ 3. Punkt

- `` leitet eine Aufzählungsliste ein (*ul = unordered list = unsortierte Liste*).
- Mit `` beginnt ein neuer Punkt innerhalb der Liste (*li = list item = Listeneintrag*).
- `` beendet den Listeneintrag.
- Wie das Bullet dargestellt wird, bestimmt dabei der Web-Browser.
- Das Verschachteln von Listen ist ebenfalls möglich.

3.5 Listen (2)

- Für **Aufzählungslisten** können Sie den Bullet-Typ bestimmen:
 - Mit `<ul type="circle">` bestimmen Sie ein rundes Bullet (*circle = Kreis*).
 - Mit `<ul type="square">` bestimmen Sie ein eckiges Bullet (*square = Rechteck*).
 - Mit `<ul type="disc">` bestimmen Sie ein Dateisymbol als Bullet (*disc = Datenträger wie Disketten oder Festplatten*).

- **Nummerierte Liste**
 - `` leitet eine nummerierte Liste ein (*ol = ordered list = nummerierte Liste*).
 - Mit `` beginnt ein neuer Punkt innerhalb der Liste (*li = list item = Listeneintrag*).
 - `` beendet den Listeneintrag.
 - `` beendet die Liste.

- Für **nummerierte Listen** können Sie die Art der Nummerierung bestimmen:
 - Mit `<ol type="I">` werden die Listeneinträge mit I., II., III., IV. usw. nummeriert.
 - Mit `<ol type="i">` werden die Listeneinträge mit i., ii., iii., iv. usw. nummeriert.
 - Mit `<ol type="A">` werden die Listeneinträge mit A., B., C. usw. nummeriert.
 - Mit `<ol type="a">` werden die Listeneinträge mit a., b., c. usw. nummeriert.

3.6 Zitate definieren

- Sie können Zitate von Fremdautoren in einem eigenen, anders formatierten (zumeist eingerückten) Absatz hervorheben.
- Es handelt sich dabei jedoch um eine logische, inhaltliche Auszeichnung. Wie diese Absätze genau formatiert werden, bestimmt letztlich der Web-Browser. Die Vorgabe ist jedoch, solche Absätze auffällig und vom übrigen Text unterscheidbar anzuzeigen.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<title>Zitate definieren</title>
</head>
<body>
<h1>Franz Kafka</h1>

<p>Über die Krähen und den Himmel schreibt Franz Kafka:</p>

<bblockquote>
  <p>Die Krähen behaupten, eine einzige Krähe könne den Himmel
  zerstören; das ist zweifellos, beweist aber nichts gegen
  den Himmel, denn Himmel bedeutet eben: Unmöglichkeit von
  Krähen.</p>
</blockquote>

</body>
</html>
```

- `<blockquote>` leitet einen eigenen Absatz für Zitate ein (*blockquote* = *geblocktes Zitat*).
- `</blockquote>` beendet den Absatz.

3.7 Trennlinien definieren

- Trennlinien dienen der optischen Abgrenzung von nicht unmittelbar zusammengehörigen Textabschnitten oder allgemein zur Auflockerung. Eine Trennlinie erzeugt einen eigenen Absatz.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<title>Trennlinien definieren</title>
</head>
<body>

<p>Hier ist ein Abschnitt zu Ende.</p>
<hr>
<p>Und hier beginnt etwas Neues.</p>

</body>
</html>
```

Hier ist ein Abschnitt zu Ende.

Und hier beginnt etwas Neues.

- `<hr>` fügt eine Trennlinie ein (*hr* = *horizontal rule* = *Querlinie*).
- Dabei ist es egal, ob das Tag am Ende der Zeile des vorherigen Absatzes steht, oder in einer eigenen Zeile (wie im Beispiel), oder am Anfang des folgenden Absatzes.

Übersicht: HTML Grundlagen

- Aufgabenstellung
- 0: Einführung in HTML
- 1: Allgemeine Regeln für HTML
- 2: Dateiweite Einstellungen
- 3: Elemente für die Textstrukturierung
- **4: Tabellen**
- 5: Verweise
- 6: Grafiken
- 7: Formulare

4.1 Tabellen definieren

- Sie können in HTML Tabellen definieren, um tabellarische Daten darzustellen, oder um Text und Grafik attraktiver am Bildschirm zu verteilen.
- Obwohl Tabellen natürlich vornehmlich zur Darstellung tabellarischer Daten geschaffen wurden, sind sie in der heutigen Praxis des Web-Designs vor allem als Grundgestaltungsmittel für Seiten-Layouts nicht mehr wegzudenken. Puristen, die HTML so weit wie möglich von gestalterischen Aufgaben befreit sehen wollen, rümpfen darüber zwar die Nase - aber so klar sind die Grenzen zwischen Struktur und Gestaltung nicht immer. Die saubere Verteilung von Information auf einer Web-Seite ist durchaus ein Aspekt der Strukturierung, obwohl sie zugleich ein Aspekt der Gestaltung ist.
- Rein optisch lässt sich also grundsätzlich unterscheiden zwischen
 - Tabellen, die Gitternetzlinien haben (für tabellarische Daten), und
 - Tabellen ohne sichtbare Gitternetzlinien (so genannte "blinde Tabellen" für mehrspaltigen Text oder für Verteilung von Inhalten auf einer Web-Seite).

4.1 Tabellen definieren (2)

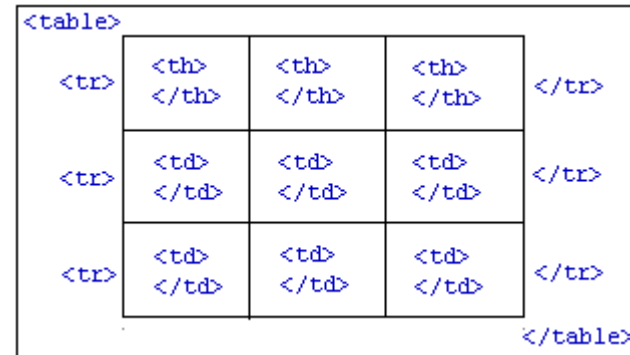
```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML
  4.01//EN"
  "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<title>Aufbau einer Tabelle</title>
</head>
<body>

  <table border="1">
    <tr>
      <th>Matrikelnummer</th>
      <th>Name</th>
      <th>Vorname</th>
    </tr>
    <tr>
      <td>123456</td>
      <td>Mustermann</td>
      <td>Max</td>
    </tr>
    <tr>
      <td>999999</td>
      <td>Müller</td>
      <td>Jörg</td>
    </tr>
  </table>

</body>
</html>

```



Matrikelnummer	Name	Vorname
123456	Mustermann	Max
999999	Müller	Jörg

4.1 Tabellen definieren (3)

- **<table>** leitet eine Tabelle ein (*table = Tabelle*). Wenn die Tabelle sichtbare Gitternetzlinien enthalten soll, müssen Sie im einleitenden `<table>`-Tag das Attribut `border` notieren und ihm einen Wert größer 0 zuweisen. Der angegebene Wert ist dann die Breite des Rahmens in Pixeln. Um eine blinde Tabelle ohne sichtbaren Rahmen und Gitternetzlinien zu erzeugen, lassen Sie die Angabe zu `border` entweder weg, oder - was sauberer ist - Sie notieren `border="0"`.
- **<tr>** leitet eine neue Tabellenzeile ein (*tr = table row = Tabellenzeile*). Im Anschluss daran werden die Zellen (Spalten) der betreffenden Reihe definiert. Am Ende einer Tabellenzeile wird ein abschließendes Tag `</tr>` notiert.
- Eine Tabelle kann Kopfzellen und gewöhnliche Datenzellen enthalten. Text in Kopfzellen wird hervorgehoben (meist fett und zentriert ausgerichtet). **<th>** leitet eine Kopfzelle ein, **<td>** eine normale Datenzelle (*th = table header = Tabellenkopf, td = table data = Tabellendaten*). Der Inhalt einer Zelle wird jeweils hinter dem Tag notiert.
- In einer Tabellenzelle können beliebige Elemente stehen, d.h. außer normalem Text z.B. auch andere Block- und Inline-Elemente. Sogar eine weitere Tabelle können Sie innerhalb einer Zelle definieren.

4.1 Tabellen definieren (4)

- Die Anzahl der Zellen sollte bei jeder Zeile gleich sein, so dass die Tabelle durchweg die gleiche Anzahl Spalten pro Zeile hat.
- In der ersten Zeile, die Sie definieren, legen Sie deshalb durch die Anzahl der dort definierten Zellen die Anzahl der Spalten Ihrer Tabelle fest.
- Tabellenzellen dürfen auch leer sein. Wenn Sie in einer Zeile für eine Spalte keine Daten eingeben wollen, notieren Sie ein einfaches `<td></td>`. Beachten Sie dabei jedoch, dass viele Web-Browser die Zelle in diesem Fall als "nicht vorhanden" darstellt. Probieren Sie deshalb auch mal die Notation `<td> </td>` für leere Tabellenzellen aus.

4.2 Spalten vordefinieren

- Die Darstellung einer Tabelle ergibt sich zwar automatisch aus den definierten Zeilen und Spalten. Doch für einen Web-Browser ist es nicht ganz einfach, die Darstellung frühzeitig zu ermitteln. Er muss erst die gesamte Tabelle einlesen, bevor er irgendetwas davon darstellen kann. Bei großen Tabellen kann das zu unschönen leeren Bildschirmhalten während des Seitenaufbaus führen.
- HTML 4.0 bietet eine Syntax an, um dem Browser gleich zu Beginn der Tabelle mitzuteilen, wie viele Spalten die Tabelle hat, und wie breit diese sind. Dadurch kann der Browser die Tabelle schneller aufbauen, d.h. bereits Teile der Tabelle anzeigen, bevor die gesamte Tabelle eingelesen ist.

```
<table border="1">
  <colgroup>
    <col width="80">
    <col width="100">
    <col width="320">
  </colgroup>
  <tr>
    <td>1. Zeile, 1. Spalte</td>
    <td>1. Zeile, 2. Spalte</td>
    <td>1. Zeile, 3. Spalte</td>
  </tr>
  <!-- usw. andere Zeilen der Tabelle -->
</table>
```

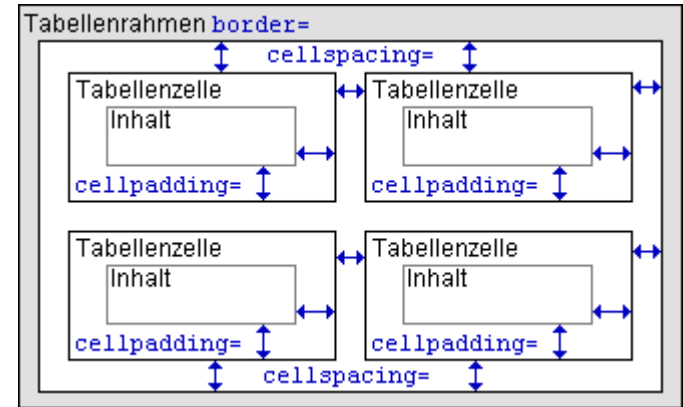
1. Zeile, 1. Spalte	1. Zeile, 2. Spalte	1. Zeile, 3. Spalte
------------------------	------------------------	---------------------

4.3 Zeilenabstand und Zeileninnenabstand

- Sie können den Abstand zwischen den Zeilen und Spalten einer Tabelle in Pixeln bestimmen. Ferner können Sie den Abstand zwischen Zellenrand und Zelleninhalt bestimmen.

```
<table border="8" cellspacing="10" cellpadding="20">
  <tr>
    <th>Matrikelnummer</th>
    <th>Name</th>
    <th>Vorname</th>
  </tr>
  <tr>
    <td>123456</td>
    <td>Mustermann</td>
    <td>Max</td>
  </tr>
  <tr>
    <td>999999</td>
    <td>Müller</td>
    <td>Jörg</td>
  </tr>
</table>
```

Matrikelnummer	Name	Vorname
123456	Mustermann	Max
999999	Müller	Jörg



- Mit **cellpadding** bestimmen Sie den Innenabstand einer Zelle, also den Abstand zwischen Zellenrand und Zelleninhalt in Pixeln (*cellpadding* = *Zelleninnenabstand*). Mit **cellspacing** bestimmen Sie den Abstand der Zellen untereinander in Pixeln (*cellspacing* = *Zellenabstand*).

4.4 Regeln für den Aussenrahmen

- Voraussetzung für all diese Angaben ist das Attribut `border` im einleitenden `<table>`-Tag. Dadurch wird ein Außenrahmen angezeigt.
- Mit dem Attribut `frame` können Sie dann bestimmen, an welchen Seiten der Tabellenrahmen gezogen werden soll (*frame = Rahmen*).
- Folgende andere Angaben sind möglich:
 - Mit **frame="above"** (*above = oberhalb*) wird nur am oberen Rand der Tabelle eine Rahmenlinie angezeigt.
 - Mit **frame="below"** (*below = unterhalb*) wird nur am unteren Rand der Tabelle eine Rahmenlinie angezeigt.
 - Mit **frame="hsides"** (*hsides = horizontal sides = horizontale Seiten*) wird nur am oberen und am unteren Rand der Tabelle eine Rahmenlinie angezeigt.
 - Mit **frame="vsides"** (*vsides = vertical sides = vertikale Seiten*) wird nur am linken und am rechten Rand der Tabelle eine Rahmenlinie angezeigt.
 - Mit **frame="lhs"** (*lhs = left hand side = links*) wird nur am linken Rand der Tabelle eine Rahmenlinie angezeigt.
 - Mit **frame="rhs"** (*rhs = right hand side = rechts*) wird nur am rechten Rand der Tabelle eine Rahmenlinie angezeigt.

4.5 Breiten- und Höhenangaben

- Sie können für die gesamte Tabelle geeignete Angaben zu Breite und Höhe notieren. Dadurch schaffen Sie "Raum".
- Breitenangaben werden durch das Attribut **width** notiert, Höhenangaben durch das Attribut **height** (*width = Breite, height = Höhe*). Bei allen Angaben sind absolute Zahlenangaben erlaubt, die Breite bzw. Höhe als Anzahl Pixel bestimmen, sowie prozentuale Angaben, die am Ende ein Prozentzeichen (%) haben.
 - **<table width="...">**
 - Bestimmt die Breite der gesamten Tabelle in Bezug auf den verfügbaren Raum im Elternelement. Das Elternelement kann das "Dokument" sein, also das body-Element, oder auch ein anderes Element, innerhalb dessen eine Tabelle vorkommen kann, z.B. ein div-Element oder eine Tabellenzelle einer äußeren Tabelle.
 - **<table height="...">**
 - Bestimmt die Höhe der gesamten Tabelle in Bezug auf das Anzeigefenster.

```
<table border="8" frame="vsides" width="60%">
...
</table>
```

60%

vsides

4.6 Ausrichtung von Zellen

links oben	zentriert oben	rechts oben
links mittig	zentriert mittig	rechts mittig
links unten	zentriert unten	rechts unten
alles zentriert mittig

- Zelleninhalte von Tabellen können aufgrund unterschiedlicher Inhalte oder durch Angaben zu Breite und Höhe mehr Raum in Anspruch nehmen, als es ihr Inhalt erfordert. Für diesen Fall können Sie Zelleninhalte sowohl horizontal (linksbündig, zentriert, rechtsbündig) als auch vertikal (obenbündig, mittig, untenbündig) am Zellenrand ausrichten.

```
<table border="1">
  <tr>
    <th align="left"    valign="top" width="200" height="75">links oben</th>
    <th align="center"  valign="top" width="200">zentriert oben</th>
    <th align="right"   valign="top" width="200">rechts oben</th>
  </tr>
  <tr>
    <td align="left"    valign="middle" height="75">links mittig</td>
    <td align="center"  valign="middle">zentriert mittig</td>
    <td align="right"   valign="middle">rechts mittig </td>
  </tr>
  <tr>
    <td align="left"    valign="bottom" height="75">links unten</td>
    <td align="center"  valign="bottom">zentriert unten</td>
    <td align="right"   valign="bottom">rechts unten</td>
  </tr>
  <tr align="center" valign="middle">
    <td height="75">alles ...</td>
    <td>... zentriert ...</td>
    <td>... mittig</td>
  </tr>
</table>
```

4.6 Ausrichtung von Zellen (2)

- Mit **align="left"** können Sie eine Kopfzelle in ihrem einleitenden <th>-Tag linksbündig ausrichten, mit **align="right"** rechtsbündig.
- Datenzellen können Sie in ihrem einleitenden <td>-Tag mit **align="center"** zentriert ausrichten und durch **align="right"** rechtsbündig.
- Die Angaben **align="center"** bei Kopfzellen und **align="left"** bei Datenzellen sind natürlich auch erlaubt, bewirken aber lediglich die Voreinstellung und sind deshalb nicht nötig (*align = Ausrichtung, left = links, center = zentriert, right = rechts*).
- Mit **valign="top"** können Sie eine Zelle in ihrem einleitenden Tag obenbündig ausrichten, durch die Angabe **valign="bottom"** untenbündig.
- Auch die Angabe **valign="middle"** (**nicht** "center"!) ist möglich. Dies ist jedoch die Voreinstellung und deshalb nicht unbedingt nötig (*valign = vertikale Ausrichtung, top = oben, bottom = unten, middle = mittig*).

4.7 Zellen in einer Zeile verbinden

- Sie können mehrere Zellen innerhalb einer Zeile verbinden, so dass sich eine Spalte in dieser Zeile über mehrere Spalten hinweg erstreckt.

```
<table border="1">
  <tr>
    <th colspan="2">Liste aller Studierenden </th>
  </tr>
  <tr>
    <td>Name</td>
    <td>Vorname</td>
  </tr>
</table>
```

Liste aller Studierenden	
Name	Vorname

- Durch das Attribut **colspan [Anzahl Spalten]** erreichen Sie, dass sich eine Zelle über mehrere Spalten hinweg erstreckt (*colspan = column span = Spalten spannen*). Die Angabe ist nur wirksam, wenn die Tabelle mindestens so viele Spalten besitzt wie angegeben.
- **Entsprechend gilt:** Sie können mehrere Zellen in einer Spalte verbinden, so dass sich eine Zeile in dieser Spalte über mehrere Zeilen hinweg erstreckt: Durch das Attribut **rowspan [Anzahl Zeilen]** erreichen Sie, dass sich eine Zelle in einer Spalte über mehrere Zeilen hinweg erstreckt (*rowspan = Zeilen spannen*).

Übersicht: HTML Grundlagen

- Aufgabenstellung
- 0: Einführung in HTML
- 1: Allgemeine Regeln für HTML
- 2: Dateiweite Einstellungen
- 3: Elemente für die Textstrukturierung
- 4: Tabellen
- **5: Verweise**
- 6: Grafiken
- 7: Formulare

5.1 Verweise

- **Verweise sind ein entscheidender Bestandteil jedes Hypertext-Projekts und der "intelligente Mehrwert" des World Wide Web.**
- **Mit Hilfe von Verweisen strukturieren Sie Ihr Projekt.** Wenn Sie beispielsweise eine HTML-Datei mit einer Projekt-Einstiegsseite und verschiedene HTML-Dateien für Themenseiten planen, brauchen Sie in der Einstiegsseite Verweise zu allen Themenseiten und in jeder Themenseite einen Rücksprungverweis zur Einstiegsseite. Erst dadurch wird aus der losen Dateisammlung ein zusammenhängendes Web-Projekt, eine Homepage.
- **Alle Verweise in HTML haben den gleichen Aufbau.** Das vereinfacht die Syntax, hat aber auch Nachteile: es gibt nämlich keine Verweise für bestimmte Aufgaben. Mittlerweile gibt es zwar auch Attribute für solche Zwecke, aber in der Praxis ist es Ihnen überlassen, wie Sie zum Beispiel Verweise so gestalten, dass dem Anwender klar wird, ob es sich um einen Rückverweis zur Einstiegsseite, um einen Querverweis zu einem assoziativ verwandten Thema, oder etwa um einen Verweis zu einer ganz anderen Web-Adresse handelt.
- Wichtig ist in allen Fällen ein aussagekräftiger Verweistext.
Schlecht ist: „zurück“ (wohin eigentlich?)
Besser ist: „zurück zur Einstiegsseite“
- Verweise lenken die Aufmerksamkeit des Anwenders sofort auf sich. Wenn Sie Verweise mitten im Text notieren, sollten Sie als Verweistext inhaltlich beschreibende Wörter anbieten, keine Wörter ohne Inhalt.

5.2 Beispiele für Verweise und Anker

- Beispiele für Verweise:

```
<p>
    <a href="http://www.tu-clausthal.de">Link zur Homepage der TU Clausthal</a>
    <a href="http://www.in.tu-clausthal.de/abteilungen/winf/lehre/ws0910/wil-
ws0910/">Wirtschaftsinformatik-1 Homepage</a>
</p>
```

- Sie können innerhalb einer HTML-Datei **Anker** definieren. **Dann können Sie Verweise zu solchen Ankern setzen, um einen Sprung genau an die Ankerstelle innerhalb der Datei zu veranlassen.**
- Der Verweis kann in der gleichen Datei stehen. Dann wird einfach ein Sprung innerhalb der angezeigten Seite ausgeführt. Der Verweis kann aber auch in einer anderen Datei stehen. Dann wird die Zieldatei geladen, und der Browser springt, sobald er die Stelle mit dem Anker geladen hat, an die entsprechende Stelle innerhalb der Datei.
- Ein Anker wird genau wie ein Verweis mit Hilfe des a-Elements erzeugt. Der Unterschied besteht darin, dass kein Attribut href notiert wird, sondern stattdessen ein Attribut name. Ein kompletter Anker sieht also so aus: **...**

```
<a name="kapitel1">Kapitel 1</a>
...Hier steht viel Text...

<a name="kapitel2">Kapitel 2</a>
...Hier steht wieder viel Text...

<a href="#kapitel1">Springe zu Kapitel 1</a>
<a href="#kapitel2">Springe zu Kapitel 2</a>
```

5.3 E-Mail-Verweise

- Sie können auf jede beliebige E-Mail-Adresse im Internet-Format einen Verweis setzen. Eine Internet-gerechte E-Mail-Adresse erkennen Sie an dem Zeichen @ in der Mitte der Adresse.
- Wenn der Anwender auf den Verweis klickt, kann er eine E-Mail an den betreffenden Empfänger absetzen.
- Normalerweise benutzen Anbieter von WWW-Seiten diese Möglichkeit, um Besuchern die Möglichkeit zu bieten, eine E-Mail an die eigene Adresse zu senden, zum Beispiel wegen Feedback zum Angebot. Sie können aber auch Verweise zu anderen E-Mail-Adressen anbieten.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<title>E-Mail-Verweise definieren</title>
</head>
<body>

<h1>Impressum</h1>

<p>Verantwortlich f&uuml;r den Inhalt dieser Seiten ist:<br>
    <a href="mailto:olivier.foalem@tu-clausthal.de">Olivier Foalem olivier.foalem@tu-clausthal.de</a>
</p>

</body>
</html>
```

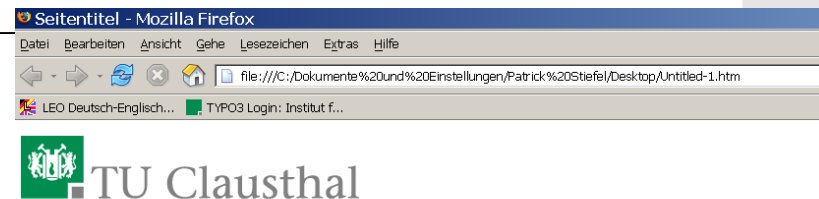
Übersicht: HTML Grundlagen

- Aufgabenstellung
- 0: Einführung in HTML
- 1: Allgemeine Regeln für HTML
- 2: Dateiweite Einstellungen
- 3: Elemente für die Textstrukturierung
- 4: Tabellen
- 5: Verweise
- **6: Grafiken**
- 7: Formulare

6.1 Grafiken einbinden

- Für Grafikreferenzen gibt es in HTML das ``-Tag (*img = image = Bild, src = source = Quelle*). Es handelt sich um ein Standalone-Tag ohne Elementinhalt und ohne End-Tag. Mit Hilfe von Attributen bestimmen Sie nähere Einzelheiten der Grafikreferenz. Zwei Attribute sind Pflicht und müssen immer angegeben werden:
 - Mit dem **Attribut src** bestimmen Sie die gewünschte Grafikdatei.
 - Wenn sich die Grafikdatei also beispielsweise im gleichen Verzeichnis wie die HTML-Datei befindet, in der die Grafikreferenz steht, dann genügt einfach die Angabe des Dateinamens der Grafik.
 - Das Referenzieren mit relativen oder absoluten Pfadangaben ist jedoch ebenso möglich, z.B.:
 - ``
 - ``
 - ``
 - ``
 - ``
 - ``
 - **Das Attribut alt ist Pflichtangabe** für jede Grafikeinbindung. Geben Sie darin einen Alternativtext an für den Fall, dass die Grafik nicht angezeigt werden kann.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<title>Seitentitel</title>
</head>
<body>
<p></p>
</body>
</html>
```



Übersicht: HTML Grundlagen

- Aufgabenstellung
- 0: Einführung in HTML
- 1: Allgemeine Regeln für HTML
- 2: Dateiweite Einstellungen
- 3: Elemente für die Textstrukturierung
- 4: Tabellen
- 5: Verweise
- 6: Grafiken
- **7: Formulare**

7.1 Formulare

- HTML stellt die Möglichkeit zur Verfügung, Formulare zu erstellen. In Formularen kann der Anwender Eingabefelder ausfüllen, in mehrzeiligen Textfeldern Text eingeben, aus Listen Einträge auswählen usw. Wenn das Formular fertig ausgefüllt ist, kann der Anwender auf einen Button klicken, um das Formular abzusenden.
- Dazu geben Sie beim Erstellen eines Formulars an, was mit den Daten des ausgefüllten Formulars passieren soll. Sie können sich die ausgefüllten Daten beispielsweise per E-Mail zuschicken lassen oder von einem CGI-Programm auf dem Server-Rechner weiterverarbeiten lassen.
- Formulare können sehr unterschiedliche Aufgaben haben. So werden sie zum Beispiel eingesetzt:
 - um bestimmte, gleichartig strukturierte Auskünfte von Anwendern einzuholen,
 - um Anwendern das Suchen in Datenbeständen zu ermöglichen,
 - um Anwendern die Möglichkeit zu geben, selbst Daten für einen Datenbestand beizusteuern,
 - um dem Anwender die Möglichkeit individueller Interaktion zu bieten, etwa um aus einer Produktpalette etwas Bestimmtes zu bestellen.
- Viele Suchdienste im Internet bieten dem aufrufenden Web-Browser Eingabeformulare an, in denen der Anwender seinen Suchwunsch spezifizieren kann. Ohne solche Formulare wäre das Durchsuchen gar nicht möglich.
- Immer zahlreicher werden auch die Online-Shops im Internet. Egal ob Tickets, Pizza oder Unterwäsche - um solche Bestell-Services zu realisieren, sind Formulare erforderlich, in denen der Anwender seine Bestellwünsche genau angeben kann.

7.2 Formularbereich definieren (1)

- Mit `<form>...</form>` definieren Sie ein Formular (*form = Formular*). Alles, was zwischen dem einleitenden `<form>`-Tag und dem abschließenden Tag `</form>` steht, gehört zum Formular. Das sind hauptsächlich Elemente des Formulars wie Eingabefelder, Auswahllisten oder Buttons.
- Um die Elemente zu platzieren und zu beschriften, können Sie dazwischen aber auch andere HTML-Elemente notieren.
- Im einleitenden `<form>`-Tag geben Sie mit dem Pflichtattribut `action` an, was mit den Formulardaten passieren soll, wenn der Anwender das Formular absendet (*action = Aktion*).
 - Die Wertzuweisung an `action` kann beispielsweise eine E-Mail-Adresse (normalerweise Ihre eigene) mit vorangestelltem `mailto:` sein. Dann werden die ausgefüllten Formulardaten an diese E-Mail-Adresse geschickt, sofern das möglich ist.
 - **Oder Sie rufen ein Programm auf dem Server-Rechner, meistens ein serverseitiges Skript, auf, das die Daten weiterverarbeitet - so wie im ersten der obigen Beispiele.**
 - Sie können bei `action` auch eine HTML-Datei angeben. Diese wird bei Absenden des Formulars aufgerufen und kann die Formulardaten z.B. mit JavaScript weiterverarbeiten. Das ist beispielsweise für mehrseitige Formulare interessant.

7.2 Formularbereich definieren (2)

- Ein weiteres wichtiges Attribut bei der Formulardefinition ist das Attribut **method**. Dabei bestimmen Sie, nach welcher HTTP-Übertragungsmethode die Formulardaten an ihr Ziel gelangen. Dabei gibt es zwei mögliche Wertzuweisungen:
 - Wenn Sie **method="get"** wählen, werden die Daten des ausgefüllten Formulars als Parameter an die Aufrufadresse angehängt. Die Anfrage, die beim Server eintrifft, sieht dann beispielsweise so aus:
`http://www.example.org/cgi-bin/feedback.cgi?AnwenderName=Stefan&AnwenderMail=beispiel@example.org`.
 - Das verarbeitende Script kann diese als Parameter übergebene Zeichenkette auslesen und für die Datenverarbeitung auseinanderdröseln.
 - Wenn Sie **method="post"** wählen, werden die Daten des ausgefüllten Formulars vom Web-Server über den Standardeingabekanal zur Verfügung gestellt, und ein auswertendes CGI-Script muss die ankommenden Daten behandeln wie eine Benutzereingabe, die auf der Kommandozeile gemacht wurde (*post = verschicken*).

```
<form action="http://www.example.org/cgi-bin/feedback.pl" method="get">
    <!-- hier folgen die Formularelemente -->
</form>
```

7.3 Einzeilige Eingabefelder definieren

- Einzeilige Eingabefelder dienen zur Aufnahme von einem oder wenigen Wörtern oder einer Zahl.

```
<form action="input_text.htm">
  <p>Vorname:<br><input name="vorname" type="text" size="30" maxlength="30"></p>
  <p>Zuname:<br><input name="zuname" type="text" size="30" maxlength="40"></p>
</form>
```

- **<input>** definiert ein einzeiliges Eingabefeld (*input = Eingabe*). Der Vollständigkeit halber sollten Sie die Angabe **type="text"** dazusetzen.
- Jedes Eingabefeld sollte einen internen Bezeichnernamen erhalten, und zwar mit dem Attribut **name**.
 - **Der vergebene Name wird bei Server-Skripten benötigt, um auf die Daten des Eingabefeldes zugreifen zu können.**
 - Der Name sollte nicht zu lang sein und darf keine Leerzeichen, Sonderzeichen oder deutsche Umlaute enthalten.
- Ferner sollten Sie bei einzeiligen Eingabefeldern immer die Anzeigelänge in Zeichen mit **size** sowie die interne Feldlänge in Zeichen mit **maxlength** bestimmen. Beide Angaben bedeuten die Anzahl Zeichen (*size = Größe, maxlength = maximal length = maximale Länge*).

7.4 Mehrzeilige Eingabebereiche definieren

- Mehrzeilige Eingabefelder dienen zur Aufnahme von Kommentaren, Nachrichten usw.

```

...
<h1>Ganz spontan</h1>

<form action="textarea.htm">
  <p>Welche HTML-Elemente fallen Ihnen ein, und was bewirken sie:<br>
    <textarea name="user_eingabe" cols="50" rows="10"></textarea>
  </p>
</form>
...

```

Ganz spontan

Welche HTML-Elemente fallen Ihnen ein, und was bewirken sie:

Hier tippen Sie Ihre Antwort.....

- `<textarea ...>` leitet ein mehrzeiliges Eingabefeld ein (*textarea = Textbereich*). Jedes mehrzeilige Eingabefeld muss ebenso wie ein einzeiliges Eingabefeld einen internen Bezeichnernamen erhalten, und zwar mit dem **Attribut name**.
- Pflicht ist die Angabe zur Höhe und Breite des Eingabebereichs. Das Attribut **rows** bestimmt die Anzahl der angezeigten Zeilen (*rows = Zeilen*) und damit die Höhe, während **cols** die Anzahl der angezeigten Spalten (*cols = columns = Spalten*) festlegt. "Spalten" bedeutet dabei die Anzahl Zeichen (pro Zeile).
- Mit `</textarea>` schließen Sie das mehrzeilige Eingabefeld ab.

7.5 Auswahllisten definieren

- Sie können dem Anwender eine Liste mit festen Einträgen anbieten, aus der er einen Eintrag auswählen kann. Der Text des ausgewählten Eintrags wird übertragen, wenn der Anwender das Formular abschickt.

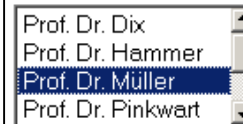
```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<title>Auswahllisten definieren</title>
</head>
<body>

<h1>Wählen Sie Ihren Favoriten!</h1>

<form action="select.htm">
  <p>
    <select name="top5" size="4">
      <option>Prof. Dr. Dix</option>
      <option>Prof. Dr. Hammer</option>
      <option>Prof. Dr. Müller</option>
      <option>Prof. Dr. Pinkwart</option>
      <option>Prof. Dr. Zachmann</option>
      <option>Prof. Dr. Müller</option>
    </select>
  </p>
</form>

</body>
</html>
```

Wählen Sie Ihren Favoriten!



Prof. Dr. Dix	▲
Prof. Dr. Hammer	
Prof. Dr. Müller	▼
Prof. Dr. Pinkwart	

7.5 Auswahllisten definieren (2)

- `<select ...>` leitet eine Auswahlliste ein. Jede Auswahlliste sollte einen internen Bezeichnernamen erhalten, und zwar mit dem Attribut **name**.
- Mit dem Attribut **size** bestimmen Sie die Anzeigegröße der Liste, d.h. wie viele Einträge angezeigt werden sollen. Wenn die Liste mehr Einträge enthält als angezeigt werden, kann der Anwender in der Liste scrollen. Wenn Sie **size="1"** angeben, definieren Sie eine so genannte "Dropdown-Liste".
- Mit `<option>...</option>` definieren Sie zwischen dem einleitenden `<select>`-Tag und dem Abschluss-Tag `</select>` jeweils einen Eintrag der Auswahlliste.
- Hinter `<option>` muss der Text des Listeneintrags stehen. Sie können so viele Listeneinträge definieren wie Sie wollen.

7.6 Radio-Buttons definieren

- Radio-Buttons sind eine Gruppe von beschrifteten Knöpfen, von denen der Anwender einen auswählen kann. Es kann immer nur einer der Radio-Buttons ausgewählt sein. Der Wert des ausgewählten Radio-Buttons wird beim Absenden des Formulars mit übertragen.

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<title>Radio-Buttons definieren</title>
</head>
<body>

<h1>Hier wird abkassiert!</h1>

<form action="input_radio.htm">
  <p>Geben Sie Ihre Zahlungsweise an:</p>
  <p>
    <input type="radio" name="Zahlungsmethode" value="Mastercard"> Mastercard<br>
    <input type="radio" name="Zahlungsmethode" value="Visa"> Visa<br>
    <input type="radio" name="Zahlungsmethode" value="AmericanExpress"> American Express
  </p>
</form>

</body>
</html>

```

Hier wird abkassiert!

Geben Sie Ihre Zahlungsweise an:

- Mastercard
- Visa
- American Express

Value
Jeder Button trägt jedoch seinen eigenen Wert!

Zahlungsmethode
Alle drei radiobuttons tragen den selben Namen; Dies führt zu einer Gruppierung! (vgl. nächste Folie)

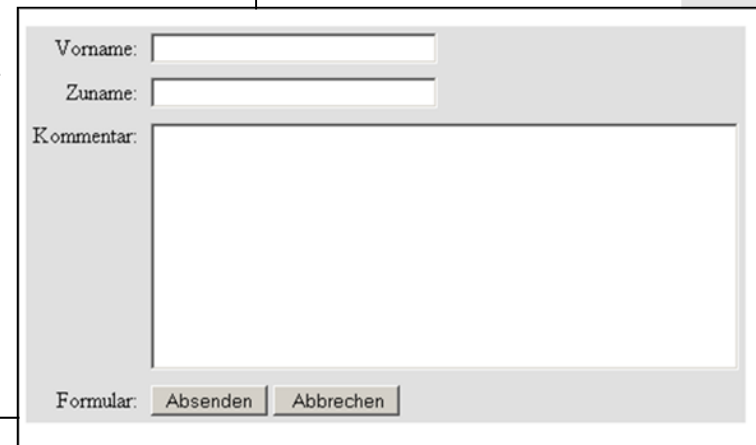
7.6 Radio-Buttons definieren (2)

- Radio-Buttons werden durch `<input type="radio">` definiert (*input = Eingabe*).
- Jeder Radio-Button sollte einen internen Bezeichnernamen erhalten, und zwar mit dem Attribut `name`. **Alle Radio-Buttons, die den gleichen Namen haben, gehören zu einer Gruppe, d.h. von diesen Buttons kann der Anwender genau einen markieren.**
- Mit dem Attribut **value** bestimmen Sie einen internen Bezeichnerwert für jeden Radio-Button (*value = Wert*). **Wenn der Anwender das Formular abschickt, wird der Bezeichnerwert des markierten Buttons übertragen.**
- Vor oder hinter dem `<input>`-Tag können Sie den Text notieren, der als Beschriftung der jeweiligen Option erscheint.
- Wenn Sie eine der Auswahlmöglichkeiten vorselektieren wollen, dann notieren Sie in dem `<input>`-Tag des entsprechenden Radio-Buttons das alleinstehende Attribut `checked`, also z.B.:
`<input type="radio" name="Typ" value="Kassenpatient" checked>`

7.7 Buttons zum Absenden oder Abbrechen definieren

- Zwei Standard-Buttons stellt HTML zur Verfügung, um Formulareingaben zu handhaben: einen Button zum "Absenden" und einen zum "Abbrechen".
 - Mit dem Absendebutton kann der Anwender das ausgefüllte Formular losschicken. Mit den Formulardaten geschieht dann das, was im einleitenden <form>-Tag mit dem Attribut action festgelegt wurde
 - Mit dem Abbrechen-Button kann der Anwender alle Eingaben verwerfen. Das Formular wird nicht abgeschickt, Eingaben werden gelöscht.

```
<form action="http://winf.in.tu-clausthal.de/foalem/ws0809/wdi-skript.php">
  <table border="0" cellpadding="5" cellspacing="0" bgcolor="#E0E0E0">
    <tr>
      <td align="right">Vorname:</td>
      <td><input name="vorname" type="text" size="30" maxlength="30"></td>
    </tr>
    <tr>
      <td align="right">Zuname:</td>
      <td><input name="zuname" type="text" size="30" maxlength="40"></td>
    </tr>
    <tr>
      <td align="right" valign="top">Kommentar:</td>
      <td><textarea name="Text" rows="10" cols="50"></textarea></td>
    </tr>
    <tr>
      <td align="right">Formular:</td>
      <td>
        <input type="submit" value=" Absenden " >
        <input type="reset" value=" Abbrechen" >
      </td>
    </tr>
  </table>
</form>
```



Übersicht Aufgabe 2: Einführung in CSS


▪ **Aufgabenstellung**

- I: Stylesheets in HTML einbinden
 - Formate zentral in HTML-Datei definieren
 - Formate zentral in separater CSS Datei definieren
 - Formate innerhalb eines HTML-Elements definieren
- II: CSS-Eigenschaften
 - Beispiel 1: Zentrale Formate für HTML-Elemente definieren
 - Hintergrundfarbe
 - Schriftfarbe
 - Einschub: Farben
 - Schriftart
 - Schriftstil und Schriftgröße
 - Außenrand und -abstand
 - Einschub: Numerische Angaben
 - Rahmen
 - Beispiel 2: Individualformate definieren
 - Innenabstand
 - Positionierung und Anzeige von Elementen
 - Zusatz 1: Formate für verschachtelte HTML-Elemente definieren
 - Zusatz 2: Formate für Klassen definieren
- III: CSS-basierte Layouts
 - Beispiel 3: Zweispaltiges CSS-basiertes Layout
 - Pseudoklassen für Verweise
 - Textumfluss

Aufgabenstellung: Überblick

- In dieser Aufgabe erweitern Sie ihre vorliegende HTML Dokument (feedback.html) mit einer geeigneten CSS-Definition.

Feedbackformular



TU Clausthal

Werkzeug der Informatik: HTML und CSS

Vielen Dank, dass Sie uns ein Feedback senden wollen!

Vorname	<input type="text" value="Olivier"/>
Name	<input type="text" value="Foalem"/>
Strasse	<input type="text" value="Julius-Albert-Str. 4"/>
PLZ und Ort	<input type="text" value="38678"/> <input type="text" value="Clausthal-Zellerfeld"/>
E-Mail	<input type="text" value="olivier.foalem@tu-clausthal.de"/>
Ihre Nachricht an uns	<pre>Hallo! Dies ist eine Testnachricht. Freundliche Grüße,</pre>
Bitte beurteilen Sie den aktuellen Aufgabenzettel	<p><input type="radio"/> Ich finde den Aufgabenzettel zu schwer</p> <p><input checked="" type="radio"/> Ich finde den Aufgabenzettel angemessen</p> <p><input type="radio"/> Mir sind die Aufgaben zu leicht</p>

Aufgabestellung: Details

- **Arbeiten Sie zunächst die Beispiele in diesem Skript durch, um die grundlegenden Prinzipien von CSS zu erlernen. Mit Hilfe der Beispiele können Sie Ideen zur Bearbeitung der Aufgabe sammeln.**
- Als Schriftart verwenden Sie bitte Helvetica, Arial oder eine Schriftart im Stil „sans-serif“
- Das Stylesheet definieren Sie bitte zentral in der HTML-Datei (vgl. Folie 73).

Aufgabenstellung: Das Stylesheet erstellen

- **Als erstes definieren Sie das Format für den „body“:**
 - Innenabstand allgemein: 1em
 - Aussenabstand allgemein: 0
 - Schriftfamilie: Helvetica,Arial,sans-serif
 - Schriftfarbe: schwarz

- **Jetzt formatieren Sie die Überschrift „h1“ (Werkzeug der Informatik: HTML und CSS) wie folgt:**
 - Schriftgröße: 1.5 em
 - Aussenabstand nach oben, rechts und links: 0
 - Aussenabstand nach unten: 1em
 - Innenabstand allgemein: 0.3em
 - Rahmen: „2px ridge silver“

- **Nun formatieren Sie ihre Tabelle „Table“**
 - Hintergrundfarbe
 - Rahmenmodell: collapse
 - **Der Tabellenbereich „TH“ :**
 - Textumfluss: Element steht in der Mitte
 - Schriftart: Fett
 - Rahmen: „2px solid #808080“
 - Hintergrundfarbe: #CCCCCC

 - **Der Tabellenbereich „TD“:**
 - Rahmen: „1px dashed silver“

Übersicht: Einführung in CSS

- Aufgabenstellung
- **I: Stylesheets in HTML einbinden**
 - **Formate zentral in HTML-Datei definieren**
 - **Formate zentral in separater CSS Datei definieren**
 - **Formate innerhalb eines HTML-Elements definieren**
- II: CSS-Eigenschaften
 - Beispiel 1: Zentrale Formate für HTML-Elemente definieren
 - Hintergrundfarbe
 - Schriftfarbe
 - Einschub: Farben
 - Schriftart
 - Schriftstil und Schriftgröße
 - Außenrand und Abstand
 - Einschub: Numerische Angaben
 - Rahmen
 - Beispiel 2: Individualformate definieren
 - Innenabstand
 - Positionierung und Anzeige von Elementen
 - Zusatz 1: Formate für verschachtelte HTML-Elemente definieren
 - Zusatz 2: Formate für Klassen definieren
- III: CSS-basierte Layouts
 - Beispiel 3: Zweispaltiges CSS-basiertes Layout
 - Pseudoklassen für Verweise
 - Textumfluss

Formate zentral in HTML-Datei definieren

- Sie können im head-Element einer HTML-Datei einen Bereich für CSS-Formate definieren.
- Mit `<style...> ... </style>` notieren Sie einen Bereich für Stylesheet-Formatdefinitionen (*style = Stil, Format*).
- Im einleitenden `<style>`-Tag müssen Sie den MIME-Typ der Stylesheet-Sprache angeben. Für CSS ist das die Angabe `type="text/css"`.
- Zwischen dem einleitenden Tag und dem abschließenden `</style>` können Sie dann zentrale CSS-Formate definieren.

```
<html>

<head>
<title>Seitentitel</title>
<style type="text/css">
  <!-- /* ... Hier werden die Formate definiert ... */ -->
</style>
</head>

<body>
</body>
</html>
```

Formate zentral in separater CSS Datei definieren

Im Dateikopf einer HTML-Datei können Sie mit `<link...>` eine CSS-Datei referenzieren, die CSS-Formatdefinitionen enthält (*link = Verweis*).

Innerhalb des `<link>`-Tags müssen die Angaben `rel="stylesheet" type="text/css"` stehen (*rel = relation = Bezug, type = MIME-Typ*).

Beim Attribut `href` geben Sie die gewünschte Datei an (*href = hyper reference = Hypertext-Referenz*).

Wenn sich die CSS-Datei in einem anderen Verzeichnis oder auf einem anderen Server befindet, müssen Sie an dieser Stelle Pfadangaben oder absolute URIs notieren.

```
<html>
<head>
  <title>Seitentitel</title>
  <link rel="stylesheet" type="text/css" href="„stylesheet.css">
</head>
<body>
</body>
</html>
```


Formate innerhalb eines HTML-Elements definieren

- Sie können Formate für ein HTML-Element definieren, indem Sie innerhalb des einleitenden HTML-Tags das Attribut `style` und dahinter die gewünschten Formatdefinitionen notieren.
- Innerhalb der Formatdefinitionen sind CSS-Eigenschaften erlaubt. Dabei gilt die gleiche Syntax wie innerhalb der geschweiften Klammern bei zentralen Formaten.
- Jede Eigenschaft besteht aus einem Namen, z.B. `color` (Schriftfarbe) und einem zugehörigen Wert, z.B. `#D8FD02`, getrennt durch einen Doppelpunkt.
- Schließen Sie jede Formatdefinition jeweils durch einen Strichpunkt ab. Nur bei der letzten Formatdefinition vor dem abschließenden Anführungszeichen darf der Strichpunkt entfallen.

```
<html>
<head>
  <title>Seitentitel</title>
  <style type="text/css"> body { background-color:#000000 } </style>
</head>
<body>
  <p style="background-color:#808040; color:#D8FD02;...">Hier steht Text.</p>
</body>
</html>
```

Beispiel 1: Zentrale Formate für HTML-Elemente definieren

- Um ein zentrales Format für alle HTML-Elemente eines bestimmten Typs zu definieren, notieren Sie als Selektor den Namen des HTML-Elementtyps, und zwar ohne spitze Klammern.
- Im nachfolgenden Beispiel (1) werden
 - body (Dokument),
 - h1 (Überschriften 1. Ordnung),
 - p (Textabsätze) und
 - li (Listeneinträge) auf diese Weise notiert.
- Über den Universalselektor * haben Sie die Möglichkeit, Eigenschaften für *alle* Elemente zu definieren, wobei jedem Element natürlich nur die jeweils zulässigen Eigenschaften zugewiesen werden.

Beispiel 1: Code und Ergebnis (1)

```
<html>
<head>
<title>Seitentitel</title>
<style type="text/css">
  body { background-color:#FFFFCC; margin-left:100px; }
  * { color:blue; }
  h1 { font-size:300%; color:#FF0000; font-style:italic; border-bottom:solid thin
black; }
  p,li { font-size:140%; font-family:Helvetica,Arial,sans-serif;
letter-spacing:0.3em; }
</style>
</head>
<body>
<h1>&U
<p>ein normaler Textabsatz</p>
<ul>
  <li>Ein Listenpunkt</li>
  <li>Ein anderer Listenpunkt</li>
</ul>
</body>
</html>
```

Dem body-Element wird eine hellgelbe Hintergrundfarbe (background-color:#FFFFCC;) und ein linker Randabstand von 100 Pixeln (margin-left:100px;) zugewiesen.

Überschrift 1. Ordnung

ein normaler Textabsatz

- Ein Listenpunkt
- Ein anderer Listenpunkt

[zurück](#)

Beispiel 1: Code und Ergebnis (2)

```
<html>
<head>
<title>Seitentitel</title>
<style type="text/css">
  body { background-color:#FFFFCC; margin-left:100px; }
  * { color:blue; }
  h1 { font-size:300%; color:#FF0000; font-style:italic; border-bottom:solid thin
black; }
  p,li { font-size:100%; line-height:140%; font-family:Helvetica,Arial,sans-serif;
letter-spacing:0.1em; }
</style>
</head>
<body>
<h1>&U
<p>ein normaler Textabsatz</p>
<ul>
  <li>Ein Listenelement</li>
  <li>Ein anderer Listenelement</li>
</ul>
</body>
</html>
```

Über den Universalselektor wird für alle Elemente zunächst blaue Schriftfarbe (color:blue;) definiert.

Überschrift 1. Ordnung

ein normaler Textabsatz

- Ein Listenelement
- Ein anderer Listenelement

[zurück](#)

Beispiel 1: Code und Ergebnis (3)

```
<html>
<head>
<title>Seitentitel</title>
<style type="text/css">
  body { background-color:#FFFFCC; margin-left:100px; }
  * { color:blue; }
  h1 { font-size:300%; color:#FF0000; font-style:italic; border-bottom:solid thin
  black; }
  p,li { font-size:100%; line-height:140%; font-family:Helvetica,Arial,sans-serif;
  letter-spacing:-0.1em; text-indent:-1em; padding-left:1em; padding-right:1em; }
</style>
</head>
<body>
<h1>&U
<p>ein normaler Textabsatz</p>
<ul>
  <li>Ein Listenelement</li>
  <li>Ein anderer Listenelement</li>
</ul>
</body>
</html>
```

Überschriften 1. Ordnung (h1) erhalten eine Schriftgröße von 300 Prozent (font-size:300%;), die abweichende Schriftfarbe rot (color:#FF0000;) und den Schriftstil kursiv (font-style:italic;).

Überschrift 1. Ordnung

ein normaler Textabsatz

- Ein Listenelement
- Ein anderer Listenelement

[zurück](#)

Beispiel 1: Code und Ergebnis (4)

```
<html>
<head>
<title>Sei
<style type="text/css">
  body { background-color: #e0e0e0; }
  * { color: black; }
  h1 { font-size: 1.2em; }
  p,li { font-size:110%; line-height:140%; font-family:Helvetica,Arial,sans-serif;
letter-spacing:0.1em; word-spacing:0.3em; }
</style>
</head>
<body>
<h1>&Uuml;berschrift 1. Ordnung</h1>
<p>ein normaler Textabsatz</p>
<ul>
  <li>Ein Listenpunkt</li>
  <li>Ein anderer Listenpunkt</li>
</ul>
</body>
</html>
```

Textabsätze (p) und Listenpunkte (li) sollen in 110 Prozent Schriftgröße (font-size:110%;) mit einer Zeilenhöhe von 140 Prozent (line-height:140%;) in Schriftart Helvetica, oder wenn nicht verfügbar, dann in Arial, und wenn ebenfalls nicht verfügbar, in irgendeiner Sans-Serif-Schrift dargestellt werden (font-family:Helvetica,Arial,sans-serif;).

Überschrift 1. Ordnung

ein normaler Textabsatz

- Ein Listenpunkt
- Ein anderer Listenpunkt

[zurück](#)

Details zum Beispiel 1: Hintergrundfarbe

■ **background-color**

- Sie können für jedes HTML-Element eine eigene Hintergrundfarbe definieren.
- Sie können Farbwerte **hexadezimal** nach dem **Schema #RRGGBB** angeben, **aber auch die verbreiteten Farbnamen verwenden**. R ist der Rotwert, G der Grünwert und B der Blauwert der Farbe.
- Eine weitere Möglichkeit ist das **Schema rgb(rrr,ggg,bbb)**. Innerhalb der Klammern müssen drei Dezimalwerte stehen, durch Kommata voneinander getrennt.
- Für alle drei Werte sind entweder absolute Zahlen zwischen 0 (kein Anteil der entsprechenden Farbe) und 255 (maximaler Anteil der entsprechenden Farbe) erlaubt.

```
<style type="text/css">
body { background-color:rgb(51,0,102) }
h1 { color:#FFCCDD; background-color:black }
</style>
```

Details zum Beispiel 1: Schriftfarbe

black	#000000	gray	#808080
maroon	#800000	red	#FF0000
green	#008000	lime	#00FF00
olive	#808000	yellow	#FFFF00
navy	#000080	blue	#0000FF
purple	#800080	fuchsia	#FF00FF
teal	#008080	aqua	#00FFFF
silver	#C0C0C0	white	#FFFFFF

■ color

- Mit dieser Angabe können Sie Textvordergrundfarbe bzw. Schriftfarbe bestimmen.
- Im Beispiel 1 wurde für die Überschrift die Farbe rot gewählt:
`h1 { font-size:300%; color:#FF0000; ... }`

■ Einschub: „Farben“

- In CSS gibt es wie in HTML die Möglichkeit, mit hexadezimalen Ziffern die 16 Grundfarben, die von jedem VGA-kompatiblen Bildschirm angezeigt werden können, zu definieren.
- Die Beschriftungen rechts zeigen noch einmal, welchen Hexadezimalnotationen entsprechend der Syntax von HTML und CSS die jeweiligen Farben entsprechen.
- Um eine Farbe mit Hilfe eines Farbnamens zu definieren, geben Sie anstelle des hexadezimalen RGB-Werts einfach den gewünschten Farbnamen an.

Details zum Beispiel 1: Schriftart

■ font-family

- **Schriftarten** sind beispielsweise Arial, Helvetica, Times Roman usw.
- Auch **Schriftfamilien** wie *Sans Serif* usw. gehören dazu.
- Bei der hier beschriebenen Eigenschaft können Sie Schriftarten angeben, ohne sich darum kümmern zu müssen, ob und wie die Schriftart beim Anwender angezeigt werden kann.
- Mit font-family können Sie **eine oder mehrere Schriftarten** bestimmen. Bei mehreren angegebenen Schriftarten ist die Reihenfolge der Angabe entscheidend: Ist die erste angegebene Schriftart verfügbar, wird diese verwendet. Ist sie nicht verfügbar, wird die zweite Schriftart verwendet, falls diese verfügbar ist usw.
- Im Beispiel sollen Textabsätze und Listenpunkte auf jeden Fall eine serifenlose Schriftart (sans-serif) verwendet werden (vgl. Folie 13):

```
p,li { ... font-family:Helvetica,Arial,sans-serif; ... }
```

<small>kleiner Beispieltext in Times New Roman</small> großer Beispieltext in Times New Roman
<small>kleiner Beispieltext in Verdana</small> großer Beispieltext in Verdana
<small>kleiner Beispieltext in Wide Latin</small> großer Beispieltext in Wide Latin
<small>kleiner Beispieltext in Comic Sans MS</small> großer Beispieltext in Comic Sans MS
<small>kleiner Beispieltext in Avantgarde</small> großer Beispieltext in Avantgarde
<small>kleiner Beispieltext in Balloon</small> großer Beispieltext in Balloon
<small>kleiner Beispieltext in Bodoni</small> großer Beispieltext in Bodoni
<small>kleiner Beispieltext in Garamond</small> großer Beispieltext in Garamond
<small>kleiner Beispieltext in Impact</small> großer Beispieltext in Impact
<small>kleiner Beispieltext in Helvetica</small> großer Beispieltext in Helvetica

Details zum Beispiel 1: Schriftstil und Schriftgröße

■ font-style

- Schriftstil bedeutet die Neigung der Schrift.
 - **italic** = kursiver Schriftstil
 - **oblique** = schräggestellter Schriftstil
 - **normal** = normaler Schriftstil

kleiner Beispieltext mit Schriftstil italic

großer Beispieltext mit Schriftstil italic

■ font-size

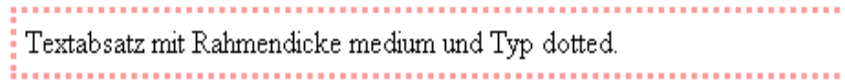
- Erlaubt sind
 - numerische Angaben (z.B. pt, px, em)
 - Prozentangaben (Prozentwerte beziehen sich auf die Schriftgröße des Elternelements.)
 - Absolute Schlüsselwerte
 - xx-small = winzig
 - x-small = sehr klein
 - small = klein
 - medium = mittel
 - large = groß
 - x-large = sehr groß
 - xx-large = riesig
- Beispiel:
 - `font-size: 300%` (Schriftgröße von 300 Prozent)

Details zum Beispiel 1: Rahmen

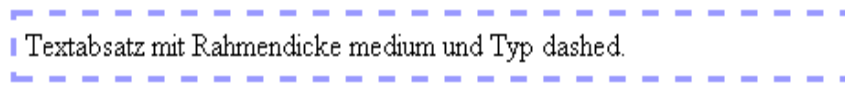
- Ein Rahmen ist anwendbar bei HTML-Elementen, die einen eigenen Absatz erzeugen bzw. einen Block bilden, also etwa für **h[1-6], p, div, table, usw.**
- **border[-top, -right, -bottom, -left]-width** (Rahmendicke)
 - Sie können für ein Element die Rahmendicke bestimmen. Sinnvollerweise sollten Sie diese Angabe zumindest mit einer Angabe zum **Rahmentyp** kombinieren, gegebenenfalls aber auch mit einer Festlegung der **Rahmenfarbe**.
 - Mit **border-width**: können Sie die Dicke des Rahmens um ein Element bestimmen. Erlaubt ist eine numerische Angabe (mit Ausnahme von Prozentwerten) für die Rahmendicke oder einer der folgenden Werte: **thin** = dünn, **medium** = mittelstark oder **thick** = dick.
 - **border-top-width** definiert die Rahmendicke für oben, **border-right-width** die Rahmendicke für rechts, **border-bottom-width** die Rahmendicke für unten und **border-left-width** die Rahmendicke für links.

Details zum Beispiel 1: Rahmen (2)

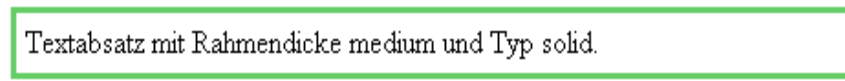
```
<p style="border-width:medium; border-color:#FF9999;  
border-style:dotted; padding:5px;">  
Textabsatz mit Rahmendicke medium und Typ dotted.</p>
```



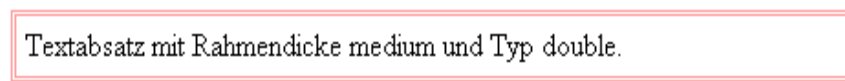
```
<p style="border-width:medium; border-color:#9999FF;  
border-style:dashed; padding:5px;">  
Textabsatz mit Rahmendicke medium und Typ dashed.</p>
```



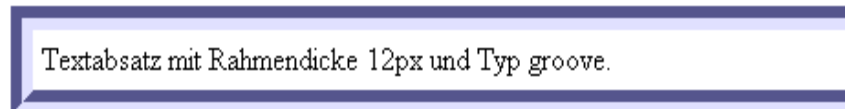
```
<p style="border-width:medium; border-color:#66CC66;  
border-style:solid; padding:5px;"> Textabsatz mit  
Rahmendicke medium und Typ solid.</p>
```



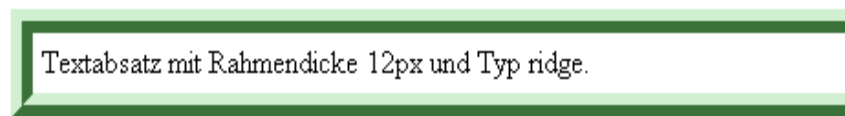
```
<p style="border-width:medium; border-color:#FF9999;  
border-style:double; padding:5px;"> Textabsatz mit  
Rahmendicke medium und Typ double.</p>
```



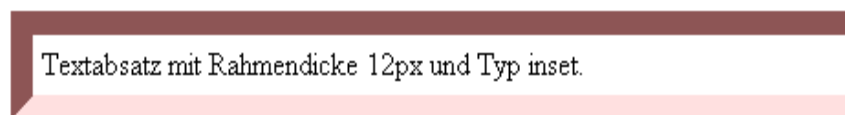
```
<p style="border-width:12px; border-color:#9999FF;  
border-style:groove; padding:5px;"> Textabsatz mit  
Rahmendicke 12px und Typ groove.</p>
```



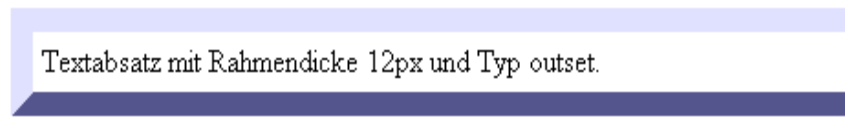
```
<p style="border-width:12px; border-color:#66CC66;  
border-style:ridge; padding:5px;"> Textabsatz mit  
Rahmendicke 12px und Typ ridge.</p>
```



```
<p style="border-width:12px; border-color:#FF9999;  
border-style:inset; padding:5px;"> Textabsatz mit  
Rahmendicke 12px und Typ inset.</p>
```



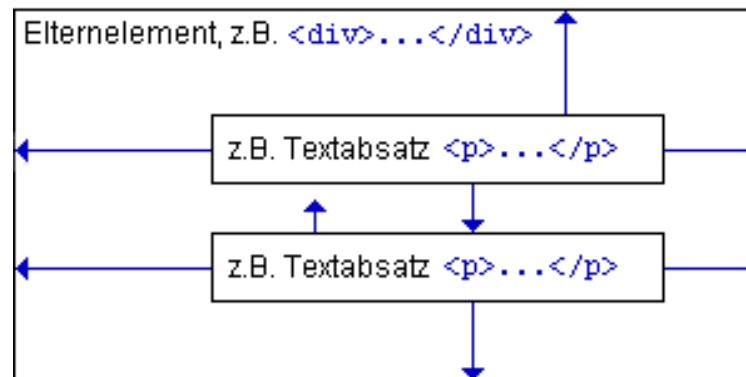
```
<p style="border-width:12px; border-color:#9999FF;  
border-style:outset; padding:5px;"> Textabsatz mit  
Rahmendicke 12px und Typ outset.</p>
```



Details zum Beispiel 1: Außenrand und -abstand

■ margin(-top/-right/-bottom/-left)

- Außenrand oder Außenabstand bedeutet:
Erzwungener Leerraum zwischen dem aktuellen Element und seinem Elternelement oder Nachbarelement.
- Für ein p-Element, also einen Textabsatz etwa, der direkt innerhalb des body-Elements notiert ist, markieren Angaben zu *linkem und rechten* Außenrand seinen Abstand zu den Elementgrenzen des body-Elements.
- Wenn mehrere solcher p-Absätze aufeinander folgen, markieren Angaben zum Außenrand *oben und unten* den Abstand zwischen den Absätzen. Das folgende Bild veranschaulicht die Möglichkeiten, Außenränder und Abstände zu definieren:

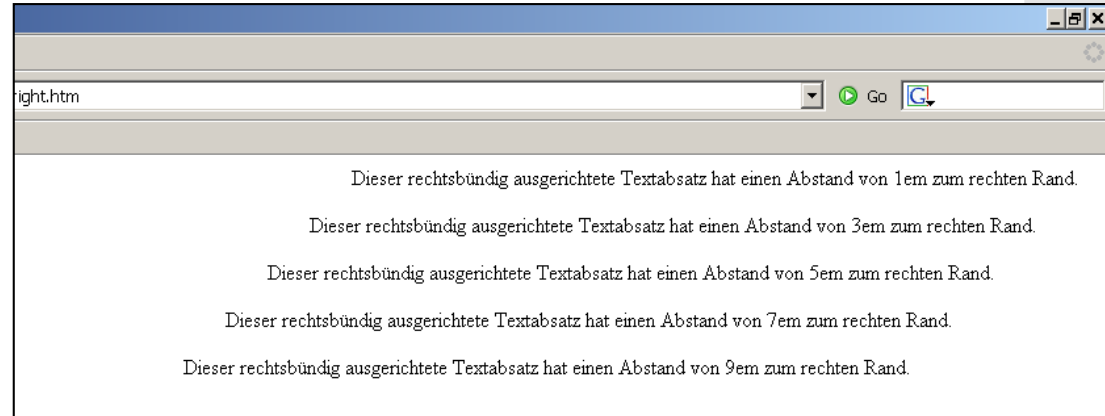


Details zum Beispiel 1: Außenrand und -abstand (2)

■ **margin right:**

```
<p align="right" style="margin-right:1em">  
Dieser rechtsbündig ausgerichtete  
Textabsatz hat einen Abstand von  
1em zum rechten Rand.</p>
```

```
<p align="right" style="margin-right:3em">  
Dieser rechtsbündig ausgerichtete  
Textabsatz hat einen Abstand von  
3em zum rechten Rand.</p>
```



■ **Einschub: „Numerische Angaben“**

- **pt (absolut):** Steht für **Punkt**. Typografische Maßeinheit. 1 Punkt entspricht 1/72 Inches. [1in = 0,0254 m]
- **px (absolut/ relativ):** Steht für **Pixel**. Abhängig von der Pixeldichte des Ausgabegeräts, relativ also von Ausgabegerät zu Ausgabegerät, absolut dagegen auf ein und das selbe Ausgabegerät bezogen.
- **em (relativ):** Steht für **Ems**. 1em entspricht der Größe des Buchstabens M in der gewählten Schriftart, basierend auf der Schriftgröße des gesamten Dokuments bzw. des Eltern-Elements. Sie können Angaben in em durch Dezimalzahlen ausdrücken, wobei 1.5em 150% der Größe des Buchstabens M entspricht.

Beispiel 2: Textumfluss

■ float

- Sie können für einen Bereich oder ein Element bestimmen, dass nachfolgende Elemente diesen Bereich bzw. dieses Element umfließen.
- Folgende Angaben sind möglich:
 - left = Element steht links und wird rechts davon von nachfolgenden Elementen umflossen.
 - right = Element steht rechts und wird links davon von nachfolgenden Elementen umflossen.

Überschrift mal anders

Eine Überschrift muss nicht immer einen langweiligen, über die ganze Breite des Anzeigefensters gehenden Block erzeugen. Sie kann auch mal seitlich vom Text stehen, und der Text fließt um die Überschrift. Auch so kann man lange Texte mal anders gestalten als sonst üblich.

```
<h1 style="width:400px; float:left; color:red">
&Uuml;berschrift mal anders</h1>
```

```
<p style="font-size:120%">Eine &Uuml;berschrift
muss nicht immer einen langweiligen, &uuml;ber die
ganze Breite des Anzeigefensters gehenden Block
erzeugen. Sie kann auch mal seitlich vom Text
stehen, und der Text flie&szlig;t um die
&Uuml;berschrift. Auch so kann man lange Texte mal
anders gestalten als sonst &uuml;blich.</p>
```

Details zum Beispiel 3: Fortsetzung bei Textumfluss

- Wenn Sie mit float einen Textumfluss definieren, möchten Sie vielleicht nur zwei oder drei Zeilen kommentierenden Text neben einem umflossenen Element haben und den folgenden Text dann unterhalb des umflossenen Elements fortsetzen.
- Mit clear können Sie einen Umfluss abbrechen und die Fortsetzung unterhalb des umflossenen Elements oder Bereichs erzwingen. Folgende Angaben sind möglich:
 - **left** = Erzwingt bei float:left die Fortsetzung unterhalb.
 - **right** = Erzwingt bei float:right die Fortsetzung unterhalb.
 - **both** = Erzwingt in jedem Fall die Fortsetzung unterhalb.

```

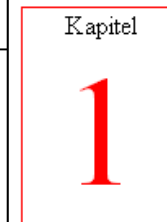
<div style="float:left;
margin-right:20px;
margin-bottom:20px;
border:solid 1px red;
text-align:center"> Kapitel

  <span style="font-family:Algerian,serif;
font-size:96px;
color:red"> 1

</span>

</div>
<p>In diesem Kapitel geht es um Einf&uuml;hrendes.</p>
<p style="clear:left">Dann kommen wir mal zur Sache: ...</p>

```



In diesem Kapitel geht es um Einführendes.

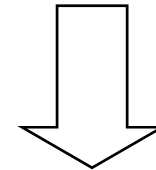
Dann kommen wir mal zur Sache: ...

Zusatz 1:

Formate für verschachtelte HTML-Elemente definieren

- Wenn Sie nichts anderes angeben, übernimmt ein HTML-Element, das innerhalb eines anderen HTML-Elements vorkommt, dessen Eigenschaften und fügt seine eigenen Eigenschaften nur hinzu.
- Wenn Sie beispielsweise für Überschriften 1. Ordnung die Schriftart Times und die Farbe rot definieren, erscheint Text, der innerhalb einer solchen Überschrift mit `<i>...</i>` formatiert wird, ebenfalls rot und in Times, aber zusätzlich kursiv.
- Sie können mit Hilfe von Stylesheets z.B. bestimmen, dass ein HTML-Element bestimmte Eigenschaften nur dann hat, wenn es innerhalb eines bestimmten anderen HTML-Elements vorkommt.
→ **So können Sie etwa bestimmen, dass `<i>...</i>` innerhalb von Überschriften nicht kursiv, stattdessen aber in blauer Farbe dargestellt wird, während das gleiche Element innerhalb anderer HTML-Tags nach wie vor nichts anderes als eine kursive Darstellung bewirkt.**

```
<html>
<head>
<title>Titel der Datei</title>
<style type="text/css">
    h1 { color:red; }
    h1 i { color:blue; font-style:normal; }
</style>
</head>
<body>
    <h1>Wir lernen <i>Stylesheets</i></h1>
    <p>Wir lernen <i>Stylesheets</i></p>
</html>
```



Wir lernen Stylesheets

Wir lernen Stylesheets

Zusatz 2: Formate für Klassen definieren

- Sie können Formate für Klassen definieren. Anwenden können Sie solche Klassen in HTML mit dem Universalattribut **class**.
- So können Sie beispielsweise für Überschriften 1. Ordnung zwei Varianten erzeugen. Dazu vergeben Sie Namen für die Klassen:

```
<html>
<head>
<title>Titel der Datei</title>
<style type="text/css">
    h1 { font-family:Arial,sans-serif; font-size:2em; font-weight:normal; }
    h1.hinterlegt { background-color:#FFFF00 }
</style>
</head>
<body>
<h1>H1 ganz normal nur etwas formatiert</h1>
<h1 class="hinterlegt">H1 knallgelb hinterlegt</h1>
</body>
</html>
```

- Es gibt zwei Möglichkeiten, Klassen für HTML-Elemente zu notieren:
 - Entweder für einen bestimmten HTML-Elementtyp,
 - oder für keinen bestimmten.
- In jedem Fall bestimmen Sie eine Formatdefinition für eine Klasse, indem Sie im Selektor einen Punkt notieren und unmittelbar dahinter einen Namen für die Klasse angeben.
- **Im obigen Beispiel wird mit `h1.hinterlegt` eine Klasse namens „hinterlegt“ angesprochen, die aber nur für HTML-Elemente vom Typ `h1` gilt: also z.B. auf HTML-Elemente wie `<h1 class="hinterlegt">` anwendbar ist.**