





Frieden

Suchbegriff eingeben ...

Suchen

Die Friedensnobelpreisträger:innen

1901

Henry Dunant (1828–1910)
Schweiz



Frédéric Passy (1822–1912)
Frankreich



1902

Élie Ducommun (1833–1906)
Schweiz



Albert Gobat (1843–1914)
Schweiz



1903

William Randal Cremer (1828–1908)
Vereinigtes Königreich



1904

Institut de Droit international
(gegründet 1873)
Sitz in Gent, Belgien



- Aufbau eines HTML-Dokuments
- Konzept: block vs. inline
- Grundlegende Elemente
- Seitenspezifische Elemente für den klassischen Aufbau einer Website:
 - header
 - nav
 - main
 - section
 - footer
- Grundlegende Attribute
 - class / id
 - style



Hypertext Markup Language

- HTML-Dokumente bilden die Grundlage des *World Wide Webs* und werden von Webbrowsern dargestellt
- HTML *strukturiert* elektronische Dokumente (man kann Texte mit Hyperlinks versehen, man kann Bilder einbetten)
- HTML-Dateien können zusätzliche Angaben in Form von Metadaten enthalten (z.B. über die verwendeten Sprachen, über den Inhalt des Textes, eingebettete Schriften)
- Weiterentwickelt und gepflegt vom *W3C* (World Wide Web Consortium) und der *WHATWG* (Web Hypertext Application Technology Working Group)
- vollständige Spezifikation: <https://html.spec.whatwg.org/multipage/>

Aufbau eines HTML-Dokuments

`<!DOCTYPE html>` Ein HTML-Dokument beginnt mit der Kennzeichnung des **Dokumententyps** (kein Element!)

`<html>` Dem folgt das “**Wurzelement**”. Damit identifizieren Browser Anfang und Ende eines HTML-Files

`<head>`

`<title></title>`

`</head>`

`<body></body>`

`</html>`

Darin kommt zunächst der Dokumenten-“**Kopf**”.

Enthält Informationen, die für das ganze Dokument relevant sind (u.a. den **Titel**)

Anschließend folgt der Dokumenten-“**Körper**”. Hier befindet sich der eigentliche Inhalt des Dokuments. Teil, der vom Webbrowser dargestellt wird

Wichtiges Konzept: block vs. inline

Hat Einfluss auf die Darstellung und den Umgang mit Elementen

“Every HTML element has a default display value, depending on what type of element it is. There are two display values: **block and inline**” (w3schools)

- Block-Elemente: container für weitere Elemente, beginnen immer in einer neuen Zeile, produzieren in der Ausgabe einen Absatz und bilden die Struktur eines HTML-Dokuments

`<div>, <p>`

- Inline-Elemente: Beginnen nicht in einer neuen Zeile, sondern stehen mitten im “Text”, produzieren im Output keinen Weißraum (whitespace)

`,
`

- `<p>` (*paragraph*) = **Absätze**
Browser fügen i.d.R. vor und nach einem Absatz eine Leerzeile ein. Will man diese vermeiden, verwendet man stattdessen
- `<div>` = (*division*) gruppierendes Element (**container**)
- `<h1>` bis `<h6>` (*heading*) = **Überschriften**
- `<address>` = führt zu einer Blockdarstellung für **Adressen**
- `<blockquote>` = hebt **Zitate** in einem eigenen, anders formatierten (meist eingerückten) Absatz hervor
- `<pre>` = leitet einen Textabschnitt mit **präformatiertem** Text ein.
Alles, was zwischen `<pre></pre>` steht wird so angezeigt, wie es eingegeben wurde (Leerzeichen und Zeilenumbrüche werden übernommen)

Blockelemente: **Listen**

geordnet und nummeriert. Eine nummerierte Liste wird durch die Tags

`` und `` erzeugt

vs. ungeordnet. Eine ungeordnete Liste wird durch die Tags

`` und `` erzeugt

In beiden Listentypen muss mindestens ein Listenelement vorhanden sein

Zutaten:

- 150g Bulgur
- 300 ml.
- Gemüsebrühe
- 1 Paprikaschote

```
<h3>Zutaten:</h3>
<ul>
  <li>150g Bulgur</li>
  <li>300 ml.</li>
  <li>Gemüsebrühe</li>
  <li>1 Paprikaschote</li>
  ...
</ul>
```


Blockelemente: **Tabellen**

Werden durch die Tags

<code><table></code>	Anfang der Tabelle
<code><caption></code>	Überschrift der Tabelle
<code><th></code>	Kopf der Tabelle
<code><tr></code>	Zeile der Tabelle
<code><td></code>	Eintrag/Zelle der Tabelle

Erzeugt.

Geht noch komplexer!

Nährwerte pro Portion			
kcal	Eiweiß	Fett	Kohlenhydrate
477	19,65 g	15,58 g	63,16 g

```
<table>
<caption>Nährwerte pro Portion</caption>
  <tr>
    <th>kcal</th>
    <th>Eiweiß</th>
    <th>Fett</th>
    <th>Kohlenhydrate</th>
  </tr>
  <tr>
    <td>477</td>
    <td>19,65 g</td>
    <td>15,58 g</td>
    <td>63,16 g</td>
  </tr>
</table>
```

Bisher befindet sich in den Blockelementen nur Text.

Zur weiteren Auszeichnung kommen nun **inline-Elemente** hinzu.

- `` erzeugt eine **fette** Schrift
- `` **gruppierendes Element**, ordnet inline-Elementen Eigenschaften zu. Hat selbst keine Formateigenschaft.
- `
` erzeugt einen **Zeilenumbruch** (eigene Syntax: hat keinen eingebetteten Inhalt. Ist ein leeres Element)

“logische” Auszeichnungselemente

- `` betont die Textausgabe (i.d.R. mit einer *Kursivierung*)
- `<code>` erzeugt eine Teletypeschrift
- `<samp>` “dies ist ein Beispiel”
- `<kbd>` “dies ist eine Tastatureingabe”

Weitere “logische” Auszeichnungselemente

- `<var>` “dies ist eine **Variable**”
- `<cite>` “dies ist ein **Querverweis** zu einer anderen Quelle”
- `<dfn>` “dies ist eine **Begriffsdefinition**”
- `<abbr>` “dies ist eine **Abkürzung**”. Erzeugt in Kombination mit dem Attribut *@title* in der Ausgabe ein Pop-up

(`<abbr title="Pop-up-Text">...</abbr>`)

Weitere wichtige Elemente

Einbau von Bildern (inline!)

Um Bilder und Grafiken in HTML-Dokumente einzubetten, referenzieren Sie die Dateien an den gewünschten Stellen im HTML-Dokument:

```

```

(Ressource aus dem WWW)

```

```

(ein relativer Pfad in Ihrem Dateisystem)

Hinter dem “=” steht der Pfad zur Grafikdatei, auf die verwiesen wird.

Der obige Dateiname als Wert des Attributes `src` verweist auf dieselbe Ressource wie der Link.

Es ist üblich Bilddateien in einer eigenen Directory zu sammeln, z.B. *images* oder *resources*.

Weitere wichtige Elemente

Einbau von Bildern

Für eine eingebundene Grafik sollten

Breite und Höhe immer mit angegeben werden:

```

```

`width="100%"` stellt das Bild in seiner
Originalgröße dar

Weiteres Attribut:

`alt = image/logo` (Screenreader)

Übliche Dateiformate für
WWW-gerechte Grafiken:

- **GIF** (Graphics Interchange Format)
- **PNG** (Portable Network Graphics)
Gutes Einzelbildformat,
Logos und Icons
- **JPEG** (Joint Photographic Experts Group) für Fotos
(verlustbehaftet)
- **SVG** (Scalable Vector Graphics)
Vektorgrafikformat,
XML-basiert, Logos, Icons

Einbau von Hyperlinks (inline!)

Verweise sind ein entscheidender Bestandteil jedes Hypertextprojekts und ein wesentlicher Gewinn des WWW. Für einen Verweis wird das sog. Ankerelement `<a>` mit einem **href**-Attribut verwendet.

`<a>` ist ein Inline-Element. Der Wert des Attributs ist eine URI:

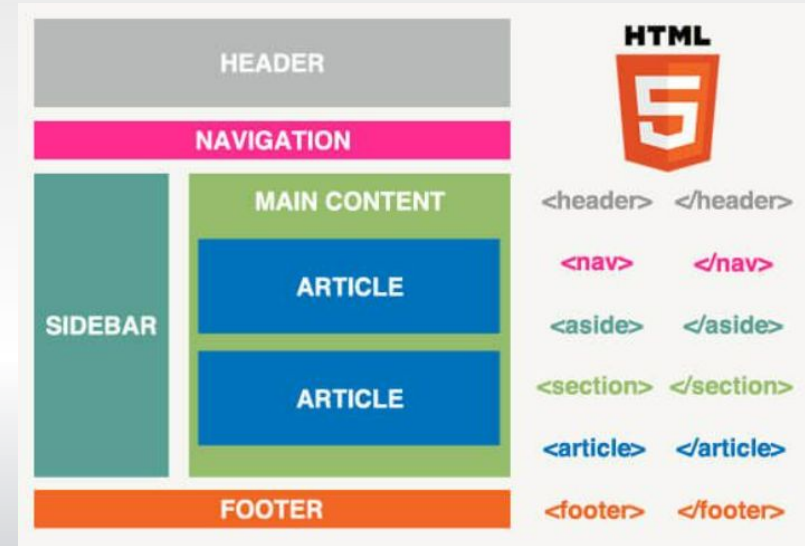
```
<a href="https://www.koeppel-sw.de/wp-content/uploads/2017/03/radkartelsidor_Ripoll.jpg">Radkarte</a>
```

Das Attribut `target=_blank | blank | _self`

- 1) öffnet jedes mal ein neues Fenster
- 2) öffnet einmalig ein neues Fenster, Tab aktualisiert sich, oder
- 3) öffnet den Link im gleichen Fenster

Seitenspezifische Elemente für den klassischen Aufbau einer Website

- header
 - nav
 - main
 - section
 - footer
 - ...
- div
 - img
 - a



<https://www.pinterest.at/pin/520306563200393879/>

Beispiel: Projekte des Zentrums für Informationsmodellierung in Graz:

<http://gams.uni-graz.at/context:gams?locale=de>

Sprachen des Webs: HTML, CSS und JS

Sviat Drach, Pia Geißel, Lena-Luise Stahn, Nadine Sutor



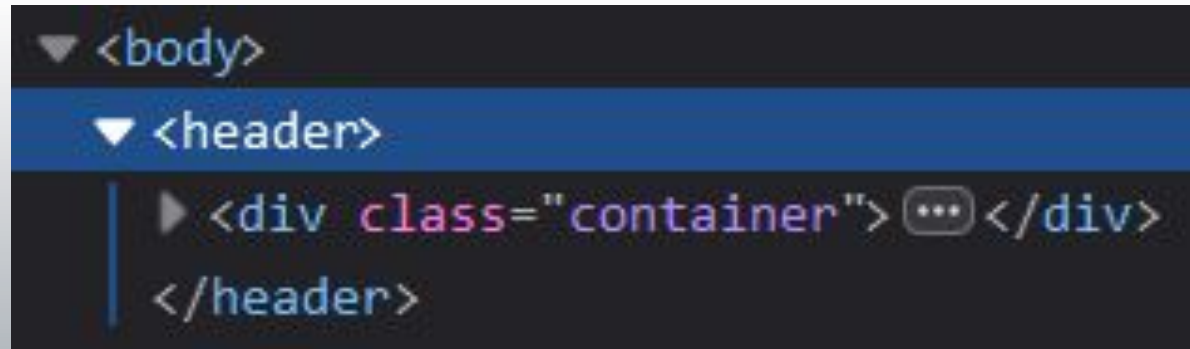
BERGISCHE
UNIVERSITÄT
WUPPERTAL



Seitenspezifische Elemente für den klassischen Aufbau einer Website: <header>

header

- repräsentiert eine Gruppe von Einführungs- oder Navigationshilfen
- auch andere Elemente wie zum Beispiel ein Logo oder ein Suchfeld
- umfasst i.d.R. `div`'s mit einer vergebenen Klasse für weitere (Text)-Container

A screenshot of a code editor with a dark background. It shows a tree view of HTML elements. The root element is <body>, which is expanded to show <header>. The <header> element is further expanded to show a <div class="container"> element, which has a tooltip icon next to it. The </div> and </header> closing tags are also visible.

```
<body>
  <header>
    <div class="container"> ... </div>
  </header>
```


Seitenspezifische Elemente für den klassischen Aufbau einer Website: <nav>

<nav> - (navigation)

- zeichnet einen Seitenbereich aus, der dazu bestimmt ist Navigationslinks bereitzustellen, die entweder zum aktuellen Dokument oder zu anderen Dokumenten zeigen
- häufige Beispiele für Navigationsbereiche sind **(drop-down)-Menüs**, sog. **sidebars** oder Inhaltsverzeichnisse

```
<nav class="menu">
  <ul>
    <li><a href="#">Startseite</a></li>
    <li><a href="#">Über</a></li>
    <li><a href="#">Kontakt</a></li>
  </ul>
</nav>
```

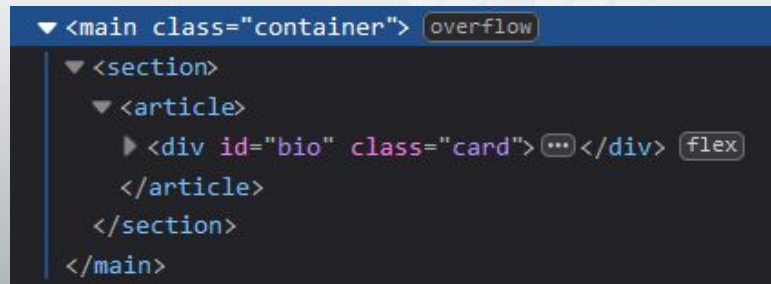
In diesem Beispiel wird das <nav> Element zusammen mit einer ungeordneten Liste () verwendet, die wiederum Links enthalten. Mit CSS kann diese Liste als Seitenleiste, als oben mittig zentrierte Navigationsleiste oder Drop-Down-Menü verwendet werden.

<https://developer.mozilla.org/de/docs/Web/HTML/Element/nav>

Seitenspezifische Elemente für den klassischen Aufbau einer Website: <main>

main

- dient als Container für den HAUPT-Inhalt einer Website
- Hauptteil besteht aus Inhalt, der auf das zentrale Thema oder der zentralen Funktionalität einer Applikation aufbaut
- Ausgenommen sind Inhalte, die in dem Projekt wiederholt dargestellt werden, wie z.B. Seitenleisten, Navigationslinks, Logos, und Sucheingaben
- <main> **darf kein** Nachfolger eines <article>, <aside>, <footer>, <header> oder <nav> Elements sein!
- **Nur ein** <main> Element kann pro Dokument verwendet werden!



```
<main class="container"> overflow
  <section>
    <article>
      <div id="bio" class="card"> ... </div> flex
    </article>
  </section>
</main>
```

Seitenspezifische Elemente für den klassischen Aufbau einer Website: <section>

section – Abschnitte

- stellt eine allgemeine Unterteilung des Dokumentes dar (z.B. eine thematische Gruppierung des Inhaltes, typischerweise mit einer Überschrift)
- jeder <section> Abschnitt sollte identifizierbar sein, üblicherweise durch Einfügen einer Überschrift (h1–h6) als Kindelement vom <section> Element oder durch Vergabe einer Klasse
- Wenn der Inhalt des <section> Elementes “sinnvoll” vom übrigen Inhalt “getrennt” werden soll, nutzt man stattdessen das <article> Element
- <section> soll nicht als allg. Containerelement benutzt werden! → <div>

```
▼ <section>
  ▼ <article>
    ► <div id="bio" class="card"> ... </div> flex
    </article>
  </section>
```

Seitenspezifische Elemente für den klassischen Aufbau einer Website: <footer>

footer – Fußzeile

- repräsentiert eine Fußzeile einer Website
- Eine Fußzeile enthält typischerweise Informationen über die Institutionen, die Copyright-Daten, das Impressum, die Lizenzen etc.

```
<footer class="footer">
  <div class="container">
    <div class="row">
      <div class="col-md-6">
        <h6>Weitere Informationen</h6>
        <p>...</p>
        <div class="icons">
          <a href="/" target="_blank">...</a>
          <a href="https://informationsmodellierung.uni-graz.at/" target="_blank">...</a>
          <a href="http://creativecommons.org/licenses/by-nc/4.0/" target="_blank">...</a>
        </div>
      </div>
    </div>
  </div>
```

Common Attributes

- id** = Unique element identifier (Eindeutige ID für Elemente)
- class** = class(es) of an element (vergibt Klassen an Elemente)
- style** = Stylesheet-Regeln (für “inline”-Stile)

Wichtig für CSS!

-
- <div>** = Mit zusätzlichen Attributen werden div's Eigenschaften zugeordnet. `div` hat häufig das `id` oder `class`-Attribut.
 - ** = Hauptaufgabe von `span` ist es inline-Textabschnitten Eigenschaften zuzuordnen. `span` hat häufig das `style`-Attribut.

id, class und style: Wann benutze ich was?

Attribute: **CSS-Klassen** gruppieren mehrere Elemente

- Der genaue Unterschied einer **CSS-ID** und **CSS-Klasse** ist zunächst nicht leicht erkennbar
- Weder im HTML-Dokument noch auf der Webseite lässt sich festmachen, warum beide Attribute notwendig sind
- Ein Blick auf das semantische Regelwerk schafft Abhilfe
- Denn wie in jeder Sprache gibt es Normen, die es zu befolgen gilt

Attribute: CSS-Klassen gruppieren mehrere Elemente

- Im HTML-Dokument dienen Klassen (*class*) dazu, **mehrere Tags zu gruppieren**
- Das hat den Vorteil, dass **alle Elemente einer Klasse im CSS angesprochen** und einheitlich designt werden können
- Klassen sind nicht auf einen HTML-Tag beschränkt, sondern beziehen sich auf mehrere Elemente
- Einer Klasse können im HTML-Dokument auch weitere verschiedene Elemente, wie etwa Überschriften (*h1, h2, h3...*), Block-Absätze (*div*), Absätze (*p*) und weitere Tags, zugewiesen werden

Attribute: **CSS-ID** bestimmt ein einzelnes Element

- Im Gegensatz zur *class* sind **IDs eindeutige Attribute**
- IDs zeichnen im HTML-Dokument **lediglich ein Element aus**
- So darf eine bestimmte ID im gesamten Code auch nur einmal definiert werden
- Bei der Benennung der ID folgt auf das Attribut ebenfalls eine spezifische Bezeichnung

class und ID: Unterschied

- Egal ob eine *ID* oder eine *class* im CSS angesprochen wird, das Erscheinungsbild auf der Webseite ist bei gleicher Formatierung dasselbe
- Der wichtige Unterschied liegt jedoch in der Semantik:
IDs werden im CSS höher gewertet als Klassen
- Das bedeutet, wenn ein Element sowohl einer Klasse als auch einer ID angehört, wird die Formatierung der ID ausgegeben
- Daher sollten IDs auch unbedingt als solche genutzt und nicht durch das *class*-Attribut ersetzt werden
- Nur so kann eine **saubere Semantik** im Dokument beibehalten werden

Attribute: Häufige Fehlerquellen bei der Anwendung

- Eine der häufigsten Fehlerquellen ist die Schreibweise der Klassen- und ID-Namen
- Die **Bezeichnungen sind case-sensitiv**, was bedeutet, dass Groß- und Kleinschreibung ausschlaggebend sind
- Wird die Klasse beispielsweise mit „**h**auptthema“ definiert, im CSS aber mit „**H**auptthema“ angesprochen, kann die Formatierung nicht ausgespielt werden
- Die Namen dürfen außerdem **keinerlei Umlaute oder Leerzeichen enthalten**
- Ziffern von null bis neun können für die Benennung genutzt werden, dürfen jedoch nicht zu Beginn der Bezeichnung stehen
- **Bei der Erstellung von HTML und CSS unbedingt auf eine korrekte Schreibweise achten!**

- Der **Unterschied zwischen Klassen und IDs ist wichtiger**, als er zunächst erscheint
- Während Klassen nicht eindeutige Attribute sind und mehrere Elemente definieren, zeichnet die ID immer nur ein Element aus
- Semantisch werden IDs in der Ausführung höher gewertet als Klassen
- Achten Sie bei der Definition im HTML-Dokument stets auf die korrekte Nutzungsweise und die korrekte Bezeichnung, um im CSS die gewollten Formatierungen umzusetzen

Quelle: [HubSpot](#)

F R A G E N ?

Ü B U N G

12:45 - 13:00 Uhr

Mittag

14:45 - 16:15 (15:15) Uhr

Fazit & Ergebnissicherung

15:15 - 16:15 Uhr