

# Schulung Web-Entwicklung

Tag 3:  
BEM, Vertiefung HTML,  
Vertiefung CSS

Michael Brüggemann  
Moritz Hipper



# Übersicht

**01**

**Vertiefung CSS**  
Selektoren

**02**

**BEM**  
**Block Element Modifier**

**03**

**Vertiefung CSS**  
Kombinatoren, Flex-Box,  
Variablen, Transitions

**Vertiefung CSS**

**01**



# Selektoren Kombinieren 1

- Die CSS-Syntax erlaubt das **Darstellen von Relationen** von durch Selektoren

→ `.klasseeins.klassezwei { ...`

`<div class=„klasseeins klassezwei“`

→ `.klasseeins .klassezwei { ...`

`<div class=„klasseeins“>`

`<div class=„klassezwei“></div>`

`</div>`

→ `input.klassenname { ...`

`<input class=„klassenname“>`



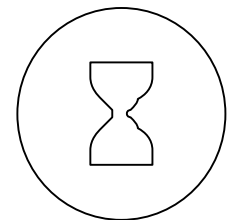
# Styles für mehrere Selektoren

- Zusätzlich ist es möglich, **Styles für mehrere Selektoren gleichzeitig** zu definieren  
→ `.klasseeins, input, #idstring, input:hover { ...`

# Aufgabe 10

Euer Chef will den Kunden ein „Daily Special“ anbieten.

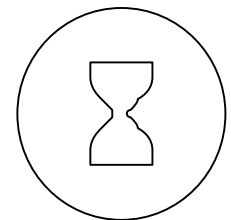
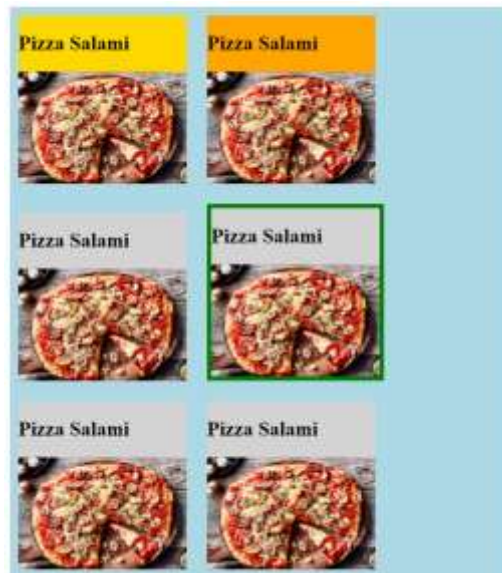
Dieses hebt sich von den anderen Pizzavorschauselementen ab: Es hat einen grünen, dicken Rand. Außerdem ist euer Chef der Meinung, dass die Pizzavorschauselemente nebeneinander und nicht untereinander platziert sein sollten.



15 Minuten

# Aufgabe 11

Nun ist euer Chef etwas zufriedener. Die Pizzavorschau sehen nun gut aus und wirken interaktiv. Allerdings sieht im das immer noch alles viel zu unübersichtlich aus. Sorge dafür, dass die Gesamtheit der Pizzavorschaulemente maximal 600px Breite einnimmt. Außerdem sollen sie mit einem hellblauen Hintergrund versehen werden



15 Minuten

# BEM (Block Element Modifier)

# 02







# Grundlagen / Motivation

- **Block Element Modifier**
- **Probleme** großer Softwareprojekte
  - keine Namenskonventionen
  - viele **komplexe CSS – Klassen**, die eigentlich etwas ähnliches tun
  - **unübersichtlicher** Code
- Was **BEM** für **Vorteile** mit sich bringt
  - **Namenskonventionen!**
  - Logische, **modulare Struktur** in den Styledateien
  - keine Stylekonflikte
  - Wartbarkeit und Erweiterbarkeit
  - Verständlichkeit

# BEM in der Praxis

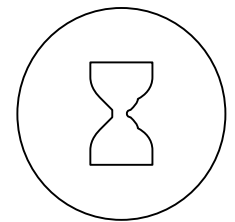
- Der **Block** definiert eine **funktional zusammenhängende Gruppe** von Elementen, wie zum Beispiel ein Bestellformular  
→ `.bestellung { ...`
- **Elemente innerhalb** dieser Gruppe haben nun immer den mit `__` verbundenen **Blocknamen als Präfix**  
→ `.bestellung__input { ...`
- Sollen Elemente des Blocks in ihrem **Aussehen abweichen**, werden sie durch einen **Modifier** ergänzt  
→ `.bestellung__input-groß { ...`  
oder  
→ `.bestellung-spezial { ...`



## Aufgabe 12

Euer Chef will irgendwann mal das Amazon der Pizzaliefersdienste werden und will deshalb euer Projekt auf großes Wachstum und einfache Wartbarkeit vorbereiten.

*Refactored alle Styledefinitionen nach BEM*



15 Minuten

**Vertiefung CSS**

**03**



# Selektoren Kombinieren 2

- Es können auch **komplexere Relationen** von Elementen angesprochen werden

→ **Kindkombinator** (für Kapselung bei Globalen Styles)

```
.klasseins > .klassezwei { ...
```

Hier wird klassezwei nur angesprochen, wenn es ein Kind der klasseins ist

→ **Geschwisterkombinator**

```
.klasseins ~ .klassezwei { ...
```

Hier wird klassezwei nur angesprochen, wenn es auf gleicher ebene wie klasseins liegt

→ **Nachbarkombinator**

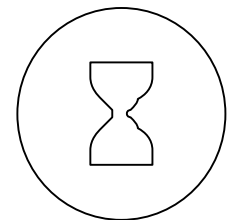
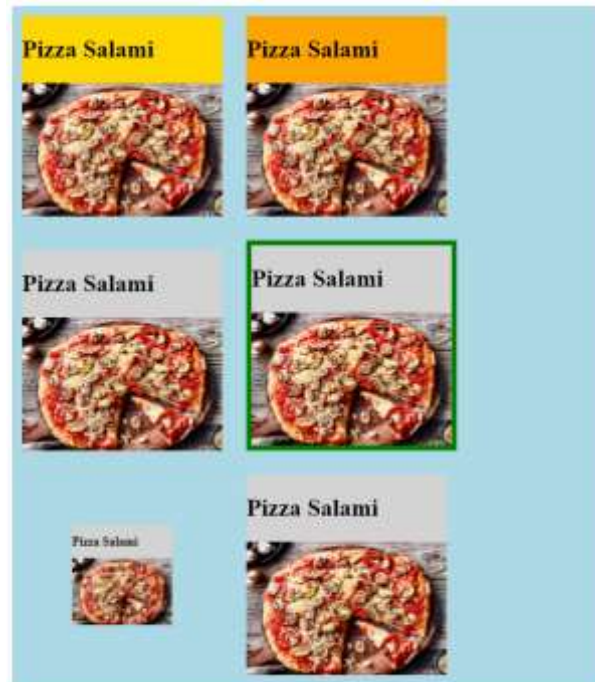
```
.klasseins + .klassezwei { ...
```

Hier wird klassezwei nur angesprochen, wenn es direkt neben klasseins liegt

# Aufgabe 13

Das Daily Special soll besonders herausstechen. Macht, dass das Element rechts daneben kleiner erscheint.

*Verzichtet auf das schreiben neuer Klassen. Nutzt intelligent die Kombinatoren.*



15 Minuten



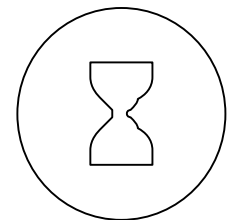
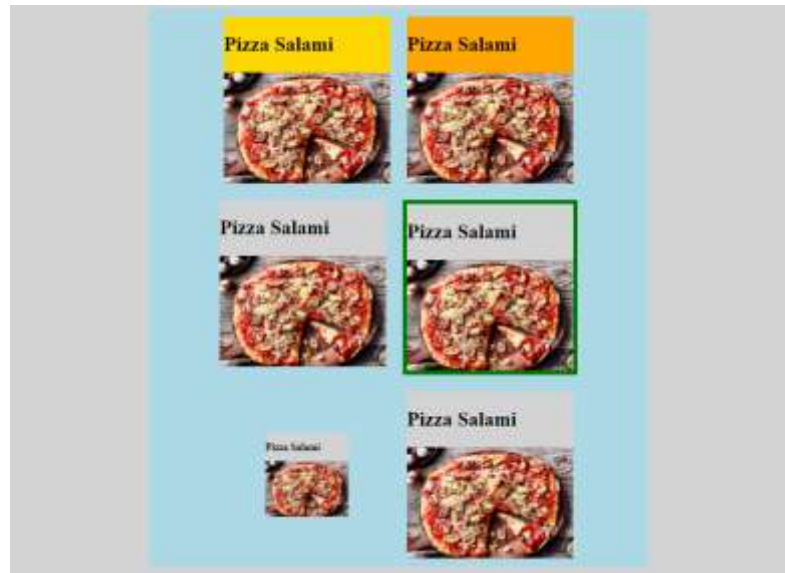
# Flexbox - Layout

- So gut wie die **Basis von modernem responsivem Webdesign**
- Viele verschieden Bildschirmgrößen
- Erlaubt es ohne großen aufwand die **Anordnung und Größe von Elementen auf die Größe ihres Containers anzupassen**
- Zur Anwendung des Flexbox – Layout benötigt es
  - einen Container mit dem CSS-Attribut `display: flex`
  - 0 – n Elemente in diesem Container

Sieht auch: <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

# Aufgabe 14

Nun will euer Chef, dass ihr alle Pizzavorschau-elemente auf die Mitte der Website zentriert. Außerdem soll dafür gesorgt sein, dass sie nie den rechten oder Linken Bildschirmrand verlassen.



10 Minuten





# Aufgabe 15

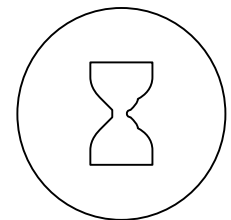
Vor Ewigkeiten hat euer Chef mal ein Bestellformular beauftragt. Sorgt auch dafür, dass dieses maximal so breit wird wie die Gesamtheit der Pizzavorschauselemente (600px). Zusätzlich sollte auch diese zentriert liegen.

**Pizza Papa**

Bei uns nur das Beste!

Größe  Nachricht für den Empfänger

Belag

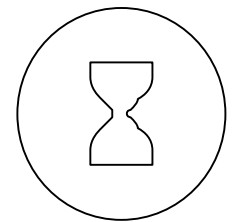


10 Minuten



# Aufgabe 16

Sowohl das Bestellformular als auch die Pizzavorschau sind maximal 600px breit und auf der Seite zentriert. Ein Seniorentwickler hat eure Mehrfache Definition des selben Styles gesehen und sich beschwert. Sorgt dafür, dass ihr die Styles, die für die Zentrierung zuständig sind, nur ein mal definiert.



10 Minuten

# Variablen

- Ja, auch hier gibt es Variablen.
- Um ein **mehrfaches definieren** des selben Stylewertes zu **vermeiden**, lassen sich variablen nutzen
- Sie werden für einen **bestimmten Scope** innerhalb einer CSS-Selektion definiert
- Zur globalen Nutzung in Pseudo :root – Element:

```
:root {  
    --background-color-light: lightgray;  
}
```

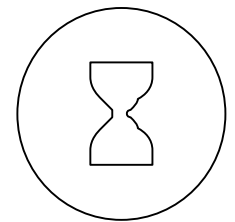
- Das Auslesen funktioniert wie folgt:

```
.klassenname {  
    background-color: var(--background-color-light);  
}
```



# Aufgabe 17

Eurem Chef ist Skalierbarkeit immer noch sehr wichtig. Schaut, welche Styleattribute ihr in Variablen auslagern könnt. Sorgt außerdem dafür, dass sowohl die Pizzavorschau und das Bestellformular einen besonderen Schrifttyp haben, der zur Identität der Website passt.



20 Minuten



# Transitions

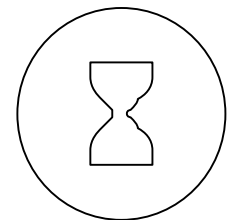
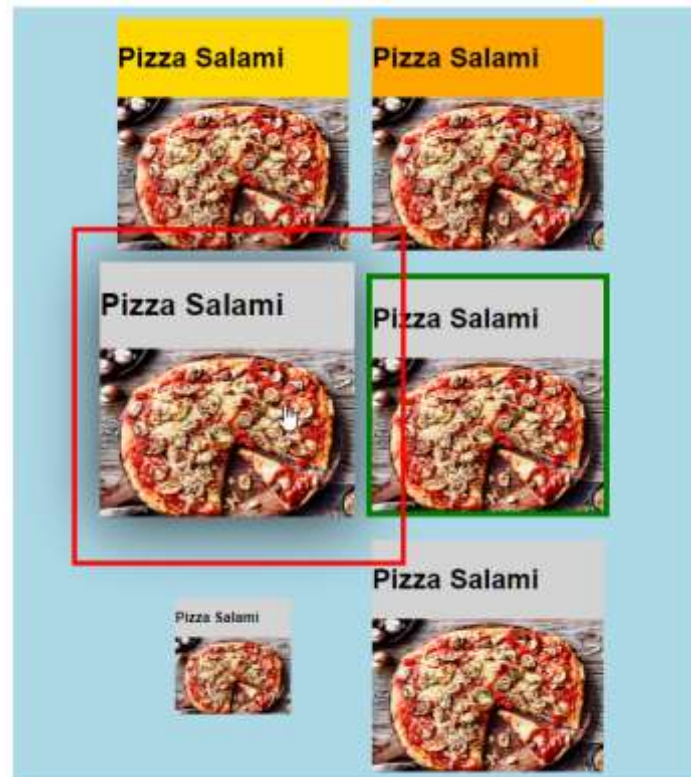
```
button {  
  background-color: white;  
  transition: background-color 0.3s;  
}
```

```
button:hover {  
  background-color: blue;  
}
```

- Nutzen
  - Verbessern die User Experience
  - Werten die Website visuell auf
  - Stellen Zusammenhänge klarer dar
  - Website wirkt „organischer“
- Smoother Übergang von einem Styleattributwert zu dem Nächsten
  - Damit ein Transition Möglich ist, muss also sowohl im Ziel als auch im Startstyle das Attribut vorhanden sein
  - Nicht alle Styleattribute unterstützen Transitions

# Aufgabe 18

Euer Chef will, dass der Effekt, welcher beim Herüberfliegen mit der Maus über eine Pizzavorschau geschieht, weicher erscheint. Außerdem, soll die Pizzavorschau dabei dem Nutzer „etwas entgegenfliegen“, also etwas größer skaliert werden und einen größeren Schatten werfen.

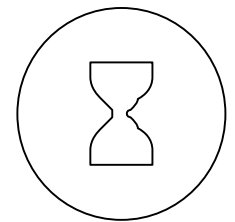


20 Minuten

# Aufgabe 19

Euer Chef will viel mehr Funktionen in die Website integrieren. Hierzu muss der Nutzer die Möglichkeit haben, über einen Menüstruktur dahinzugelangen. Baut ein Menü mit einem Link zu dem Bestellformular und 4 untereinander angeordneten Platzhalterlinks, welches immer auf der Linken Seite der Website schwebt und 250px breit ist. Hierzu hat euch euer Chef eine Skizze gezeichnet.

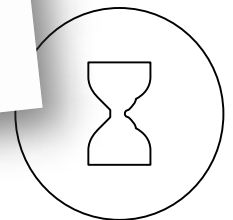
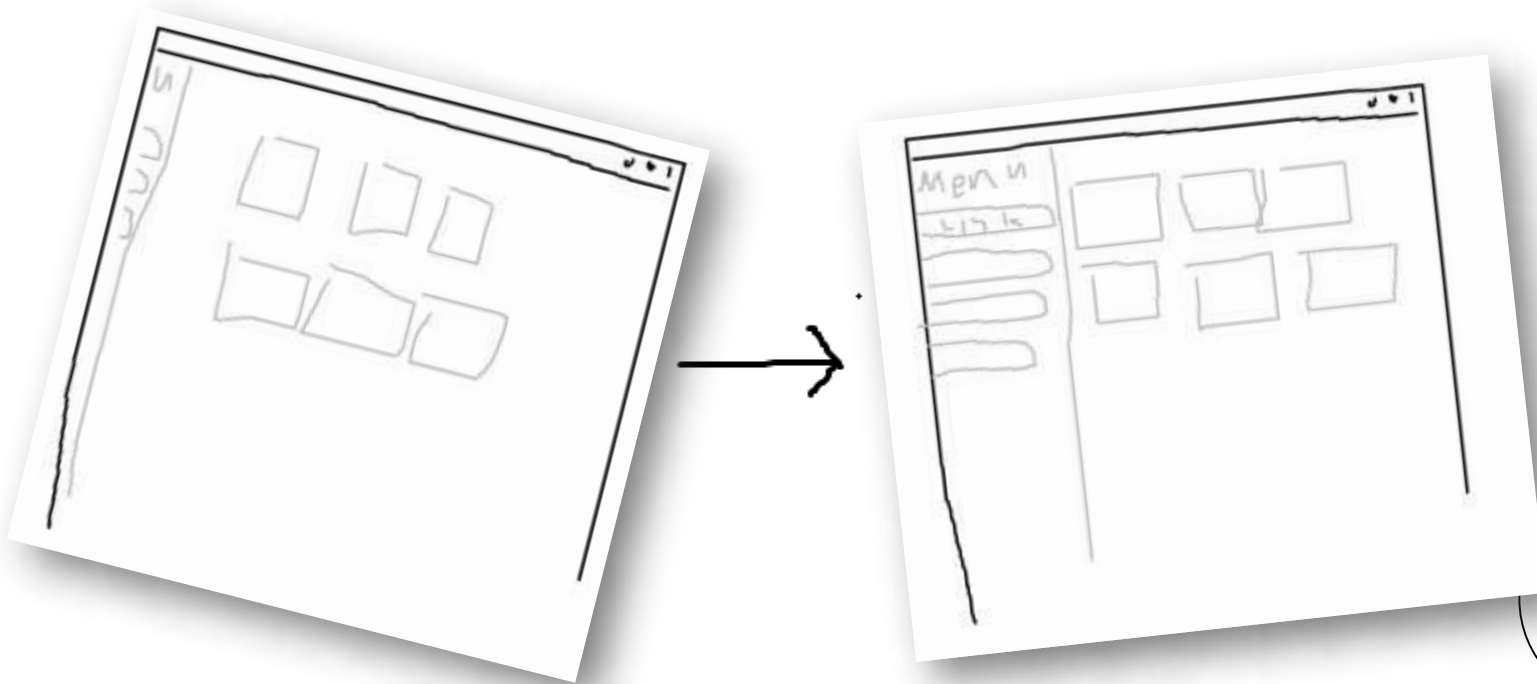
*! Legt neue Variablen an  
oder nutzt bestehende,  
falls nötig*



45 Minuten

# Aufgabe 20

Das Menü ist eurem Chef zu präsent. Er wünscht sich, dass es sich versteckt und nur dann ganz erscheint, wenn der User mit der Maus darüberschwebt.



15 Minuten