WebDesign

HERZLICH WILLKOMMEN !

Themen Vorlesung 9

- CSS Style Guide
- Flexbox

Themen Vorlesung 9

- CSS Style Guide

Mehrere Selektoren: jeweils eine eigene Zeile

```
h1,
h2,
h3 {
  font-weight: normal;
  line-height: 1.2;
}
```

Namen: kleingeschrieben mit Bindestrichen

```
.meine-klasse {
  font-weight: normal;
  line-height: 1.2;
}
```

Semikolon nach der letzten Deklaration in einer Regel

```
.test {
  display: block;
  height: 10em;
}
```

Alphabetische (oder eine sinnvolle) Sortierung der Eigenschaften in einer Regel

```
background: fuchsia;
border: .1em solid;
border-radius: .4em;
color: black;
text-align: center;
text-indent: 2em;
```

Entities

- Nicht nutzen, when vermeidbar

Themen Vorlesung 9

- Flexbox

Flexbox / Flex-Container

- Eine spezielle Container-Rolle für HTML-Elemente
- CSS-Deklaration: `display: flex;` macht ein Element (in der Regel ein `div` oder ein `main`) zu einem Flex-Container.
- Elemente im Container werden automatisch zu "Flex-Items" und stehen unter Verwaltung des Containers
- Geschachtelte Flex-Container möglich
- Weitere Eigenschaften/Gestaltungsoptionen auf der Ebene des Containers und/oder auf Ebene der Flex-Items.

CSS-Eigenschaften für Flex-Container (zusätzlich zur `display: flex;`)

- `flex-direction`
- `flex-wrap`
- `justify-content`
- `align-items`
- `align-content`
- `column-gap`
- `row-gap`
- aufgelistet ungefähr in der Reihenfolge von abnehmender Bedeutung
- darüber hinaus gibt es kombinierte Eigenschaften (shorthands)

CSS-Eigenschaften für Flex-Items

- `flex-basis`
- `flex-grow`
- `flex-shrink`
- `align-self`
- `order`
- aufgelistet ungefähr in der Reihenfolge von abnehmender Bedeutung
- darüber hinaus gibt es kombinierte Eigenschaften (shorthands)

flex-direction

- DEFAULT Hauptachse nach rechts. Querachse nach unten.
  - `flex-direction: row;`
- Hauptachse nach links. Querachse nach unten.
  - `flex-direction: row-reverse;`
- Hauptachse nach unten. Querachse nach rechts
  - `flex-direction: column;`
- Hauptachse nach oben. Querachse nach rechts
  - `flex-direction: column-reverse;`

justify-content

- "linksbündig"
  - `justify-content: flex-start;`
- "rechtsbündig"
  - `justify-content: flex-end;`
- zentriert
  - `justify-content: center;`
- verteilt
  - `justify-content: space-around`
- verteilt (an die Ecken gedrängt)
  - `justify-content: space-between`

justify-content:

Anordnung entlang der Hauptachse

```
normal | <content-distribution> | <overflow-position>? [
<content-position> | left | right ]

where

<content-distribution> = space-between | space-around |
space-evenly | stretch

<overflow-position> = unsafe | safe

<content-position> = center | start | end | flex-start |
flex-end
```

align-items:

Anordnung entlang der Querachse

```
align-items: normal;
align-items: stretch;
align-items: center;
align-items: start;
align-items: end;
align-items: flex-start;
align-items: flex-end;
align-items: baseline;
align-items: first baseline;
align-items: last baseline;
align-items: safe center;
align-items: unsafe center;
```

align-self: Anordnung entlang der Querachse für einzelne Flex-Items

```
align-self: auto;
align-self: normal;
align-self: center;
align-self: start;
align-self: end;
align-self: self-start;
align-self: self-end;
align-self: flex-start;
align-self: flex-end;
align-self: baseline;
align-self: first baseline;
align-self: last baseline;
```

flex-wrap: Ob eine zweite Reihe aufgemacht werden darf

```
flex-wrap: nowrap; /* DEFAULT */
flex-wrap: wrap;
flex-wrap: wrap-reverse;
```

`flex-grow flex-shrink flex-basis`

- Für Flex-Items: Wenn zu viel oder zu wenig Platz zur Verfügung steht.
- `flex-basis`: die normale Größe
- `flex-grow`: mit welcher Priorität den freien Platz beanspruchen
- `flex-shrink`: mit welcher Priorität Platz abgeben
- `flex`: kombinierte Eigenschaft auf `flex-grow`, `flex-shrink` und `flex-basis`
- `flex-grow` und `flex-shrink`: Werte: Zahlen, die nur relativ zu anderen Flex-Items im selben Container bewertet werden