

# Moderne Webtechnologien I (WT 1)

## Übung

---

WS 2022/23

Prof. Dr. C. Köhn

Daniela Böing

19. Oktober 2022

**Website:**

<https://www.hochschule-bochum.de/fbe/fachgebiete/institut-fuer-informatik/labor-fuer-medienkommunikation-internet-und-robotik/>

Hochschule Bochum  
Bochum University  
of Applied Sciences



# Inhaltsverzeichnis

<b>1</b>	<b>CSS</b>	<b>2</b>
1.1	CSS-Syntax . . . . .	2
1.2	CSS im HTML-Dokument anwenden . . . . .	2
1.3	CSS-Beispiele . . . . .	3
1.3.1	Farbangaben im Browser . . . . .	3
1.3.2	Außenabstand ( <code>margin</code> ) eines Elementes . . . . .	3
1.3.3	Innenabstand ( <code>padding</code> ) eines Elementes . . . . .	3
1.3.4	Schriftformatierung . . . . .	4
1.3.5	Textformatierung . . . . .	5
1.4	Webseitenprojekt erweitern . . . . .	5
1.4.1	Übergreifende Formatierungen . . . . .	5
1.4.2	Header-Formatierungen . . . . .	6
1.4.3	Formatierungen des Hauptinhalts . . . . .	7
1.5	Ergebnis . . . . .	7

# 1 CSS

## 1.1 CSS-Syntax

Cascading Style Sheets (kurz: CSS) gehört zu der Gruppe der Beschreibungssprachen und ist mittlerweile ein Standard zur Formatierung von HTML-Inhalten geworden.

Die CSS-Syntax besteht aus einfachen Regeln, den Zuweisungen zu verschiedenen Deklarationsbereichen. Diese wiederum bestehen aus einer oder mehrerer Deklarationen, die sich aus Eigenschaft (property) und Wert (value) zusammensetzen, zu Tags (selector), die zum Beispiel HTML-Elemente sein können. In den geschweiften Klammern können sich beliebig viele Anweisungen befinden, die mit einem Semikolon voneinander getrennt werden.

```
/* Kommentar */
Tag {
    Eigenschaft: Wert;
    Eigenschaft: Wert;
}
Tag.Klasse {
    Eigenschaft: Wert;
}
Tag#ID {
    Eigenschaft: Wert;
}
```

### Hinweis:

Wenn es mehrere Deklarationen für dasselbe Element gibt, gilt folgende Rangordnung: Die höchste Priorität haben die Deklarationen für eine ID (mit `Tag#ID`), dann folgen die Klassendeklarationen (mit `Tag.Klasse`), dann die für das Element (so spezifisch wie möglich). Für mehrere Deklarationen derselben Rangordnung überschreibt das zuletzt deklarierte die vorherige(n).

## 1.2 CSS im HTML-Dokument anwenden

Die Integration in ein HTML-Dokument kann auf drei verschiedene Wege erfolgen, wobei hier nur die empfohlene Variante gezeigt werden soll: die externe Einbindung mittels `link`-Tag. Diese Vorgehensweise ermöglicht auch die Wiederverwendung von Stylesheets, die Pfadangabe erfolgt relativ vom aktuellen Verzeichnis aus.

```
1 <html>
  <head>
    <link rel="stylesheet" type="text/css" href="style/style.css">
    ...
```

**Übung 1:**

Laden Sie sich die Vorgabeseite herunter (eine Erweiterung zur Musterlösung der letzten Übung).

Es existiert bereits eine CSS-Datei *style.css* in einem Unterordner *styles* mit einigen CSS-Formatierungen. Binden Sie diese Datei wie gezeigt in der *index.html* und *game.html* im Head-Bereich der Seite ein. Diese Datei sollen Sie im Laufe der Übung erweitern.

## 1.3 CSS-Beispiele

### 1.3.1 Farbangaben im Browser

Die Farbangaben für die Attribute erfolgen im Hexcode, wobei es einige vordefinierte Farben gibt, die direkt genutzt werden können. Hier ein kleines Beispiel:

```
1 body {  
  color: white; /* Schriftfarbe wird auf Weiß gesetzt; äquivalent zu: color: #  
    ffffff; */  
  background-color: #ffff00; /* Hintergrundfarbe ist auf Gelb gesetzt */  
}
```

### 1.3.2 Außenabstand (margin) eines Elementes

Ein Element hat vier Seiten: rechts, links, oben und unten (right, left, top und bottom). Der Außenabstand *margin* ist die Entfernung jeder Seite zum Nachbarelement (oder den Rändern des Web-Dokuments). Beispiel:

```
1 body {  
  margin-top: 100px;  
  margin-right: 40px;  
  margin-bottom: 10px;  
5  margin-left: 70px;  
}
```

Sie können aber auch diese elegantere Version wählen:

```
1 body {  
  margin: 100px 40px 10px 70px;  
}
```

### 1.3.3 Innenabstand (padding) eines Elementes

Der Innenabstand (*padding*) kann auch als „Füllung“ verstanden werden. Dies macht Sinn, da der Abstand zu anderen Elementen nicht verändert wird, sondern nur der innere Abstand vom Rand (*border*) zum Inhalt des Elementes. Beispiel:

```
1 h1 {  
  background: yellow;  
  padding: 20px 20px 20px 80px;  
}
```

### 1.3.4 Schriftformatierung

Um die Schrift zu formatieren, gibt es verschiedene `font`-Eigenschaften, die auch über die Eigenschaft `font` alle auf einmal zugewiesen werden können:

```
[font-style] | [font-variant] | [font-weight] | [font-size] |
[font-family]
```

Ein Auszug aus der Bedeutung der einzelnen Parts:

- `font-style`: Hierüber kann zum Beispiel angegeben werden, dass der Text kursiv (*italic*) angezeigt werden soll.
- `font-weight`: Dies kann zur Angabe der Schriftstärke angegeben werden (Angabe wie bei der Schriftgröße).
- `font-size`: Hiermit wird die Schriftgröße angegeben, s. Beispiel unten.

Hier ein Beispiel zur Angabe der Schriftgröße:

```
1 h3 {
  font-size: 120%;
}
```

Es gibt einen grundlegenden Unterschied zwischen den vier obigen Einheiten. '**px**' und '**pt**' bringen die Schrift auf eine absolute Größe, während es '**%**' und '**em**' dem Nutzer erlaubt, die Größe so zu verändern. Um Ihre Webseite barrierefrei zu halten, sollten Sie veränderbare Einheiten, wie '**%**' oder '**em**', verwenden.

Das Attribut `font-family` soll einmal gesondert betrachtet werden. Hier wird eine priorisierende Liste der Zeichensätze angegeben, in der ein bestimmtes Element oder eine Webseite angezeigt werden soll. Wenn der erste Zeichensatz nicht verfügbar ist (zum Beispiel nicht auf dem Computer installiert), wird der nächstfolgende probiert, bis ein vorhandener bzw. passender Zeichensatz gefunden ist.

Es gibt zwei Arten die Zeichensätze zu kategorisieren: Schriftart und generische (gattungsmäßige) Familie.

Times New Roman  
Garamond  
Georgia

Diese drei Schriftarten gehören zur generischen Familie **serif**. Diese sind an ihren "Füßchen" erkennbar.

Trebuchet  
Arial  
Verdana

Diese drei Schriftarten gehören zur generischen Familie **sans-serif**. Diese erkennt man daran, dass sie keine "Füßchen" haben.

Courier  
Courier New  
Andale Mono

Diese drei Schriftarten gehören zur generischen Familie **monospace**. Sie haben alle eine feste Breite.

Beispiel:

```
1 h2 {  
  font-family: "Times New Roman", serif;  
}
```

### 1.3.5 Textformatierung

Texte zu formatieren ist eine der Hauptaufgaben für jeden Webdesigner. Hier ein Auszug an interessanten Eigenschaften:

- `text-align`: bestimmt die Ausrichtung des Textes (zum Beispiel `left` oder `justify`)
- `text-decoration`: legt zusätzliche Formatierungen fest (zum Beispiel `underline`)

## 1.4 Webseitenprojekt erweitern

### 1.4.1 Übergreifende Formatierungen

Für die folgenden Übungen können Sie gerne das Internet als Hilfe nutzen. Gute Nachschlageseiten sind zum Beispiel <https://www.w3schools.com/css/> oder <https://wiki.selfhtml.org/wiki/CSS>.

#### Übung 2:

Formatieren Sie die gesamte Seite (Tags `html`, `body`) wie folgt:

1. Setzen Sie die Höhe auf 100 %.
2. Setzen Sie den Außenabstand auf 0.
3. Der Hintergrund soll schwarz sein.
4. Die Schriftfarbe soll weiß sein.
5. Die Texte sollen im Blocksatz angezeigt werden.

Alle vorher beschriebenen Eigenschaften können auch auf Links angewendet werden (Farben verändern, Schriftarten, Unterstreichungen usw.). Es ist auch möglich, zu kennzeichnen, ob der Link schon besucht (`visited`) oder noch nicht besucht (`link`) wurde, gerade angeklickt wird (`active`) oder nur der Mauszeiger über dem Link platziert ist (`hover`). Um diese Effekte zu steuern, benutzt man sog. Pseudoklassen (`pseudo-classes`).

Eine Pseudoklasse erlaubt es Ihnen, zwischen verschiedenen Zuständen oder Ereignissen zu unterscheiden, wenn Sie eine Eigenschaft für einen HTML-Tag festlegen. Ein Beispiel:

```
1 a:hover{  
  color: blue;  
}
```

### Übung 3:

Nun sollen die Links formatiert werden. Der Text soll standardmäßig in einer gewählten Highlight-Farbe, die noch mehrfach verwendet wird, angezeigt werden (im Beispiel #C2FFD7) und nicht unterstrichen werden. Beim Mouseover über den Link soll sich die Textfarbe in weiß verwandeln.

## 1.4.2 Header-Formatierungen

Um Blöcke, zum Beispiel Container, automatisch zentrieren zu lassen, können die Eigenschaften `margin-left` und `margin-right` auf `auto` gesetzt werden.

### Übung 4:

Nun sollen die Überschriften formatiert werden:

1. Der umgebende Container `hero-header` soll nach oben einen Innenabstand von 5% besitzen, 40% der Webseitenbreite einnehmen und wie beschrieben zentriert dargestellt werden.
2. Die Überschrift erster Ordnung soll eine zentrierte Textausrichtung haben sowie einen Außenabstand von 0. Außerdem soll die Highlight-Farbe (im Beispiel #C2FFD7) als Textfarbe benutzt werden und die Schriftgröße 500% betragen.
3. Die Überschrift zweiter Ordnung erhält nur dieselbe zentrierte Textausrichtung.

### Übung 5:

Es folgen Formatierungen des Navigationsbereichs:

1. Der umgebende Container `nav` soll dabei wie folgt aussehen:
  - a) Er soll einen oberen und unteren Innenabstand von 1% und einen rechten und linken Innenabstand von 2% haben.
  - b) Der Container soll 100% der Webseitenbreite einnehmen.
  - c) Die Position soll absolut am unteren Rand der Anzeige festgelegt sein.
2. Das Menü soll ebenfalls einen Außenabstand von 0 haben und soll rechts positioniert werden (Stichwort: `float`).
3. Die einzelnen Listenitems sollen einen rechten und linken Innenabstand von 20 Pixeln erhalten.

**Übung 6:**

Als kleine Erweiterungen sollen nun die Schriftarten festgelegt werden. Da die vorinstallierten Fonts von System zu System variieren und manchmal recht klein im Umfang sind, sollen außerdem Fonts importiert und dann genutzt werden.

1. Setzen Sie die Standard-Schriftart über den `html`- bzw. `body`-Tag zunächst auf „Verdana“ bzw. `sans-serif`.
2. Wählen Sie dann eine Akzentschriftarten zum Beispiel von dieser Seite: <https://fonts.google.com/>. Per Plus können Sie diese auswählen und anschließend rechts unten die nutzbaren `link`-Tags einsehen und in Ihre beiden HTML-Dateien importieren. Im Beispiel wurden die Schriftarten „Rock Salt“ und „Over the Rainbow“ (bereits eingebunden) gewählt.
3. Verwenden Sie dann diese Schriftart für die Elemente `h1` und `brand`.

**1.4.3 Formatierungen des Hauptinhalts**

Umrandungen (`border`) können vielseitig eingesetzt werden. Ein paar Beispiele:

- `border-width`: setzt die Breite eines Rands (in Pixeln oder in Worten, wie `thin`)
- `border-style`: legt die Umrandungsart fest, zum Beispiel `solid` oder `dotted`
- `border-color`: bestimmt die Randfarbe

Diese können ebenfalls unter der Eigenschaft `border` zusammengefasst werden, wie das folgende Beispiel zeigt:

```
1 p {
  border: 1px solid blue;
}
```

**Übung 7:**

Der Container des Hauptinhalts (Klasse `main`) soll einige defaultmäßige Formatierungen enthalten:

1. Es soll ein Innenabstand erzeugt werden, der dadurch auf alle Kindelemente übertragen wird. Rechts und links soll es einen Abstand von 20% geben, nach oben hin von 50 Pixeln.
2. Es soll einen oberen Rand von einem Pixel Breite geben, der in weiß angezeigt wird.

**1.5 Ergebnis**

Dem Beispiel folgend sollte die formatierte Webseite nun wie folgt aussehen:



