

***Einführung HTML******Inhalt***

Inhalt .....	1
Vorbemerkung .....	2
Grundbegriffe .....	2
Aufbau eines HTML-Dokumentes .....	2
Festlegen von Farben .....	3
Einbinden externer Dateien .....	3
Einfügen von Bildern .....	3
Verlinken von Dokumenten .....	4
Textstrukturierung.....	4
Tabellen .....	5
Entwicklung eines Projektes - Gestaltung.....	7
Frames .....	7
Stylesheets .....	8
Bereiche (<div>).....	9
Formulare .....	10

## Vorbemerkung

Hypertext Markup Language ist die Sprache des WWW. HTML ist eine Auszeichnungssprache zur Strukturierung und Gestaltung von Webseiten. Hypertext bedeutet, dass die Textstruktur nicht linear ist, wie vom klassischen Medium Text gewohnt, sondern dass der Gesamttext in eine logische, untereinander verbundene, Teile untergliedert ist. Der Leser kann also selbst über die Reihenfolge des Lesens entscheiden.

Außerdem besteht bei Hypertext in besonderem Maße die Möglichkeit, Text durch Multimedia-Inhalte zu veranschaulichen oder zu ergänzen.

HTML ist eine Software-unabhängige Sprache, d.h. sie kann mit einem einfachen Texteditor erzeugt werden. HTML-Dokumente werden als einfache ASCII-Dateien gespeichert und vom Browser interpretiert.

## Grundbegriffe

**WWW:** Weltweites Hypertextsystem, basierend auf einem standardisierten Protokoll (http) zur Anzeige HTML-codierter Informationen.

**Web-Server:** Mit dem Internet verbundener Computer, auf dem eine Server-Software zur Verteilung von HTML-Seiten arbeitet.

**Web-Seite:** Jede HTML-codierte Seite, die mit einem Internet-Browser angezeigt werden kann.

**Web-Site:** Gesamtheit aller Web-Seiten eines Internetangebotes.

**Homepage:** Startseite; erste Web-Seite einer Web-Site. (meist index.html)

## Aufbau eines HTML-Dokumentes

Die Auszeichnungsmarken zur Textstrukturierung heißen Tags.

```
<tag> Inhalt </tag>
<title> Meine Homepage </title> → Titel der Seite
<p> Das ist ein Absatz ...</p> → Absatz
```

Tags kennzeichnen Anfang (öffnender Tag) und Ende (schließender Tag) einer Textauszeichnung.

Einige Tags stehen auch für sich allein.

```
 → Zeilenumbruch
```

Tags können zur weiteren Beschreibung Attribute enthalten

```
<tag attribut="Wert"> Inhalt </tag>
 Tabelle → Tabelle mit der Breite 400 Pixel
</table>
```

Jedes HTML-Dokument hat die gleiche Grundstruktur

```
<!doctype html public "-//W3C//DTD HTML 4.01 Transitional//EN"> [A]
<html> [B]
<head> [C]
<title> Beispiel</title>
<meta name="author" content="Carsten">
<meta name="generator" content="Ulli Meybohms HTML EDITOR">
</head> [D]
<body text="#000000" bgcolor="#FFFFFF" link="#FF0000" alink="#FF0000"
vlink="#FF0000"> [E]
<!-- Inhalt der Seite -->
</body> [F]
</html>
```

[A] Doctype-Definition: Welche Auszeichnungssprache und welche Version wird verwendet

[B] Beginn des HTML-Quellcodes

[C] Head: Informationen über das Dokument, insbesondere für Suchmaschinen

[D] Ende des Head-Teils

[E] Body: enthält den Inhalt der Webseite, im Bodytag können Einstellungen z.B. für Text und Hintergrundfarbe vorgenommen werden

[F] Body- und HTML-Tag werden wieder geschlossen

## Festlegen von Farben

Farben können z.B. im text-Attribut des Body-Tags festgelegt werden. Dazu kann man vordefinierte Farbkonstanten verwenden, wie *blue*, *lightblue*, *darkblue*, aber auch so vielsagende wie *thistle* oder *plum* (beides Violettöne). Genauer kann man Farben mit Hilfe des RGB-Modells einstellen. Für jede Grundfarbe stehen 8 Bit zur Verfügung, d.h. 256 Farbtöne bzw. die Werte von 0 bis 255 (volle Intensität). Die Angabe erfolgt im Hexadezimalsystem mit den Ziffern 0,1,2 ..., 9, A(10), B(11), C(12), D(13), E(14), F(15)

Die Hexadezimalzahl 2E bedeutet also:  $2E = 2 \cdot 16 + 14 = 46$ , FF hat den Wert  $FF = 15 \cdot 16 + 15 = 255$

1. Berechne den Dezimalwert der Hex-Zahlen E1, AB, 23, 0A, 4B

2. Schreibe die Zahlen 11, 45, 111, 154, 200 als Hexadezimalzahlen auf

Schwarz bedeutet im RGB-Modell, dass alle drei Grundfarben den Wert 0 haben, für Weiß sind alle Wert maximal also FF, für Rot wählt man den roten Anteil maximal, die beiden anderen Null, d.h.

schwarz → #000000 (# Kennzeichnung einer Hex-Zahl, je zwei Stellen pro Farbe)

weiss → #FFFFFF

rot → #FF0000; blau → #0000FF; gelb → #FFFF00

Für weiße Schrift auf blauem Untergrund schreibt man

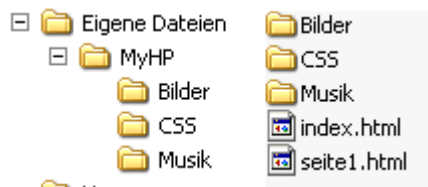
<body text="white" bgcolor="blue"> ... oder

<body text="#FFFFFF" bgcolor="#0000FF"> ...

## Einbinden externer Dateien

Egal, ob man Bilder einfügen, auf anderen Webseiten verweisen oder Stilvorlagen verwenden will, in jedem Fall muss man eine externe Datei angeben (referenzieren). Dazu muss sowohl der Dateiname als auch der Pfad zur Datei bekannt sein. Da Webseiten i.A. auf Webservern abgelegt (gehostet) werden, d.h. auf einem andern Rechner, als der, auf dem sie erstellt wurden, liegen die Dateien auf anderen Pfaden. Beim Referenzieren müssen also relative Pfade verwendet werden, d.h. ausgehend von der referenzierenden Datei.

Bsp: Klaus hat sich für seine Homepage in Eigene Dateien (Laufwerk N:) folgende Ordner angelegt:



Im Ordner *Bilder* gibt es eine Datei *Bild1.jpg*. Der absolute Pfad dieser Datei ist:

N:\MyHP\Bilder\Bild1.jpg.

Soll dieses Bild im Dokument *index.html* angezeigt werden, ist der relative Pfad Bilder\Bild1.jpg.

Man benötigt also in diesem Dokument die Zeile:

```

```

Der Ordner *Musik* enthält die Datei *seite2.html*. Soll von dieser Seite Bild1 referenziert werden, so schreibt man

```

```

## Einfügen von Bildern

Der im vorigen Abschnitt erklärte <img>-Tag enthält einige weitere wichtige Attribute.

```

```

Die Parameter *width* und *height* geben die Dimension des Bildes in Pixeln an. Die Angabe ist besonders dann wichtig, wenn es Übertragungsfehlern kommt und das Bild nicht dargestellt werden kann. Ohne Angabe dieser Parameter kann durch das fehlende Bild der gesamte Aufbau der Seite zerstört werden.

Der Parameter *alt* ist ebenfalls dann wichtig, wenn das Bild nicht angezeigt wird. Der Alternativtext soll dem Nutzer eine Vorstellung vom fehlenden Inhalt vermitteln.

## Verlinken von Dokumenten

Eine der wesentlichen Eigenschaften von Hypertext ist das Verbinden (Verlinken) von Dokumenten untereinander bzw. von Teilen innerhalb eines Dokumentes. Der grundsätzliche Aufbau eines Links ist:

```
<a href="Ziel">tail</a>
```

Man unterscheidet: externe Links (d.h. auf Seiten außerhalb des eigenen Projekts)

```
<a href="http://www.google.de">Für noch mehr Informationen</a>
```

interne Links (eigene Seiten) [Klaus setzt auf seiner Indexseite einen Link zu *seite1.html* und *seite2.html*]

```
<a href="seite1.html">Über mich</a>
```

```
<a href="Musik/seite2.html">Meine Musik</a>
```

Um Verweise innerhalb eines Dokumentes zu realisieren, muss man zunächst die gewünschte Zielstelle festlegen. Wenn also Klaus in seinem Musikdokument verschiedene Musikstile vorstellt und im Inhaltsverzeichnis einen Link zum Thema Jazz erzeugen will, setzt er am Beginn des Themas Jazz einen „Anker“

```
<a name="jazz">...</a>
```

Im Inhaltsverzeichnis setzt er den Link zum Anker:

```
<a href="#jazz">Jazz</a>
```

Wird dieser Link aktiviert, so scrollt der Browser bis zur gesetzten Marke. Natürlich ist es möglich, ein anderes Dokument mit einem Anker zu verlinken, z.B. von der Indexseite zum Thema Jazz:

```
<a href="Musik/seite2#jazz"> mehr zum Thema Jazz</a>
```

Der Tail eines Links sollte eindeutig und möglichst kurz erklären, was den Benutzer erwartet, wenn er diesem Link folgt.

1. Klaus will von *seite1.html* und *seite2.html* je einen Link zurück zur Startseite *index.html* setzen. Gib den HTML-Code an.
2. Klaus will von jedem Thema seiner Musikseite einen Link zum Seitenbeginn (Inhaltsverzeichnis) setzen. Gib den HTML-Code an!
3. Diskutiere, ob die im Beispiel verwendeten Tails günstig gewählt sind.

## Textstrukturierung

Gegliederte Texte sind übersichtlicher, besser lesbar und verständlicher. Zur Gliederung dienen Absätze und Listen (Nummerierung, Aufzählung) und Überschriften. Nach neueren Festlegungen, die auch schon von einigen Browsern umgesetzt werden, wird Text ohne umschließende Tags nicht mehr angezeigt.

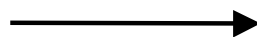
```
<p> dies ist ein Absatz</p>
```

```
<ul>
<li>Gestern</li>
<li>Heute</li>
<li>Morgen</li>
</ul>
```



- Gestern
- Heute
- Morgen

```
<ol start="1" type="1">
<li>Gestern</li>
<li>Heute</li>
<li>Morgen</li>
</ol>
```



1. Gestern
2. Heute
3. Morgen

Aufzählungen können mit den Attributen *start* (Startwert der Aufzählung) und *type* (Art der Aufzählung) formatiert werden. Mögliche Einstellungen des Aufzählungstyps sind:

```
type="1" Ziffern
```

```
type="A" Großbuchstaben
```

```
type="a" Kleinbuchstaben
```

```
type="I" römische Zahlen (I,II,III,IV ...)
```

```
type="i" römische Zahlen (i,ii,iii,iv, ...)
```

Überschriften sind ebenfalls ein wichtiges Mittel zur Textgestaltung und Strukturierung, sie sollen den Inhalt und den Aufbau des Textes verdeutlichen. HTML stellt dazu Überschriften 1. bis 6.

Ordnung bereit. Ordnung bedeutet hier Gliederungsebene. Etwa

```
<h1>Musik</h1>
```

```

<h2>Klassik</h2>
<h2>Jazz</h2>
<h3>Bebop</h3>
<h3>Ragtime</h3>
<h4>Scott Joplin</h4>
<h3>Dixieland</h3>
<h2>Pop</h2>

```

Das große Thema ist Musik (Überschrift 1. Ordnung). Teilgebiete der Musik (Klassik, Jazz, Pop) werden in eigenen Kapiteln abgehandelt. (Überschrift 2. Ordnung). Sind die Kapitel so umfangreich, dass sie ebenfalls weiter untergliedert werden, folgt die nächste Gliederungsebene (Jazz→Bebop, Ragtime, Dixieland; Überschrift 3. Ordnung) usw.

1. Warum sind Absätze wichtig?
2. Eine Webseite will Deutschland vorstellen. Entwirf eine Gliederung und gib passende Überschriften mit zugehöriger Ordnung an!
3. Eine Nummerierung soll die Plätze 10-14 deiner persönlichen Lieblingshits aufzählen. Gib den HTML-Quellcode an!

## Tabellen

Um Daten in Tabellen darzustellen braucht man verschiedene Tags.

```

<table> ... </table>   Tabelle anlegen
<tr> ... </tr>        Tabellezeile
<td> ... </td>        Tabellen(Daten)Zelle
<th> ... </th>        Tabellenkopfzelle

```

Eine Tabelle mit 3 Zeilen und 2 Spalten sieht folgendermaßen aus

```

<table border="1">
<tr>
<th>Mond </th>
<th>Sonne </th>
</tr>
<tr>
<td>1738 km </td>
<td> 696 000 km</td>
</tr>
<tr>
<td> -160°C ... 130°C</td>
<td> 6000 K</td>
</tr>
</table>

```

Mond	Sonne
1738 km	696 000 km
-160°C ... 130°C	6000 K

Zellen können innerhalb einer Zeile (colspan) oder einer Spalte(rowspan) zusammengefasst werden

```

<table border="1">
<tr>
<th colspan="2">Himmelskörper </th>
</tr>
<tr>
<th>Mond </th>
<th>Sonne </th>
</tr>
...

```

Himmelskörper	
Mond	Sonne
1738 km	696 000 km

1. Gib den Quellcode für folgende Tabelle an:

Machos		
	Vorteile	Nachteile
	Schön	Doof
Softies		
	Vorteile	Nachteile
	Treu	Doof

## Entwicklung eines Projektes - Gestaltung

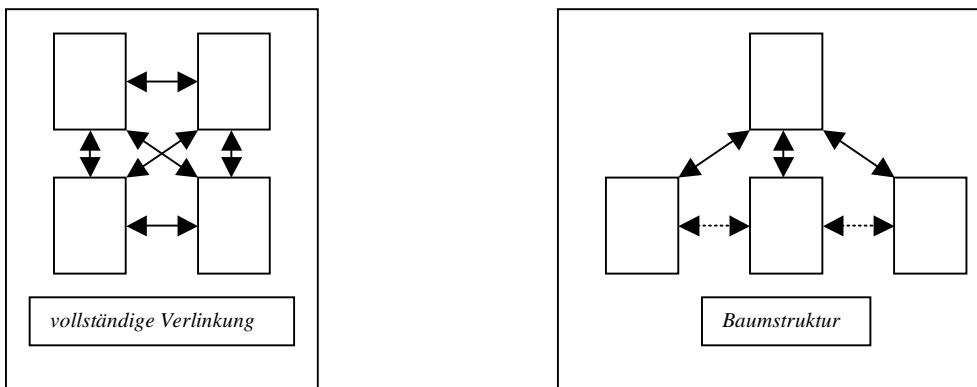
Bei der Entwicklung eines Projektes sollte stets der Inhalt im Vordergrund stehen. Seiten, die nur auf Effekte setzen werden schnell langweilig. Es gelten die Grundsätze:

1. Form ohne Inhalt ist hohl.
2. Inhalt ohne ansprechende Form ist langweilig
3. FFF (form follows function) - Funktion und Inhalt bestimmen die Form
4. Texte müssen gegliedert und durch geeignete Grafiken aufgelockert sein
5. Originelles, funktionelles und einheitliches Design stärkt den Webaufttritt
6. Inhalte müssen gut zu finden und zu erreichen sein (Navigation)

Es ist sinnvoll, eine Grobstruktur des Designs auf Papier zu entwerfen.

Damit die Einbindung von Grafiken und das Verlinken der Einzelseiten auch nach dem Veröffentlichen des Projektes im Internet funktionieren; damit man den Überblick nicht verliert und für eine bequeme Navigation, ist es ratsam eine geeignete Ordnerstruktur zu entwickeln, die dann komplett auf den Webserver übertragen wird. Für kleine Projekte ist es sicher ausreichend, je einen Ordner für Bilder, Stylesheets und HTML-Dateien zu benutzen, bei größeren Projekten ist ein Ordner pro Themenschwerpunkt zu empfehlen

Bei kleinen Projekten empfiehlt sich eine vollständige Verlinkung aller Seiten, für große Projekte ist eine Baumstruktur vorzuziehen.



## Frames

Eine weitere Möglichkeit, dem Nutzer eine elegante Navigation zu ermöglichen sind sogenannte Frames (Rahmen). Der Bildschirm wird hierbei in verschiedene Anzeigebereiche aufgeteilt, in denen die verschiedenen Webseiten angezeigt werden. Typisches Beispiel ist die Teilung in einen Hauptframe für die Inhalte und einen Navigationsframe, der ein Inhaltsverzeichnis anzeigt, von dem aus man die Einzelseiten anwählen kann.

Der Einsatz von Frames ist sehr umstritten, da Frame-Seiten nur unvollkommen durch Suchmaschinen-Robots indiziert werden können, fremde Seiten problemlos in das eigene Projekt eingefügt werden können, was (a) ein Urheberrechtsverstoß ist und (b) für Phishing-Attacken missbraucht wird. Bei unsachgemäßem Einsatz der Frame-Technik besteht die Gefahr einer zu starken Aufspaltung des Bildschirms.

Um ein Frame-Projekt umzusetzen benötigt man folgenden Dateien:

Die Dokumente mit dem eigentlichen Inhalt (seite1.html - seite\_n.html); eine Datei mit dem Inhaltsverzeichnis (z.B. inhalt.html) und eine Framesetdatei (frameset.html), in der die Anzahl und die Größe der Frames festgelegt wird.

Der Quellcode des Framesets sieht folgendermaßen aus:

```
<frameset cols="30%,70%">
  <frame src="inhalt.html">
  <frame name="main" src="seite1.html" >
</frameset>
```

Mit dem Attribut cols (Spalten) oder rows (Zeilen) legt man fest, in welcher Ausrichtung der Bildschirm geteilt wird. Im Beispiel werden für die erste Spalte 30 Prozent des Anzeigfensters reserviert, der Rest für den Inhaltsframe „main“

Das Inhaltsverzeichnis könnte in etwa so aussehen:

```
<ol start="1" type="i">
<li><a href="seite1.html" target="main">seite1.html</a></li>
<li><a href="seite2.html" target="main">seite2.html</a></li>
<li><a href="seite3.html" target="main">seite3.html</a></li>
</ol>
```

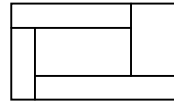
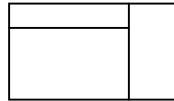
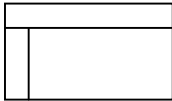
Ohne die Angabe des Attributes target würde die Zielseite des Links im aktuellen Frame angezeigt. Statt für jeden Link ein Target zu setzen kann man auch im Head ein Basisziel für Links festlegen:

```
<base target="main">
```

Davon abweichende Verweisziele müssen dann gesondert festgelegt werden. Insbesondere externe Seiten sollten nicht in einem bestehenden Frameset angezeigt werden, z.B.

```
<a href=http://www.google.de target="_blank">...
  → öffnet ein neues Browser-Fenster
<a href=http://www.google.de target="_top">...
  → zeigt die Seite im gesamtem Browserfenster an
```

1. Finde Beispiele, für sinnvollen Einsatz von Frames. Wann sollte man lieber ohne sie arbeiten?
2. Gib für die folgenden Aufteilungen den Quellcode der Framesets an!



## Stylesheets

Die Möglichkeiten von HTML zur Formatierung des Textes (Schriftart, -größe, Ausrichtung, Textfluss, Einzüge ...) sind begrenzt und teilweise ineffektiv. Mehr Möglichkeiten bietet die Ergänzungssprache CSS (Cascade Style Sheets). Mit CSS ist es möglich, Formateinstellungen zentral zu definieren (d.h. im Head festlegen, wie etwa Überschriften gestaltet werden), anstatt bei jeder Überschrift die gleichen Angaben neu einzugeben. Dadurch werden die HTML-Dokumente kürzer und übersichtlicher. Weiterhin kann man die CSS-Einstellungen in eine externe Datei auslagern und für beliebig viele HTML-Dokumente verwenden.

CSS-Festlegungen haben das Schema

```
Element {Attribut:Wert;Attribut:Wert;...}
```

zum Beispiel

```
h1,h2,h3,h4,h5,h6 {           →Einstellungen für alle Überschriften
  font-size: 20 pt;           →Schriftgröße 20
  font-weight: bold;          →Schriftgewicht Fett
  font-style: italic;         →Schriftstil kursiv
  text-align: center;         →Ausrichtung zentriert
  color: #FFFF00;             →Schriftfarbe Gelb
  background: #000080;        →Hintergrund der Überschrift dunkelblau
}
h2 {font-size: 18 pt;}       →Einstellungen nur für h2
h3 {font-size: 16 pt;}
```

Einstellungen für ein einzelnes Element erfolgen im Tag mit Hilfe des style-Attributs, z.B. um einen Absatz vom restlichen Text abzuheben:

```
<p style="text-align: justify; text-style: italic; ..."> ...</p>
```

Zentrale Festlegungen erfolgen im Head

```
<head>
...
<style type="text/css">
<!--
p {
text-align: left;
...

```



```

}
h1, h2, h3 {
...
}
-->
</style>
</head>

```

Der Code zum Referenzieren einer externen CSS-Datei lautet

```
<link rel=stylesheet type="text/css" href="CSS/allgemein.css">
```

Farben werden durch Angaben von Konstanten (blue) oder durch ihre hexadezimalen Werte angegeben. Texteneinstellung werden über font-family(Schriftart); font-size oder font-style, aber auch mit text-decoration (z.B. underlined), text-align (Ausrichtung) oder text-indent (Einzug) vorgenommen.

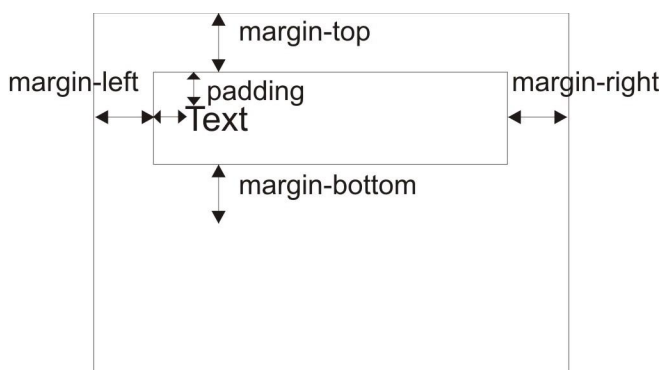
Für Rahmeneinstellungen verwendet man das Attribut border, z.B.

```

table {border-color: #ABCDEF; →Rahmenfarbe
       border-style: solid;   →Rahmenstil (durchgezogen)
       border-width: 5px}    →Rahmendicke

```

Ränder werden über margin und padding festgelegt. Dabei ist margin der so genannte



Aussenabstand, der den Abstand des aktuellen Elementes vom Elternelement (i.A. body) beschreibt. Im Gegensatz dazu ist padding der Innenabstand, also der Abstand von Elementgrenze zum Inhalt.

Sowohl die Rahmen als auch die Abstandseinstellung lassen sich mit den Parametern -top, -right, -bottom, -left feinjustieren.

1. Welche Vorteile hat die Verwendung von CSS?
2. Gib die CSS-Einstellungen für eine weiße Überschrift mit blauer Unterlegung in Fettdruck, Schriftgröße 16, Times New Roman, Rahmen links und rechts mit 20 Pixeln, Innenabstand 20, Außenabstand 10

### **Bereiche (<div>)**

Oft ist es wünschenswert, die normale Abfolge der Text- und Grafikelemente aufzubrechen. Durch die Festlegung von Bereichen können mehrspaltige oder zeitungähnliche Layouts realisiert werden. Bei der Gestaltung dynamischer Seiten spielen Bereiche ebenfalls eine große Rolle. Die Einstellungen werden per CSS festgelegt. Im Beispiel wird mit zwei Bereichen gearbeitet, die zwei Textspalten festlegen

```

<head>
<style type="text/css">
<!--
div {                               → Einstellungen für beide Bereiche
font-family: Arial;
text-align: left;
position: absolute;
padding: 10px;
top: 0px;
}
div#left {                           → Einstellungen für linken Bereich
width:200px;
left:0;
color:#EAE2CC;
}
#main {                               → Einstellungen für rechten Bereich
width:600px;
left: 200px;

```

```

}
-->
</style>
</head>

```

Im Body werden die Bereiche angegeben und mit Inhalt gefüllt. Angezeigt werden sie nebeneinander. (Vergleiche die Einstellungen von `left`)

```

<div id="left">
Inhalt 1
</div>
<div id="main">
Inhalt 2
</div>

```

1. Drei quadratische Bereiche sollen

a) nebeneinander                      b) untereinander                      c) diagonal versetzt  
angeordnet werden. Gib den HTML-Quellcode an.

d) Alle drei Bereiche sollen unterschiedliche Farbeinstellungen haben. Gib die CSS-Anweisungen an!

## Formulare

Formulare dienen zur Dateneingabe, etwa um Bestelldaten abzuschicken oder Suchabfragen zu formulieren. Sie können Textfelder, Optionslisten, Checkboxen und Buttons enthalten. Ein typisches Bestellformular könnte so aussehen:

```

<form action="mailto: service@abc.com" method="get">
Name:<input type="Text" name="Kunde">
Adresse:<input type="Text" name="Adresse"> <br>
<select name="Artikel" size="1">
<option value="1">VirusBlokker PE</option>
<option value="2">VirusBlokker Enterprise</option>
<option value="3">VirusBlokker Professional</option>
</select> <br>
Zahlungsart:<input type="Radio" name="zahlung" value="1">Rechnung
<input type="Radio" name="zahlung" value="2">Vorauszahlung
<input type="Radio" name="zahlung" value="3">Kreditkarte <br>
<input type="Submit" name="ok" value="OK">
<input type="reset">
</form>

```

The screenshot shows a rendered version of the HTML form above. It includes text input fields for 'Name' and 'Adresse', a dropdown menu for 'Artikel' with 'VirusBlokker PE' selected, radio buttons for 'Zahlungsart' (Rechnung, Vorauszahlung, Kreditkarte), and two buttons: 'OK' and 'Zurücksetzen'.

Der `form`-Tag definiert das Formular, für eine automatisierte Auswertung der Formulare Daten ist es oft günstig, ein Formular mit dem Attribut `id` eindeutig zu benennen. Besonders wichtig ist das Attribut `action`, es legt fest, wie die Daten verarbeitet werden. Möglich sind die Angabe einer Mailadresse zum Verschicken der Daten per E-Mail, einer HTML-Seite, die Daten per JavaScript o.ä. auswertet und die Ergebnisse sofort anzeigt oder eines serverseitigen CGI-Scriptes.

Das Attribut `method` legt die Übertragungsmethode fest. Möglich sind die Einstellung `get` und `post`. Im ersten Fall werden die Daten als Parameter an die URL der aufgerufenen Webseite angehängt. JavaScript hat nur Zugriff auf Daten, die mit `get` versandt wurden.

`post` verschickt die Daten über den Standardeingabekanal an den Webserver. Mit dieser Methode können mehr Daten verschickt werden, sind Datenbankoperationen und umfangreichere Auswertungen möglich. Zur Auswertung ist aber ein CGI-Script notwendig.

Eingabefelder werden mit `input` realisiert, `type` entscheidet dann über die Art der Eingabe (`text` → einzelzelliges Textfeld, `radio` → Radiobutton). Mehrere Radiobuttons können den gleichen Namen enthalten, sie bilden dann eine Gruppe, von der nur eine Option wählbar ist. Welche Option

gewählt wurde, ist über `value` abrufbar. Bei Textfeldern ist `value` der vom Nutzer eingetragene Inhalt.

Listen mit vordefinierten Einträgen, aus denen der Nutzer auswählen kann, erhält man mit der Kombination `select/option`.

Um die eingestellte Aktion auszulösen, d.h., die Daten zu versenden, benötigt man ein `Submit-Button`. Das Attribut `value` dient hier zusätzlich als Beschriftung des Buttons. `Reset-Buttons` löschen sämtliche Eingaben im Formular.