

MOTIVATION

A) Öffne die Vorlage zu "Chat oder E-Mail" und untersuche, welche Tags und wahrscheinlich warum.

B) Alle **fett** markierten Textteile sollen nun rot und alle *kursiv* markierten Teile blau gefärbt werden. Welche Vorgehensweise aus Sicht eines HTML-Entwicklers wäre sinnvoll.

Chat oder E-Mail - Ein Vergleich

blau

rot

E-Mail und **Chat** sind beides sehr praktische Programme im Internet, um sich mit verschiedenen Menschen Nachrichten auszutauschen.

Eine **E-Mail** ist vergleichbar zu einem *Brief aus Papier*. Die **E-Mail** enthält genau wie ein *Brief* einen Sender, einen Empfänger, einen Betreff und eine Nachricht. Eine **E-Mail** wird an ein Postfach im Internet geschickt und dort gespeichert. Der Empfänger holt dann diese **E-Mail** ab, indem er seinen Rechner, seinen Laptop, sein Smartphone oder sein Tablet über das Internet mit seinem Postfach verbindet und die **E-Mail** aus dem Postfach holt.

Ein **Chat** ist so etwas wie ein *Raum* im Internet, in dem sich mehrere Leute treffen. Solch einen *Raum* bezeichnet man als "*Chat-Room*", das heißt auf deutsch "Gesprächsraum". Im Gegensatz zu einer E-Mail gibt es kein Postfach, in dem die Nachricht gespeichert wird. Sie wird direkt im "*Chat-Room*" angezeigt.

E-Mails werden oft benutzt, wenn man mit öffentlichen Einrichtungen wie zum Beispiel Schulen, Behörden, Polizei oder Ordnungsamt kommunizieren möchte.

C) Formatiere mit CSS "Schriftart Arial". Finde einen geeigneten TAG und benutze:
`font-family: Arial;`

D) Führe weitere geeignete Ergänzungen durch.

ÜBERSICHT

HTML → *Inhalt* (mit einigen wenigen von Browsern interpretierbaren Auszeichnungsmöglichkeiten wie z.B. `<b>` oder `<i>`)

CSS → *Darstellung* (wie und wo Elemente im Browser angezeigt werden sollen)

Formatierungszuweisung im `<head>`

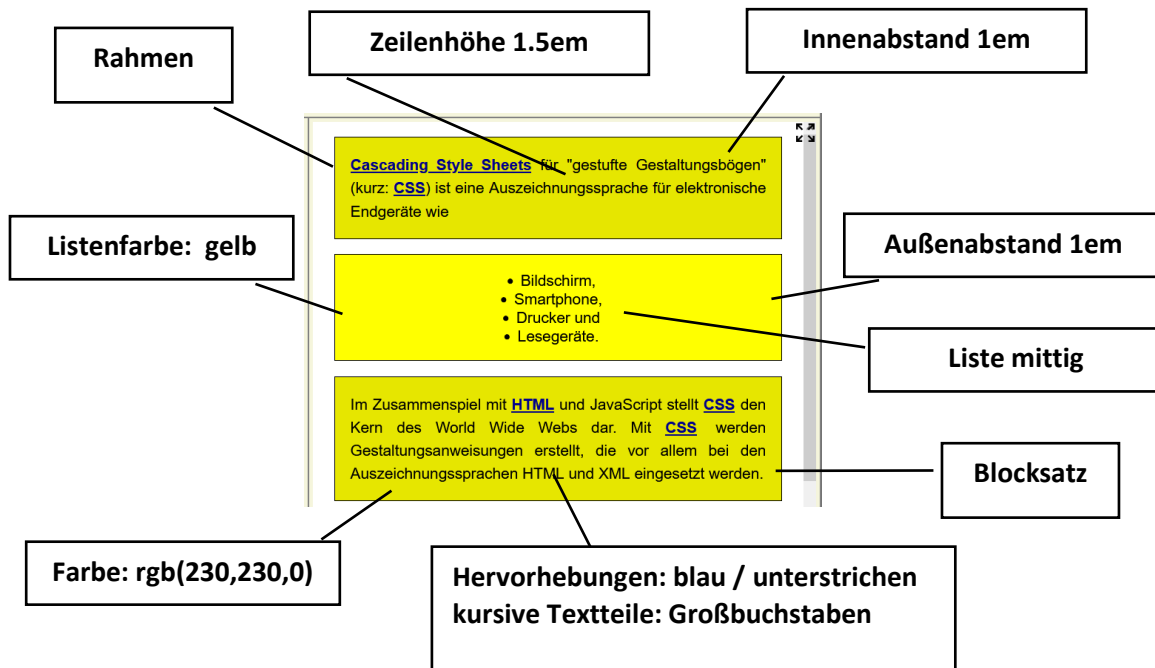
```
<style>
  b {
    font-family: Arial, Helvetica, sans-serif;
  }
</style>
```

Übersicht über eine Auswahl von **CCS-Formatierungen** im **Anhang**.

A U F G A B E N

Aufgabe 1

- a) Lade die Vorlage "Cascading Style Sheets" und formatiere die Listenelemente (Bildschirm bis Lesegerät) als Liste.
b) Formatiere den Text mit CSS wie in der Vorgabe.



Aufgabe 2

- a) Lade die Vorlage "CSS als living standard" und formatiere die Liste mit kleinen Buchstaben (a bis d).
b) Formatiere den Text mit CSS:
- Überschrift groß, fett und orange.
 - alle Wörter mit "Darstellung" fett, braun und unterstrichen
 - Blocksatz
 - Außenabstand von 2 em.
 - Den Abschnitt "CSS ist ein living standard ..." gelb hervorheben.

Aufgabe 3

- a) Öffne "iPhone - Entwicklung und Übersicht" und formatiere mit CSS:
Schrift Arial oder serifenlos, Zeilenabstand 2em, Absätze, gestrichelter Rahmen mit Innen- und Außenabstand 1em
b) Das Menü als sortierte Liste
c) Setze von jedem Menüpunkt eine Sprungmarke zum entsprechenden Absatz und ergänze Rücksprünge

Sprung / Menüpunkt: `<a href="#Absatz1"> Menüpunkt 1 </a>`

Anker / Sprungziel: `<a name = "Absatz1"> </a>`

- c) Formatiere die Links (Menüpunkte) in grau mit orangem Hintergrund, Außenabstand 0.5em und ohne Unterstreichung. Beim Drüberfahren soll der Hintergrund grün werden.

	CSS Grundlagen
	Layout zusammenfassen

Übersicht über eine Auswahl von **CCS-Formatierungen**

1.) Textformatierung	
font-size: 12px; font-size: 1.5em; (1.5em ohne Leerzeichen)	Schriftgröße: mögliche Werte: pt, px, em pt: Point - absolute Schriftgröße px: Pixel - absolute Schriftgröße em: Breite des Buchstabens M der Standardschrift
font-family: Arial; font-family: Times, serif;	Schriftart: Monospace, Fantasy, sans-serif (Schrift ohne Serifen), serif (Schrift mit Serifen), monospace
font-weight: bold;	Breite der Schrift: normal, bold, bolder, lighter
font-style: italic;	Auszeichnung: italic, oblique, normal
line-height: 1.5em;	Zeilenhöhe: Bei mehrzeiligem Text sind relative Angaben über die Einheit „em“ empfehlenswert.
text-decoration: underline;	Textgestaltung: underline, overline, line-through, blink
text-indent: 2em;	Einrückung: erste Zeile eines Textblocks
text-align: right;	Textposition: center, left, right
text-transform: uppercase;	Umgestaltung der Buchstaben: uppercase, lowercase, capitalize, none

2.) Farben	
color: tomato; color: rgb(120,120,120); color: #FF10AF	Farbe für Schrift: - Vorgabe: red, blue, green, lime, maroon, yellow, lightgray, etc. - RGB-Dezimal: rgb(0,255,10) - RGB-Hexadezimal: #00FF0A
background-color: brown;	Hintergrundfarbe: Farbe für den Hintergrund

3.) Rahmen	
border-width: 1px; border-top-width: 2px;	Rahmenstärke: in Pixel, soll nur ein Bereich geändert werden: border-top-width, border-right-width, border-bottom-width und border-left-width
border-style: solid; border-top-style: double;	Rahmenart: solid, double, none, hidden, dotted, dashed, groove, ridge, inset - Soll nur ein Teil des Rahmens definiert werden, dann: border-top-style, border-right-style, border-bottom-style und border-left-style
border-color: black; border-right-color:	Rahmenfarbe: Soll nur ein Teil des Rahmens gefärbt werden, dann: border-top-color, border-right-color, border-bottom-color und border-left-color.

	CSS Grundlagen
	Layout zusammenfassen

<code>rgb(0,255,0);</code>	
<code>border: 2px solid Tomato;</code> <code>border: 2pt dotted</code> <code>rgb(255,0,10);</code>	Zusammenfassung der drei Attribute: Rahmenstärke, -art, -farbe in einer Zeile.
<code>border-style: dotted</code> <code>solid solid dotted;</code>	Zusammenfassung der vier Himmelsrichtungen: in oben, rechts, unten und links einer Zeile.

4.) Ausrichtung von Elementen	
<code>text-align: center;</code>	Horizontale Ausrichtung: left, center, right, justify (Seitenbreite)
<code>vertical-align: top;</code>	Vertikale Ausrichtung: text-bottom, top, middle, bottom, baseline, sub (tiefer), sup (höher), text-top

5.) Rahmenabstände beim Boxmodell	
<code>margin: 2em;</code>	Abstand zum nächsten Element: kann auch für jede Seite einzeln angegeben werden: margin-left, margin-right, margin-bottom, margin-top
<code>padding: 3px;</code>	Abstand zum Rand: kann auch für jede Seite einzeln angegeben werden: padding-left, padding-right, padding-bottom, padding-top

6.) Links/Referenzen (Pseudoklassen)	
<code>a:link {</code> <code>color: blue;</code> <code>text-decoration: none;</code> <code>}</code>	Auf die Reihenfolge achten (Merkregel dazu: LoVe For HAtE) a:link, a:visited, a:focus, a:hover, a:active Kein Leerzeichen beim ":"

7.) Listenelemente	
<code>list-style: none;</code> <code>list-style: upper-roman;</code>	beim Tag → Listenelemente: none, upper-roman, lower-roman, decimal
<code>display: block;</code> <code>display: inline;</code>	beim Tag → Listenelemente werden als Block untereinander und als inline-Elemente nebeneinander angeordnet