

IA Konferenz 2014

# <Developing UXD>

So leicht ist Frontendentwicklung für  
Konzepter und Designer heute

Conni und Jerry: Wer wir sind

Conni



Cornelia Erbs

Freelancerin als Senior User  
Experience Architect

Davor Senior User Experience Architect  
für große Hamburger Agenturen

„HTML-Wireframing ist so spannend  
wie Rätsel lösen.“

# Jerry



## Jeremias Dombrowsky

Senior Frontend Engineer bei XING

Davor Creative Technologist für  
namhafte Agenturen und Unternehmen

„Wenn UX Designer HTML und CSS  
lernen, haben Entwickler mehr Zeit  
zum Programmieren.“





# Ziele des Workshops

- Quellcode lesen und verstehen

# Ziele des Workshops

- Quellcode lesen und verstehen
- Die erste Hürde des Selbst-Codens nehmen



# Ziele des Workshops

- Quellcode lesen und verstehen
- Die erste Hürde des Selbst-Codens nehmen
- Wissen, wie man weiter lernen kann

Nach dem Workshop könnt ihr noch nicht:

Axure wegschmeißen  
und direkt alle Wireframes  
per Code erzeugen.



Einführung: Warum wir dazu lernen müssen



# Warum sollten wir HTML und CSS lernen?

- Die Basis dessen, was wir gestalten, sollten wir kennen.

# Warum sollten wir HTML und CSS lernen?

- Die Basis dessen, was wir gestalten, sollten wir kennen.
- Zumal: Die Tools wechseln, HTML bleibt!

# Warum sollten wir HTML und CSS lernen?

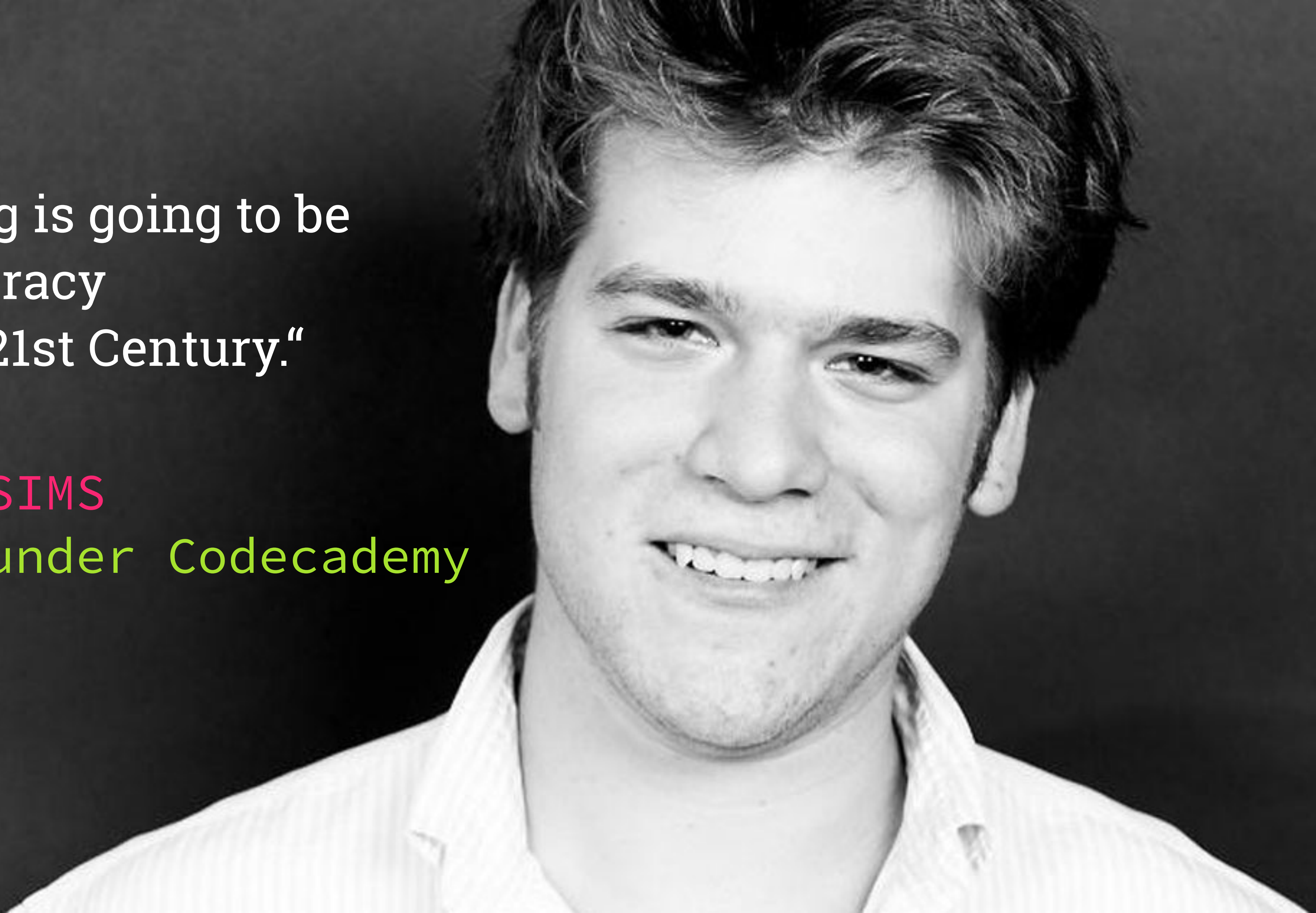
- Die Basis dessen, was wir gestalten, sollten wir kennen.
- Zumal: Die Tools wechseln, HTML bleibt!
- Und: Eine neue Generation „native Developers“ kommt.



“Coding is going to be  
the literacy  
of the 21st Century.”

ZACH SIMS

Co-Founder Codecademy





# Es kann unsere Arbeitsweise verbessern!

Anforderungen  
klären



Wireframes  
erstellen



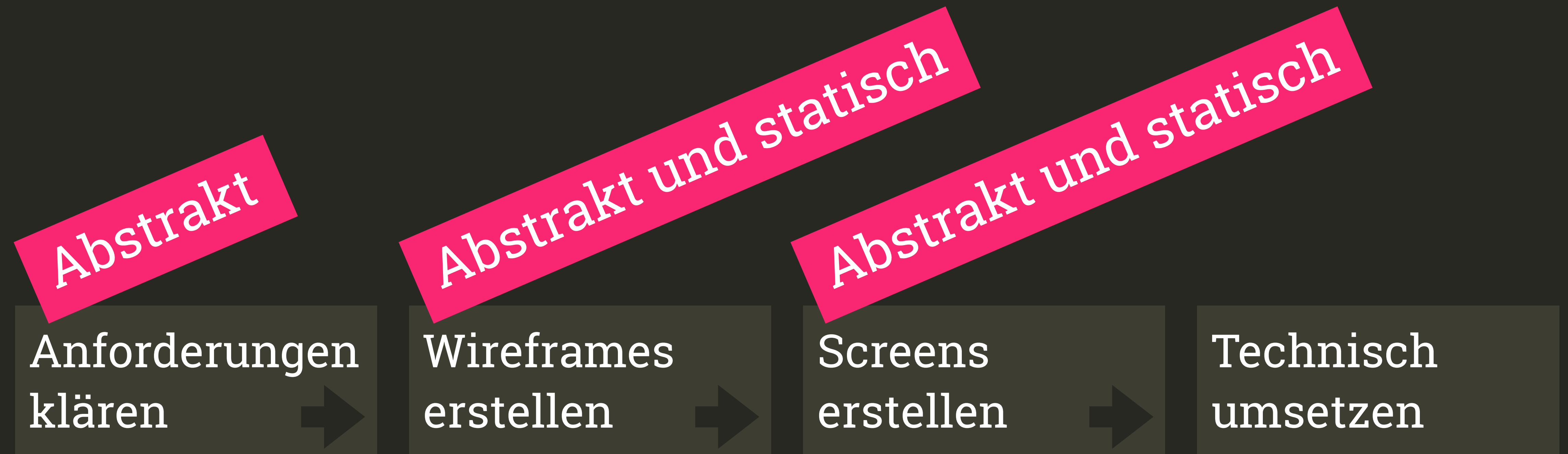
Screens  
erstellen



Technisch  
umsetzen

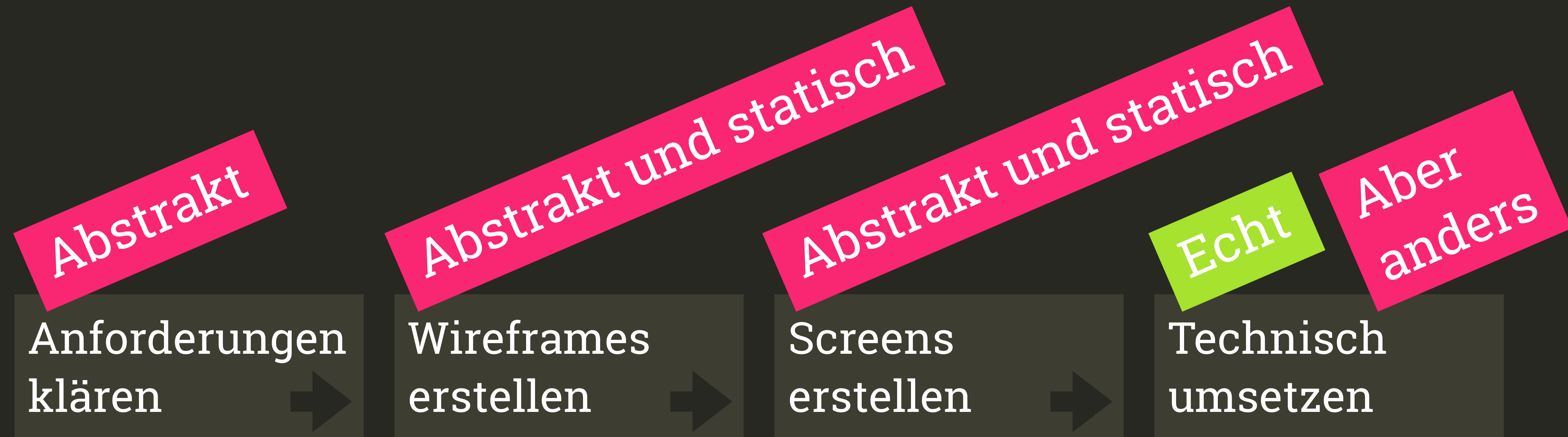
## Heutige Arbeitsweise

# Es kann unsere Arbeitsweise verbessern!



## Heutige Arbeitsweise

# Es kann unsere Arbeitsweise verbessern!



## Heutige Arbeitsweise

Es kann unsere Arbeitsweise verbessern!

Zukünftige Arbeitsweise

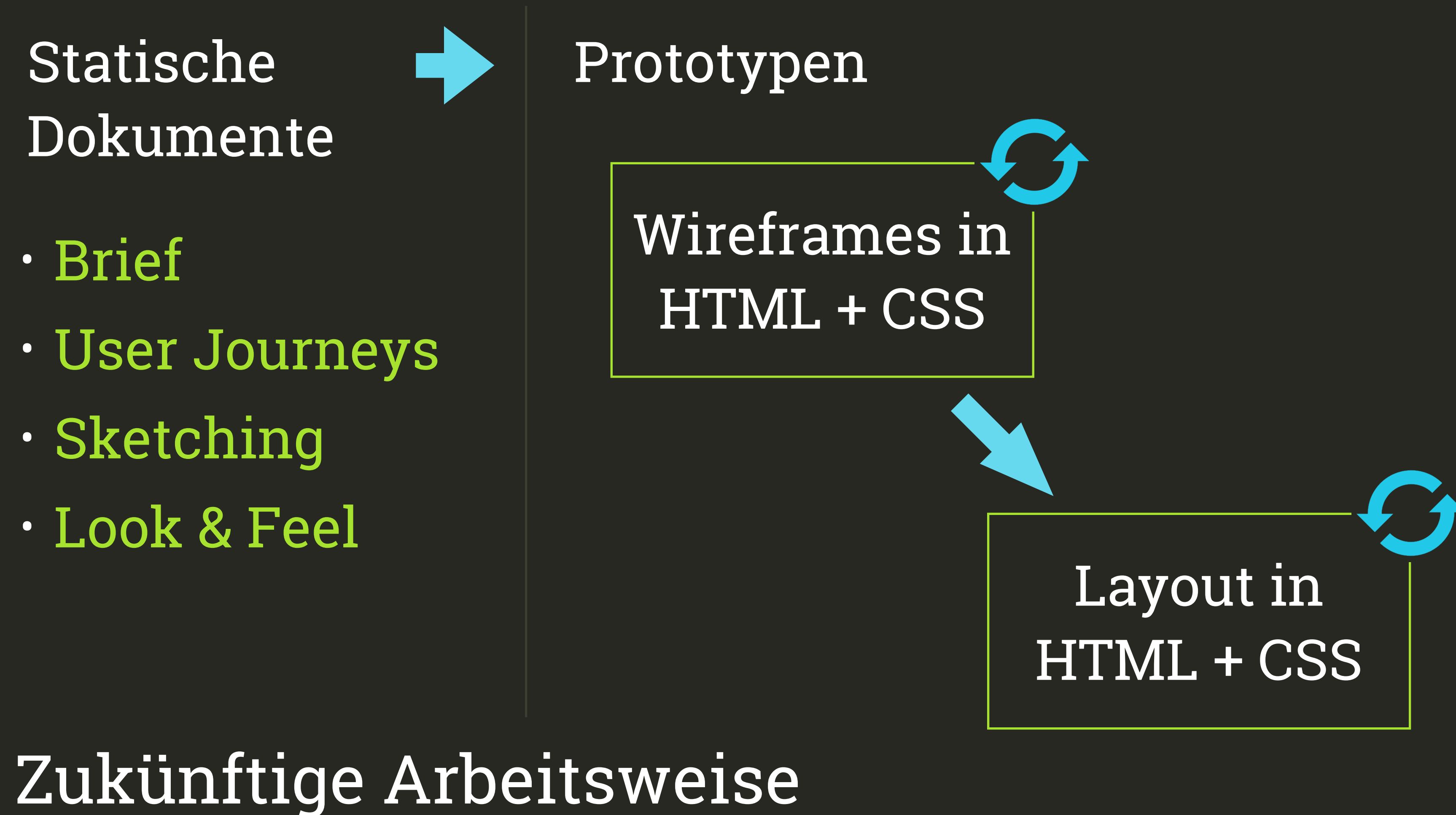
# Es kann unsere Arbeitsweise verbessern!

Statische  
Dokumente

- Brief
- User Journeys
- Sketching
- Look & Feel

Zukünftige Arbeitsweise

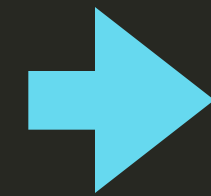
# Es kann unsere Arbeitsweise verbessern!





# Es kann unsere Arbeitsweise verbessern!

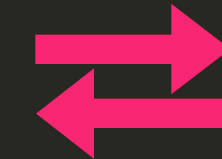
Statische  
Dokumente



Prototypen

- Brief
- User Journeys
- Sketching
- Look & Feel

Wireframes in  
HTML + CSS

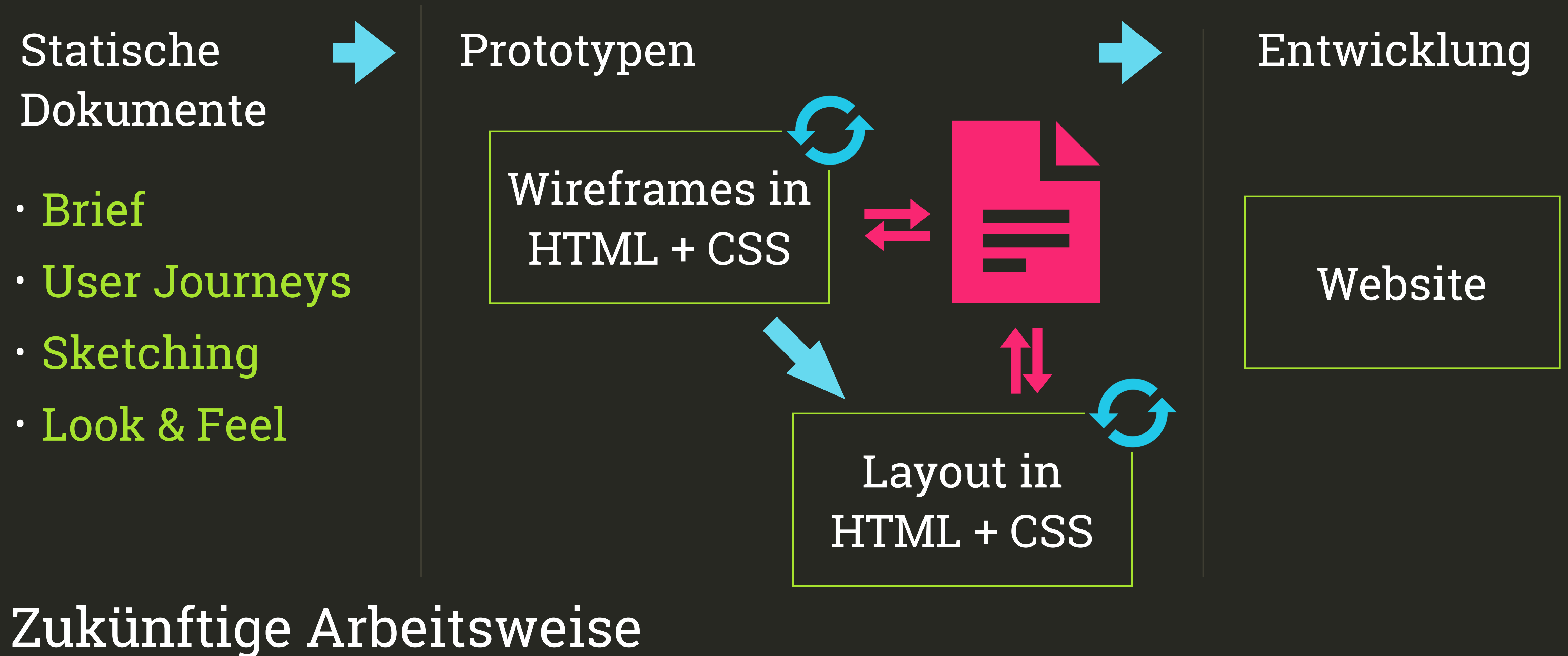


Layout in  
HTML + CSS

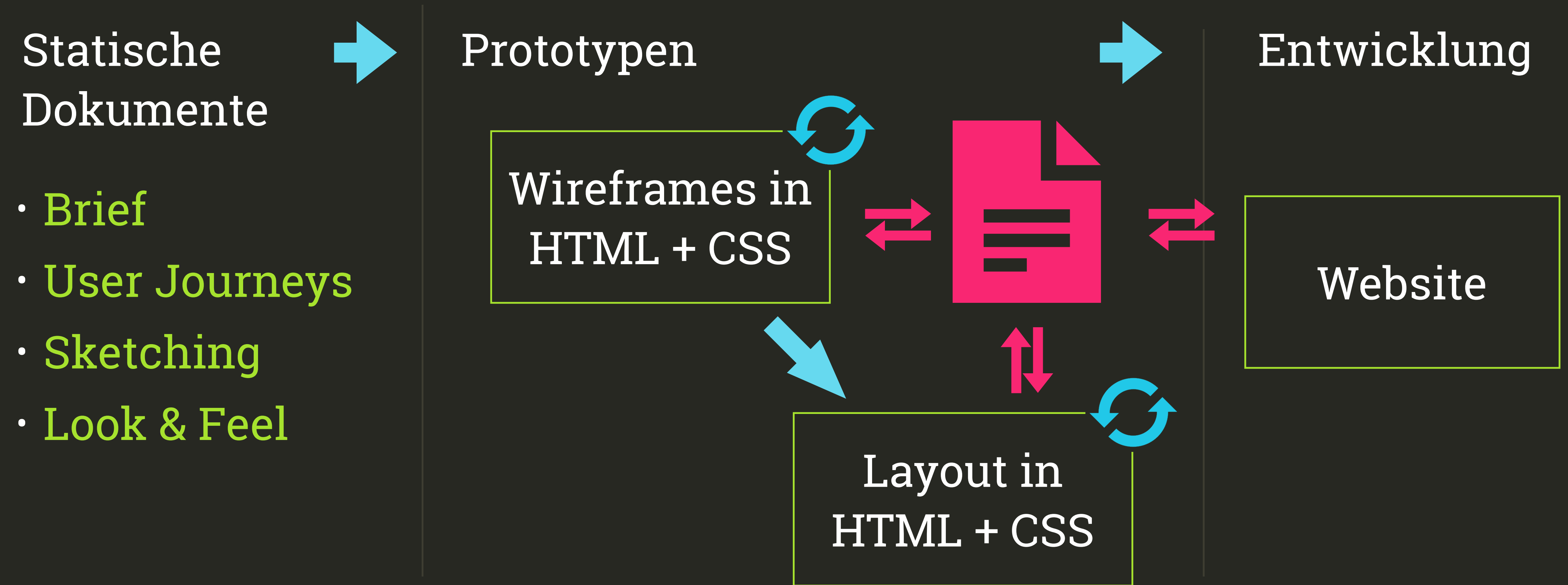


Zukünftige Arbeitsweise

# Es kann unsere Arbeitsweise verbessern!



# Es kann unsere Arbeitsweise verbessern!



Zukünftige Arbeitsweise

Ende der Einführung. Fragen?

# Agenda

- ✓ Einführung
- HTML-Grundlagen
- CSS-Grundlagen
- Bootstrap
- Ausblick



# HTML: Aufbau und Inhalte einer Website

# Interaktiv: Layout-Elemente kennenlernen



# HTML: Layout-Elemente

# HTML: Layout-Elemente

- header, footer, aside, nav, main, article, section:  
"Sectioning elements" gruppieren inhaltlich

# HTML: Layout-Elemente

- header, footer, aside, nav, main, article, section:  
"Sectioning elements" gruppieren inhaltlich
- **main:** Einzigartig! Hauptinhalte kommen hier rein.

# HTML: Layout-Elemente

- **header, footer, aside, nav, main, article, section:**  
"Sectioning elements" gruppieren inhaltlich
- **main:** Einzigartig! Hauptinhalte kommen hier rein.
- **article:** Muss alleine stehen können ( RSS-Feed, Pocket)

# HTML: Layout-Elemente

- **header, footer, aside, nav, main, article, section:** "Sectioning elements" gruppieren inhaltlich
- **main:** Einzigartig! Hauptinhalte kommen hier rein.
- **article:** Muss alleine stehen können ( RSS-Feed, Pocket)
- **nav:** Nur für Navigation auf der Website

# HTML: Layout-Elemente

- **header, footer, aside, nav, main, article, section:** "Sectioning elements" gruppieren inhaltlich
- **main:** Einzigartig! Hauptinhalte kommen hier rein.
- **article:** Muss alleine stehen können (RSS-Feed, Pocket)
- **nav:** Nur für Navigation auf der Website
- **div:** Gruppiert für Styling, nicht inhaltlich (semantisch)

Interaktiv: Inhaltselemente kennenlernen

# HTML: Inhaltselemente – Textinhalte

- **h1, h2:** Heading. Pro **section** ein neues h1.
- **p:** Absatz ("paragraph")
- **a:** Link ("anchor"). Darf weitere Inhalte enthalten.



# HTML: Inhaltselemente – Weitere Inhalte

# HTML: Inhaltselemente – Weitere Inhalte

- `img`: "image" – Bildelement

# HTML: Inhaltselemente – Weitere Inhalte

- **img**: "image" – Bildelement
- **ul, ol**: "unordered/ordered list" – eine Liste mit Bulletpoints oder Nummerierungen.  
Darf nur **li**-Elemente enthalten.

# HTML: Inhaltselemente – Weitere Inhalte

- **img**: "image" – Bildelement
- **ul, ol**: "unordered/ordered list" – eine Liste mit Bulletpoints oder Nummerierungen.  
Darf nur **li**-Elemente enthalten.
- **li**: "List item" – darf andere Elemente enthalten.

# HTML: Inhaltselemente – Weitere Inhalte

- **img**: "image" – Bildelement
- **ul, ol**: "unordered/ordered list" – eine Liste mit Bulletpoints oder Nummerierungen.  
Darf nur **li**-Elemente enthalten.
- **li**: "List item" – darf andere Elemente enthalten.
- **audio, video, canvas, embed**: Media-Inhalte

Fragen soweit?

# HTML: Syntax

# Tags

```
<p>Das ist mein Absatz</p>
```



# Tags

Opening **<tag>**

Closing **</tag>**

**<p>**Das ist mein Absatz**</p>**

# Tags

Opening **<tag>**

Closing **</tag>**

**<p>**Das ist mein Absatz**</p>**

- Layout-Elemente haben "opening" und "closing tags"

# Tags

Opening `<tag>`

Closing `</tag>`

```
<p>Das ist mein Absatz</p>
```

- Layout-Elemente haben "opening" und "closing tags"
- Ermöglicht Verschachtelung

# Tags

Opening `<tag>`

Closing `</tag>`

```
<p>Das ist mein Absatz</p>
```

- Layout-Elemente haben "opening" und "closing tags"
- Ermöglicht Verschachtelung
- Elemente ohne "closing tags" heißen "Void Tags"

# Tags: Verschachtelung

```
<article>  
  <h1>Welcome</h1>  
  <p>Das ist mein Absatz, der <em>auch</em>  
verschachtelte Elemente enthält.</p>  
</article>
```

# Tags: Verschachtelung

Opening `<article>`

```
<article>  
  <h1>Welcome</h1>  
  <p>Das ist mein Absatz, der <em>auch</em>  
verschachtelte Elemente enthält.</p>  
</article>
```

Closing `<article>`

# Tags: Verschachtelung

Opening `<article>`

Nested `<h1>`

```
<article>  
  <h1>Welcome</h1>  
  <p>Das ist mein Absatz, der <em>auch</em>  
verschachtelte Elemente enthält.</p>  
</article>
```

Closing `<article>`

# Tags: Verschachtelung

Opening `<article>`

Nested `<h1>`

```
<article>  
  <h1>Welcome</h1>  
  <p>Das ist mein Absatz, der <em>auch</em>  
verschachtelte Elemente enthält.</p>  
</article>
```

Nested `<p>`

Closing `<article>`



# Tags: Verschachtelung

Opening `<article>`

Nested `<h1>`

```
<article>
```

```
  <h1>Welcome</h1>
```

```
  <p>Das ist mein Absatz, der <em>auch</em>  
verschachtelte Elemente enthält.</p>
```

```
</article>
```

Nested `<em>`

Nested `<p>`

Closing `<article>`

# Attribute

```

```

# Attribute

attribute  
= Eigenschaft

```

```

# Attribute

attribute  
= Eigenschaft

"value"  
= Wert der Eigenschaft

```

```

# Attribute

**attribute**  
= Eigenschaft

**"value"**  
= Wert der Eigenschaft

```

```

- Elemente haben unterschiedliche **attributes**

# Attribute

**attribute**  
= Eigenschaft

**"value"**  
= Wert der Eigenschaft

```

```

- Elemente haben unterschiedliche **attributes**
- Es gibt **attributes** ohne **value**  
z.B. `<input type="email" required>`

Attribute: Wichtige

Attribute: Wichtige

```
<input id="i-am-unique">
```



## Attribute: Wichtige

```
<input id="i-am-unique">
```

Kennzeichnet ein **zwingend einzigartiges** Element, z.B. **input** im Formular; **#jumpmark** im gleichen Dokument

## Attribute: Wichtige

```
<input id="i-am-unique">
```

Kennzeichnet ein **zwingend einzigartiges** Element, z.B. **input** im Formular; **#jumpmark** im gleichen Dokument

```
<a class="nav-link nav-link--active">Home</a>
```

## Attribute: Wichtige

```
<input id="i-am-unique">
```

Kennzeichnet ein **zwingend einzigartiges** Element, z.B. **input** im Formular; **#jumpmark** im gleichen Dokument

```
<a class="nav-link nav-link--active">Home</a>
```

Wird fürs Styling verwendet. Elemente können mehrere "class names" enthalten.

# Attribute: Beispiele

# Attribute: Beispiele

```

```

# Attribute: Beispiele

```
  
<a href="contact.html">Contact</a>
```

# Attribute: Beispiele

```

```

```
<a href="contact.html">Contact</a>
```

```
<a href="http://google.com" target="_blank">Leave</a>
```

# Attribute: Beispiele

```

```

```
<a href="contact.html">Contact</a>
```

```
<a href="http://google.com" target="_blank">Leave</a>
```

```
<label for="user-mail">Your email address:</label>
```



# Attribute: Beispiele

```
  
<a href="contact.html">Contact</a>  
<a href="http://google.com" target="_blank">Leave</a>  
<label for="user-mail">Your email address:</label>  
<input id="user-mail" type="email"
```

# Attribute: Beispiele

```
  
<a href="contact.html">Contact</a>  
<a href="http://google.com" target="_blank">Leave</a>  
<label for="user-mail">Your email address:</label>  
<input id="user-mail" type="email"  
      placeholder="me@me.io" required>
```

# Attribute: Beispiele

```
  
<a href="contact.html">Contact</a>  
<a href="http://google.com" target="_blank">Leave</a>  
<label for="user-mail">Your email address:</label>  
<input id="user-mail" type="email"  
      placeholder="me@me.io" required>  
<script src="js/scripts.js"></script>
```

# Attribute: Beispiele

```
  
<a href="contact.html">Contact</a>  
<a href="http://google.com" target="_blank">Leave</a>  
<label for="user-mail">Your email address:</label>  
<input id="user-mail" type="email"  
      placeholder="me@me.io" required>  
<script src="js/scripts.js"></script>
```

- Die Attribute von **allen** Elementen findet ihr unter MDN (Mozilla Developer Network)

Interaktiv: HTML-Elemente untersuchen

# HTML-Elemente untersuchen

- **Google Chrome:** "Chrome Developer Tools (DevTools)"
- **Rechtsklick auf Element:** "Element untersuchen"  
(Profis drücken `cmd+shift+c`)
- Aufklappen, Auswählen, per "Brotkrumen" navigieren
- Text ändern, Elemente verschieben und löschen

HTML: Dokument erstellen

# Sublime Text 3 einrichten: komplett

- **Download:** <http://www.sublimetext.com/3>
- **Einstellungen:** "Default Settings" in "User Settings" kopieren und anpassen
- **Package Control** installieren für Erweiterungen
- **cmd+shift+p "Package Install":**  
Emmet, SidebarEnhancements, BracketHighlighter, Color Highlighter



# Interaktiv: Dokument erstellen





# Dokument erstellen

- Mit Emmet: `html:5Tab`

# Dokument erstellen

- Mit Emmet: `html:5Tab`
- HTML5 Doctype: `<!doctype html>`

# Dokument erstellen

- Mit Emmet: `html:5Tab`
- HTML5 Doctype: `<!doctype html>`
- `head`: Metadaten, Dokumenttitel, Link zu Stylesheets

# Dokument erstellen

- Mit Emmet: `html:5Tab`
- HTML5 Doctype: `<!doctype html>`
- `head`: Metadaten, Dokumenttitel, Link zu Stylesheets
- `body`: Sichtbare Inhalte, Link zu JavaScripts am Ende

Interaktiv: HTML schreiben





# Weitere Ressourcen

- [codecademy.com](https://www.codecademy.com) – kostenloser Onlinekurs

# Weitere Ressourcen

- [codecademy.com](https://www.codecademy.com) – kostenloser Onlinekurs
- [developer.mozilla.org](https://developer.mozilla.org) – alles zu HTML und CSS

# Weitere Ressourcen

- [codecademy.com](https://www.codecademy.com) – kostenloser Onlinekurs
- [developer.mozilla.org](https://developer.mozilla.org) – alles zu HTML und CSS
- [stackoverflow.com](https://stackoverflow.com) – Fragen & Antworten

## Weitere Ressourcen

- [codecademy.com](https://www.codecademy.com) – kostenloser Onlinekurs
- [developer.mozilla.org](https://developer.mozilla.org) – alles zu HTML und CSS
- [stackoverflow.com](https://stackoverflow.com) – Fragen & Antworten
- [abookapart.com](https://abookapart.com) – Bücher zu HTML5, CSS3 und mehr

## Weitere Ressourcen

- [codecademy.com](https://www.codecademy.com) – kostenloser Onlinekurs
- [developer.mozilla.org](https://developer.mozilla.org) – alles zu HTML und CSS
- [stackoverflow.com](https://stackoverflow.com) – Fragen & Antworten
- [abookapart.com](https://abookapart.com) – Bücher zu HTML5, CSS3 und mehr
- [html5weekly.com](https://html5weekly.com) – HTML News

## Weitere Ressourcen

- [codecademy.com](https://www.codecademy.com) – kostenloser Onlinekurs
- [developer.mozilla.org](https://developer.mozilla.org) – alles zu HTML und CSS
- [stackoverflow.com](https://stackoverflow.com) – Fragen & Antworten
- [abookapart.com](https://abookapart.com) – Bücher zu HTML5, CSS3 und mehr
- [html5weekly.com](https://html5weekly.com) – HTML News
- [codepen.io](https://codepen.io) – Online Frontend Code schreiben und teilen

Das ist HTML. Fragen?

# Agenda

- ✓ Einführung
- ✓ HTML-Grundlagen
- CSS-Grundlagen
- Bootstrap
- Ausblick





# CSS: Elemente finden und verändern

Selektoren: class

```

```

# Selektoren: class

```

```

Nicht jedes `<img>` soll gleich gestylt werden

# Selektoren: class

```

```

Nicht jedes `<img>` soll gleich gestylt werden

```

```

# Selektoren: class

```

```

Nicht jedes `<img>` soll gleich gestylt werden

```

```

Durch das `class`-Attribut können wir das Element stylen

Selektoren: class

```

```

# Selektoren: class

```

```

```
.logo {  
  width: 200px;  
  height: 100px;  
}
```



# Selektoren: class

```

```

**.class-selector** beginnt immer mit einem Punkt (**.logo**).

```
.logo {  
  width: 200px;  
  height: 100px;  
}
```

# Selektoren: class

```

```

**.class-selector** beginnt immer mit einem Punkt (**.logo**).

```
.logo {  
  width: 200px;  
  height: 100px;  
}
```

- Wir verwenden (fast) immer **.class**-Selektoren!

# Selektoren: class

```
.logo {  
  width: 200px;  
  height: 100px;  
}
```

# Selektoren: class

```
.logo {  
  width: 200px;  
  height: 100px;  
}
```

Eine "declaration" besteht aus property und value.

# Selektoren: class

```
.logo {  
  width: 200px;  
  height: 100px;  
}
```

Eine "declaration" besteht aus property und value.

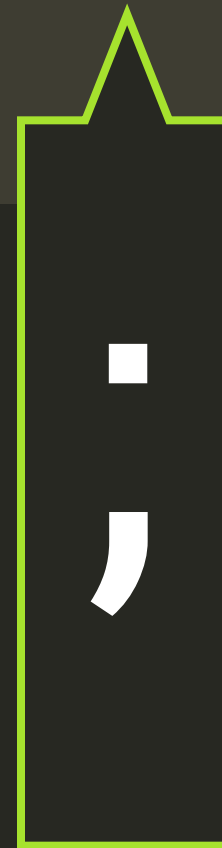
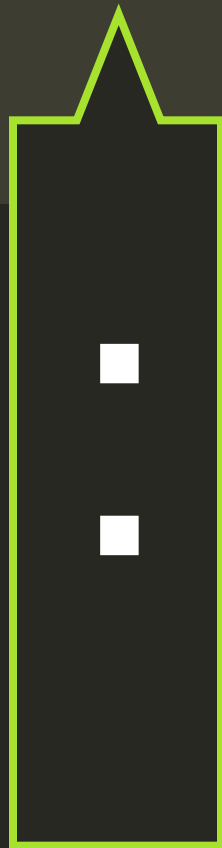
- 

-

# Selektoren: class

```
.logo {  
  width: 200px;  
  height: 100px;  
}
```

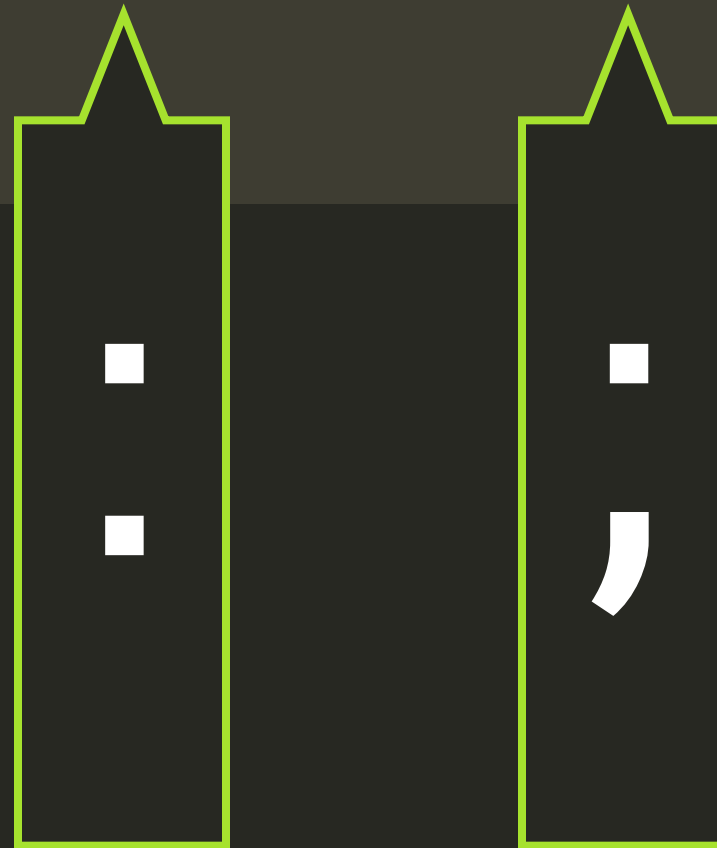
Eine "declaration" besteht aus property und value.



# Selektoren: class

```
.logo {  
  width: 200px;  
  height: 100px;  
}
```

Eine "declaration" besteht aus **property** und **value**.



**selector** und  
{ **declaration block** }  
bilden ein "Rule Set"

Interaktiv: CSS untersuchen



# CSS untersuchen

- Element auswählen, Declarations ein- und ausschalten, Werte ändern (auch mit Pfeiltasten und Shift)
- Element auswählen, class-Attribut hinzufügen und entfernen

# CSS: Learnings durch DevTools

- **h1**: "Element-Selektoren" nur für **sehr allgemeine** Styles, sind weniger spezifisch als Klassen.
- **h1, .main-headline**: Ein Rule Set kann mehrere kommagetrennte Selektoren haben.
- **.page-nav .page-nav\_\_link**: Mehrteilige Selektoren sind spezifischer. Hohe Spezifität ist schwer überschreibbar.
- **#register**: Ids haben höhere Spezifität als Klassen.

# CSS: Wichtige Konzepte

# CSS: Block- und Inline-Elemente

## Block-Elemente



Nehmen sich immer die ganze verfügbare Breite.

z. B. `h1`, `p`, `ul`, `li`, `div`

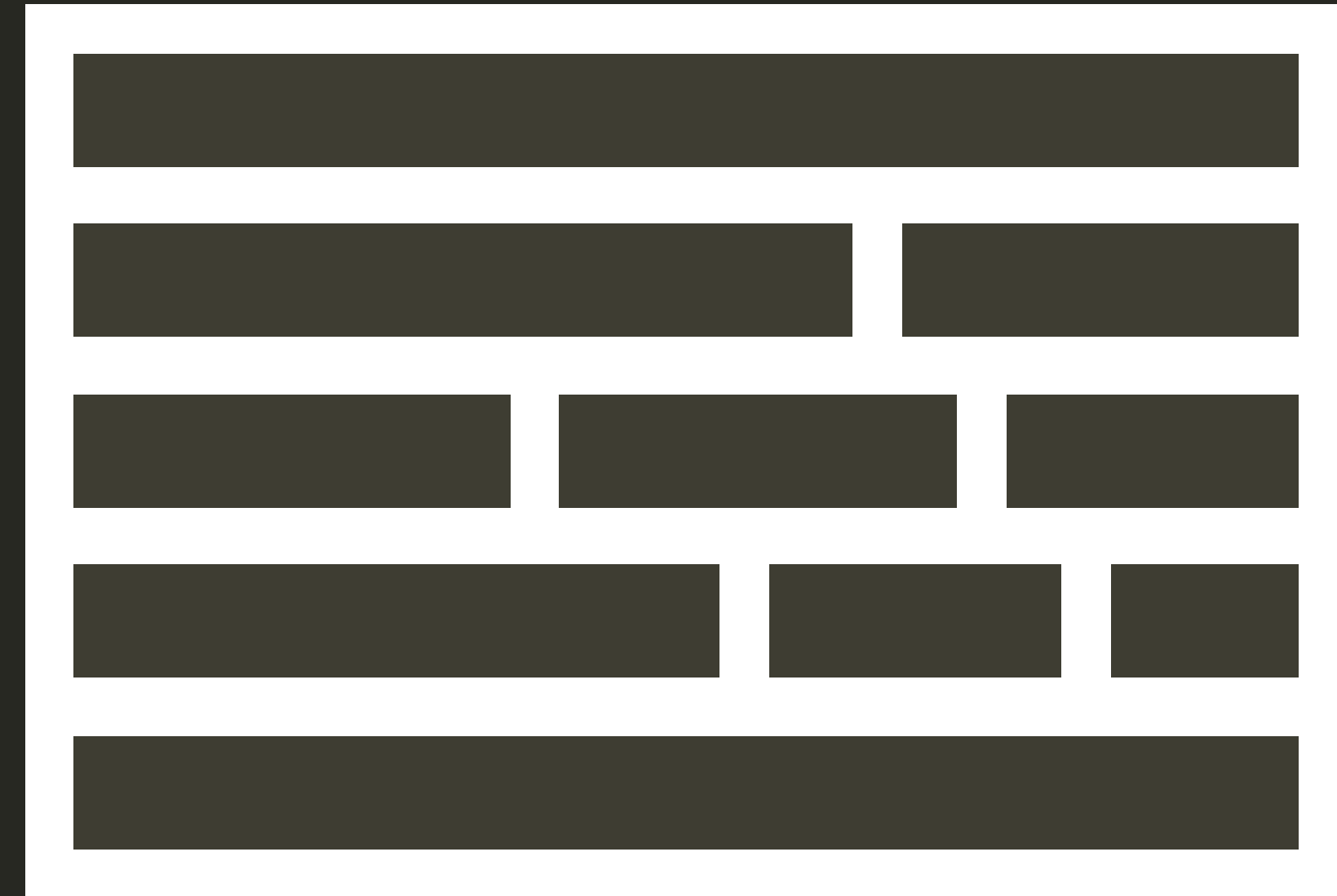
# CSS: Block- und Inline-Elemente

## Block-Elemente



Nehmen sich immer die ganze verfügbare Breite.  
z. B. **h1**, **p**, **ul**, **li**, **div**

## Inline-Elemente



Teilen sich eine Zeile.  
z. B. **a**, **em**, **img**, **span**

Demonstration: Block- und Inline-Elemente

# CSS: Box Model

# CSS: Box Model

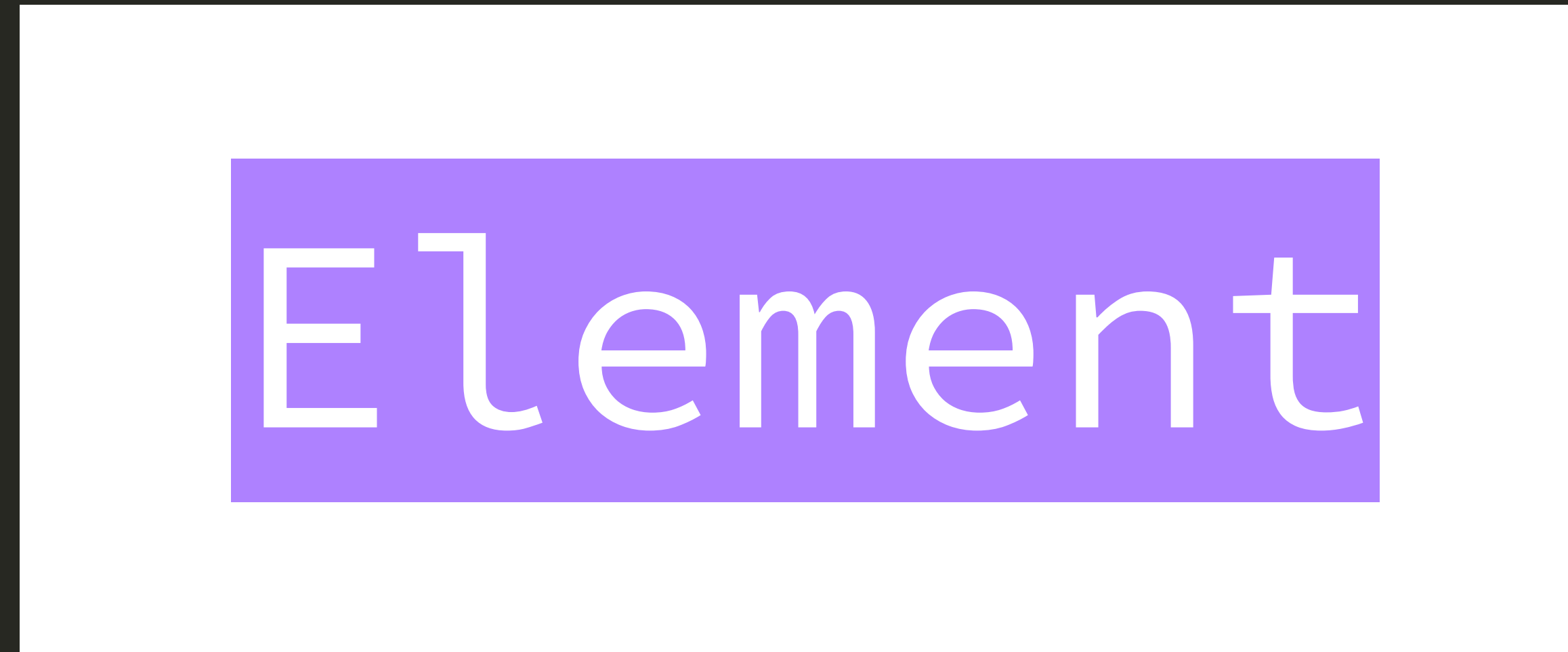
Element

Element



# CSS: Box Model

Element  
Padding

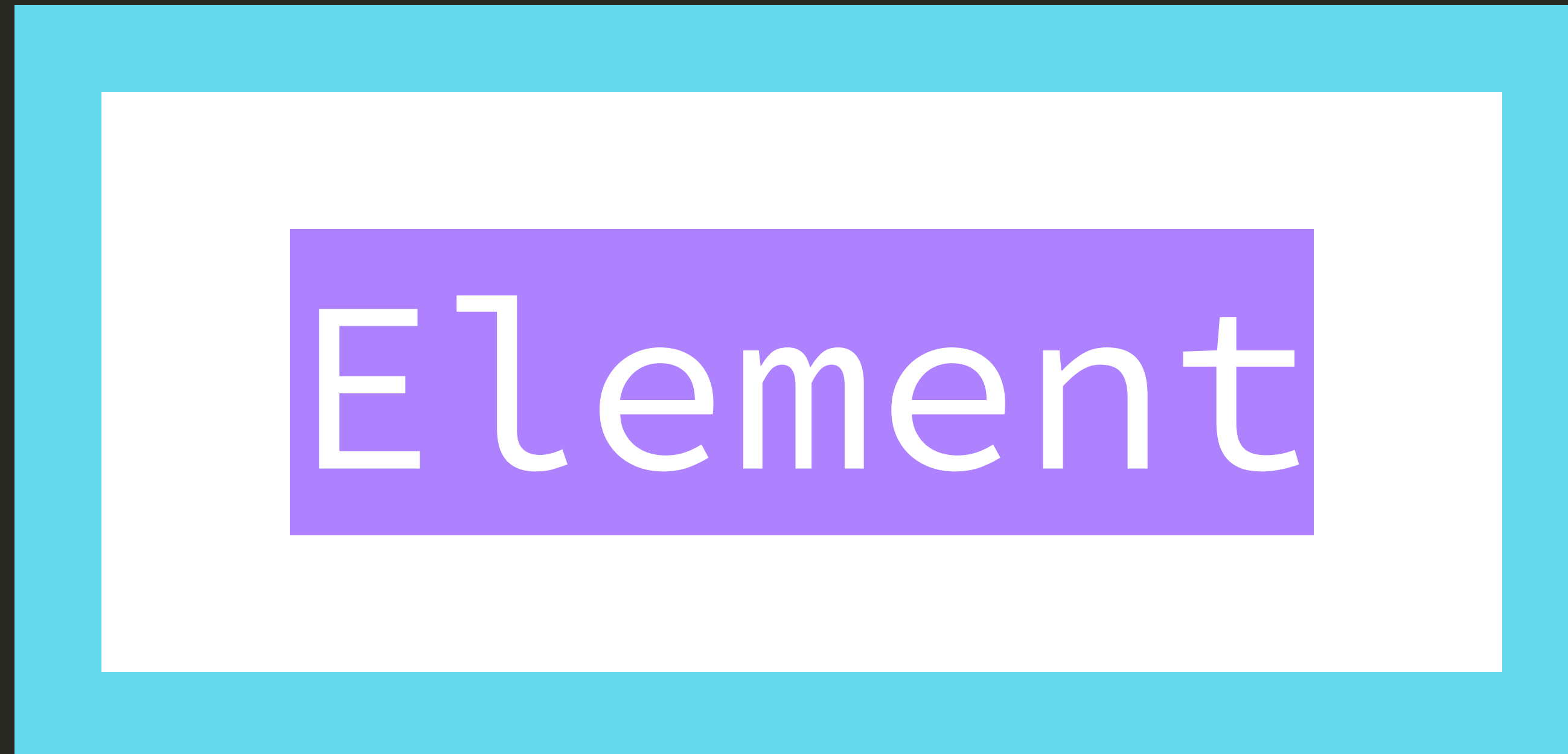


# CSS: Box Model

Element

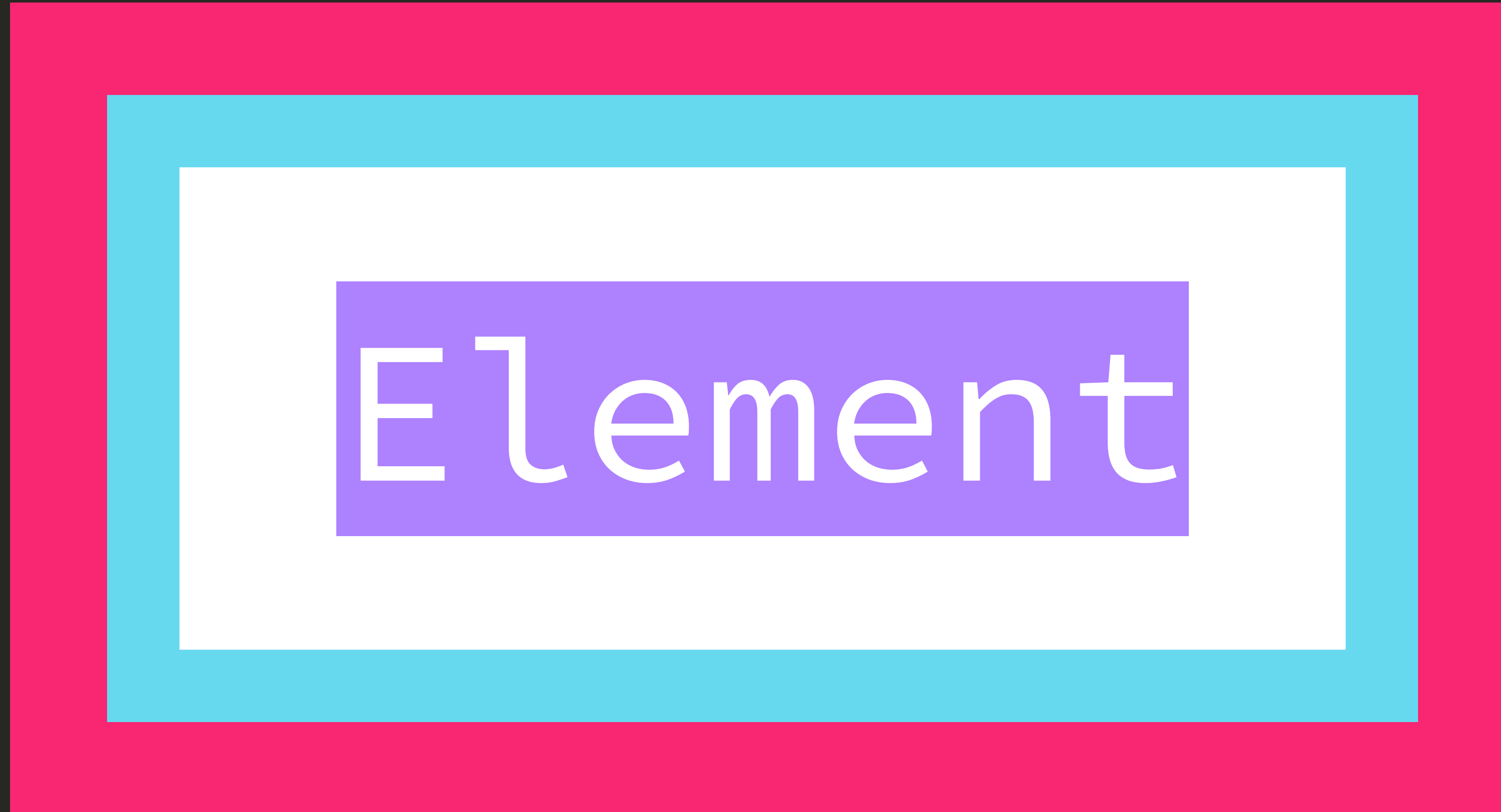
Padding

Border



# CSS: Box Model

Element  
Padding  
Border  
Margin



# Demonstration: Box Model

# CSS: Learnings Boxmodel

- Browser geben den Elementen Standard-Styles.
- `padding` wird auf die `width` und `height` draufgerechnet
- Inline-Elemente bekommen kein `margin-top` und `margin-bottom`.
- `margin-left: auto; margin-right: auto;` zentriert Block-Elemente horizontal.

# Border, Margin und Padding: Werte

```
margin: 5px 10px 20px 3px;
```



# Border, Margin und Padding: Werte

Oben

```
margin: 5px 10px 20px 3px;
```



# Border, Margin und Padding: Werte

Oben

Rechts

```
margin: 5px 10px 20px 3px;
```





# Border, Margin und Padding: Werte

Oben

Rechts

```
margin: 5px 10px 20px 3px;
```

Unten



# Border, Margin und Padding: Werte

Oben

Rechts

```
margin: 5px 10px 20px 3px;
```

Unten

Links



# Border, Margin und Padding: Werte

Oben

Rechts

```
margin: 5px 10px 20px 3px;
```

Unten

Links

```
margin-top: 5px;
```

```
margin-right: 10px;
```

```
margin-bottom: 20px;
```

```
margin-left: 3px;
```



# Border, Margin und Padding: Werte

# Border, Margin und Padding: Werte

```
margin: 5px 10px 20px 3px;
```

Oben

Rechts

Unten

Links

# Border, Margin und Padding: Werte

```
margin: 5px 10px 20px 3px;
```

Oben

Rechts

Unten

Links

```
margin: 5px 10px 20px 10px;
```

# Border, Margin und Padding: Werte

```
margin: 5px 10px 20px 3px;
```

Oben

Rechts

Unten

Links

```
margin: 5px 10px 20px 10px;
```

```
margin: 5px 10px 5px 10px;
```

# Border, Margin und Padding: Werte

```
margin: 5px 10px 20px 3px;
```

Oben

Rechts

Unten

Links

```
margin: 5px 10px 20px 10px;
```

```
margin: 5px 10px 5px 10px;
```

```
margin: 5px 5px 5px 5px;
```



# Interaktiv: CSS schreiben





# Weitere Ressourcen

- [css-tricks.com](https://css-tricks.com) – CSS Tutorials und Tipps

## Weitere Ressourcen

- [css-tricks.com](https://css-tricks.com) – CSS Tutorials und Tipps
- [developer.mozilla.org](https://developer.mozilla.org) – alles zu HTML und CSS

# Weitere Ressourcen

- [css-tricks.com](https://css-tricks.com) – CSS Tutorials und Tipps
- [developer.mozilla.org](https://developer.mozilla.org) – alles zu HTML und CSS
- [stackoverflow.com](https://stackoverflow.com) – Fragen & Antworten

## Weitere Ressourcen

- [css-tricks.com](https://css-tricks.com) – CSS Tutorials und Tipps
- [developer.mozilla.org](https://developer.mozilla.org) – alles zu HTML und CSS
- [stackoverflow.com](https://stackoverflow.com) – Fragen & Antworten
- [shop.smashingmagazine.com](https://shop.smashingmagazine.com) – Smashing Library

## Weitere Ressourcen

- [css-tricks.com](https://css-tricks.com) – CSS Tutorials und Tipps
- [developer.mozilla.org](https://developer.mozilla.org) – alles zu HTML und CSS
- [stackoverflow.com](https://stackoverflow.com) – Fragen & Antworten
- [shop.smashingmagazine.com](https://shop.smashingmagazine.com) – Smashing Library
- [abookapart.com](https://abookapart.com) – Bücher zu HTML5, CSS3 und mehr

## Weitere Ressourcen

- [css-tricks.com](https://css-tricks.com) – CSS Tutorials und Tipps
- [developer.mozilla.org](https://developer.mozilla.org) – alles zu HTML und CSS
- [stackoverflow.com](https://stackoverflow.com) – Fragen & Antworten
- [shop.smashingmagazine.com](https://shop.smashingmagazine.com) – Smashing Library
- [abookapart.com](https://abookapart.com) – Bücher zu HTML5, CSS3 und mehr
- [codepen.io](https://codepen.io) – Online Frontend Code schreiben und teilen



Das ist CSS. Fragen?

# Agenda

- ✓ Einführung
- ✓ HTML-Grundlagen
- ✓ CSS-Grundlagen
- Bootstrap
- Ausblick



Bootstrap: Was ist das?

# Was ist Bootstrap?



The most popular front-end framework for developing responsive, mobile first projects on the web.

# Was ist Bootstrap?

„Millions of amazing sites across the web are being built with Bootstrap.“



The most popular front-end framework for developing responsive, mobile first projects on the web.

# Was ist Bootstrap?

„Millions of amazing sites across the web are being built with Bootstrap.“

Sammlung von HTML, CSS und JS



The most popular front-end framework for developing responsive, mobile first projects on the web.

# Was ist Bootstrap?

„Millions of amazing sites across the web are being built with Bootstrap.“

Sammlung von HTML, CSS und JS



The most popular front-end framework for developing responsive, mobile first projects on the web.

NEU in Bootstrap 3





# Wie funktioniert Bootstrap?

- **Verknüpfung:** von Bootstrap-CSS sowie -JS mit dem eigenen HTML

# Wie funktioniert Bootstrap?

- **Verknüpfung:** von Bootstrap-CSS sowie -JS mit dem eigenen HTML
- **Automatisches Styling:** von Typographie und Links

# Wie funktioniert Bootstrap?

- **Verknüpfung:** von Bootstrap-CSS sowie -JS mit dem eigenen HTML
- **Automatisches Styling:** von Typographie und Links
- **Spezielles Styling:** über das Einfügen von Klassen

Demonstration: Das ist Bootstrap.

# Bootstrap: CSS und Components

CSS	Components	
Typography	Glyphicons	Badges
Code	Dropdowns	Jumbotron
Tables	Button groups	Page header
Forms	Button dropdowns	Thumbnails
Buttons	Input groups	Alerts
Images	Navs	Progress bars
Helper Classes	Navbar	Media object
Responsive Utilities	Breadcrumbs	List group
	Paginations	Panels
	Labels	Wells

# Bootstrap: das Grid

# Bootstrap: Grid



# Bootstrap: Grid

- Grid-System für Adaptive und Responsive Web Design

# Bootstrap: Grid

- Grid-System für Adaptive und Responsive Web Design
- Ist standardmäßig mobile first („scale up“)

# Bootstrap: Grid

- Grid-System für Adaptive und Responsive Web Design
- Ist standardmäßig mobile first („scale up“)
- Funktioniert über verschachtelte Klassen

# Bootstrap: Grid

## 1. Klasse „Container“

`.container` (fixed-width für Adaptive-Verhalten)  
oder `.container-fluid` (full-width für Fluid-Verhalten)

# Bootstrap: Grid

## 1. Klasse „Container“

`.container` (fixed-width für Adaptive-Verhalten)  
oder `.container-fluid` (full-width für Fluid-Verhalten)

```
<div class="container-fluid">  
</div>
```

# Bootstrap: Grid

## 2. Klasse „Row“

um Elemente nebeneinander anzuordnen

# Bootstrap: Grid

## 2. Klasse „Row“

um Elemente nebeneinander anzuordnen

```
<div class="container-fluid">  
  <div class="row">  
  </div>  
</div>
```

# Bootstrap: Grid

## 3. Klasse „Column“

für die Definition der Spaltenbreite, basierend auf einem 12-spaltigen Grid-System



# Bootstrap: Grid

## 3. Klasse „Column“

für die Definition der Spaltenbreite, basierend auf einem 12-spaltigen Grid-System

```
<div class="container-fluid">  
  <div class="row">  
    <div class="col-md-6">Halbe Reihe</div>  
    <div class="col-md-6">Halbe Reihe</div>  
  </div>  
</div>
```

# Bootstrap: Grid

```
<div class="container-fluid">
```

1. Container

```
<div class="row">
```

2. Row

```
<div class="col-md-6">Halbe Reihe</div>
```

3. Columns

```
<div class="col-md-6">Halbe Reihe</div>
```

```
</div>
```

```
</div>
```

# Bootstrap: Grid

Halbe Reihe

Halbe Reihe

```
<div class="container-fluid">
```

1. Container

```
<div class="row">
```

2. Row

```
<div class="col-md-6">Halbe Reihe</div>
```

3. Columns

```
<div class="col-md-6">Halbe Reihe</div>
```

```
</div>
```

```
</div>
```

# Bootstrap Grid: Columns

Extra small devices Phones <768px	Small devices Tablets ≥768px	Medium devices Desktops ≥992px	Large devices Desktops ≥1200px
.col-xs-	.col-sm-	.col-md-	.col-lg-

# Bootstrap Grid: Columns

Extra small devices Phones <768px	Small devices Tablets ≥768px	Medium devices Desktops ≥992px	Large devices Desktops ≥1200px
.col-xs-	.col-sm-	.col-md-	.col-lg-

Eine Grid-Größe gilt immer für diese Größe und darüber:  
**sm** gilt für **sm**, **md** und **lg**.

# Bootstrap Grid: Columns mit Versatz

4 Spalten

4 Spalten

# Bootstrap Grid: Columns mit Versatz

4 Spalten

4 Spalten


```
<div class="container-fluid">  
  <div class="row">  
    <div class="col-md-4">4 Spalten</div>  
    <div class="col-md-4 col-md-offset-4">4 Spalten</div>  
  </div>  
</div>
```

# Bootstrap Grid: Columns mit Versatz

4 Spalten

4 Spalten

```
<div class="container-fluid">  
  <div class="row">  
    <div class="col-md-4">4 Spalten</div>  
    <div class="col-md-4 col-md-offset-4">4 Spalten</div>  
  </div>  
</div>
```





# Responsive und Mobile first: nicht heute

- Bootstrap ist von Haus aus responsive.
- Spezielle Elemente:
  - Responsive tables
  - Responsive images
  - Responsive Utilities  
(Sichtbarkeit toggeln)



# Interaktiv: Bootstrap kennenlernen

# Bootstrap: Loslegen in 6 Schritten

- Schritt 1: Download unter <http://getbootstrap.com/>

# Bootstrap: Loslegen in 6 Schritten

- **Schritt 1:** Download unter <http://getbootstrap.com/>

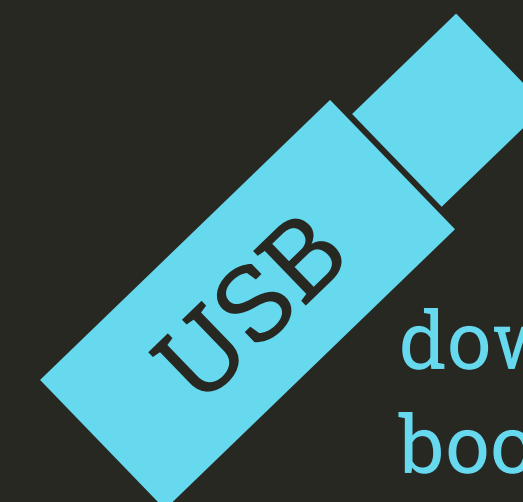
<h2>Bootstrap</h2> <p>Compiled and minified CSS, JavaScript, and fonts. No docs or original source files are included.</p> <p>Download Bootstrap</p> <p>Currently v3.1.1.</p>	<h2>Source code</h2> <p>Source Less, JavaScript, and font files, along with our docs. <b>Requires a Less compiler and some setup.</b></p> <p>Download source</p>	<h2>Sass</h2> <p><a href="#">Bootstrap ported from Less to Sass</a> for easy inclusion in Rails, Compass, or Sass-only projects.</p> <p>Download Sass</p>
---	--	---

# Bootstrap: Loslegen in 6 Schritten

- **Schritt 1:** Download unter <http://getbootstrap.com/>

Bootstrap	Source code	Sass
Compiled and minified CSS, JavaScript, and fonts. No docs or original source files are included.	Source Less, JavaScript, and font files, along with our docs. <b>Requires a Less compiler and some setup.</b>	<a href="#">Bootstrap ported from Less to Sass</a> for easy inclusion in Rails, Compass, or Sass-only projects.
<a href="#">Download Bootstrap</a>	<a href="#">Download source</a>	<a href="#">Download Sass</a>
Currently v3.1.1.		

Dieses, bitte!



downloads /  
bootstrap-3.1.1-dist

# Bootstrap: Loslegen in 6 Schritten

- **Schritt 1:** Download unter <http://getbootstrap.com/>

```
▼ bootstrap-3.1.1-dist
  ▼ css
    bootstrap-theme.css
    bootstrap-theme.css.map
    bootstrap-theme.min.css
    bootstrap.css
    bootstrap.css.map
    bootstrap.min.css
  ▼ fonts
    glyphicons-halflings-regular.eot
    glyphicons-halflings-regular.svg
    glyphicons-halflings-regular.ttf
    glyphicons-halflings-regular.woff
  ▼ js
    bootstrap.js
    bootstrap.min.js
```

# Bootstrap: Loslegen in 6 Schritten

- Schritt 1: Download
- **Schritt 2:** Dateistruktur anlegen



# Bootstrap: Loslegen in 6 Schritten

- Schritt 1: Download
- **Schritt 2:** Dateistruktur anlegen

getbootstrap.com



css/bootstrap.css

fonts

js/bootstrap.js



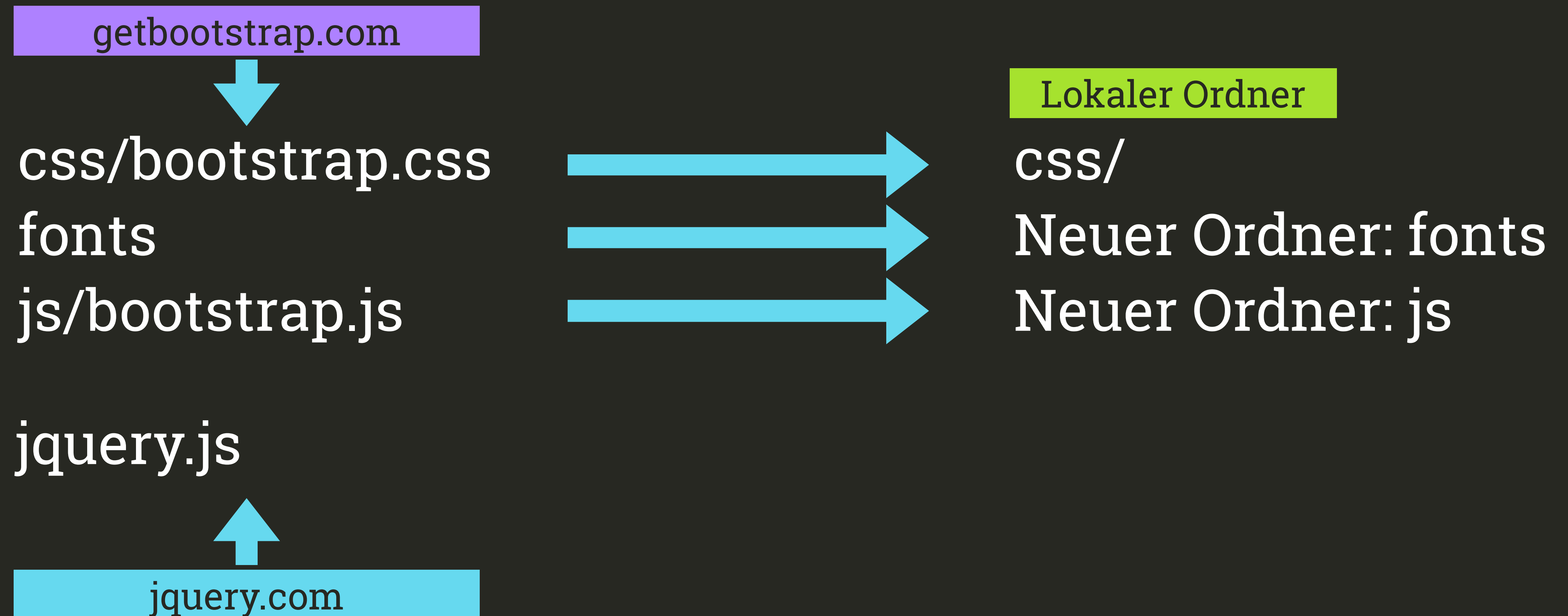
# Bootstrap: Loslegen in 6 Schritten

- Schritt 1: Download
- **Schritt 2:** Dateistruktur anlegen



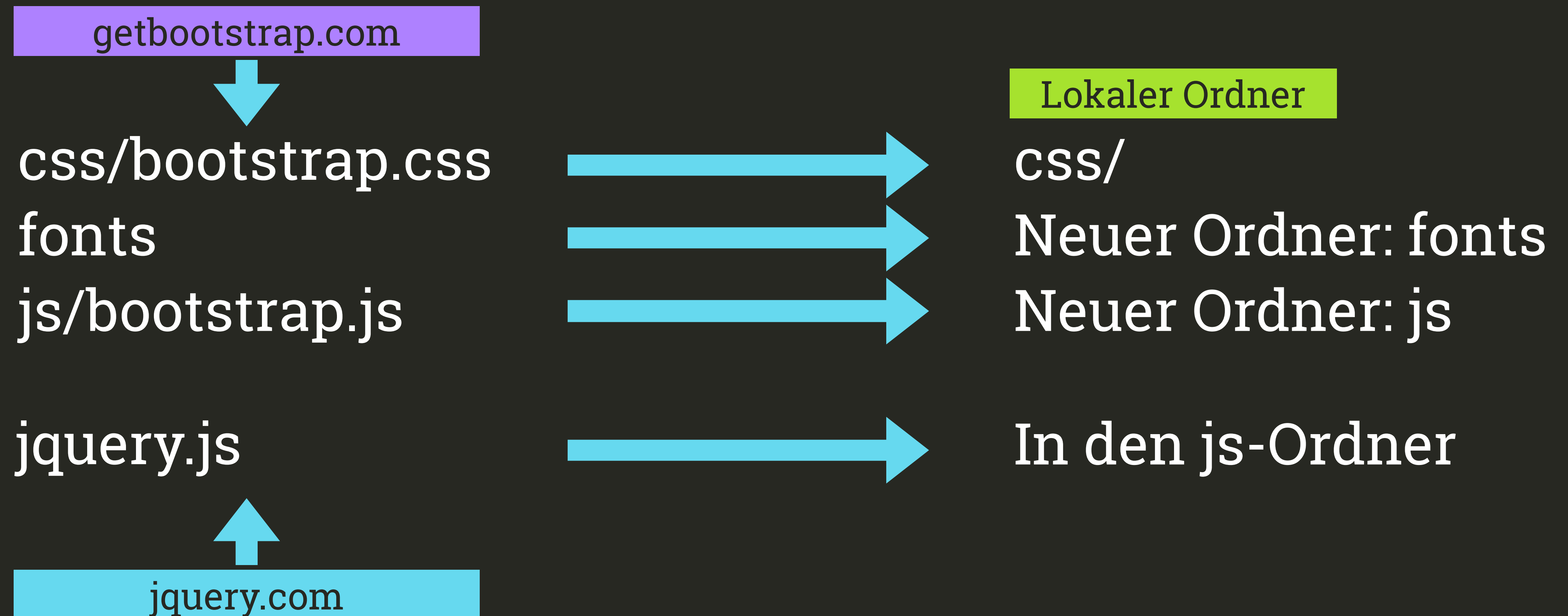
# Bootstrap: Loslegen in 6 Schritten

- Schritt 1: Download
- **Schritt 2:** Dateistruktur anlegen



# Bootstrap: Loslegen in 6 Schritten

- Schritt 1: Download
- **Schritt 2:** Dateistruktur anlegen



# Bootstrap: Loslegen in 6 Schritten

- Schritt 1: Download
- **Schritt 2:** Dateistruktur anlegen



getbootstrap.com



css/bootstrap.css

fonts

js/bootstrap.js

jquery.js



jquery.com

Lokaler Ordner

css/

Neuer Ordner: fonts

Neuer Ordner: js

In den js-Ordner

# Bootstrap: Loslegen in 6 Schritten

- Schritt 1: Download
- Schritt 2: Dateistruktur anlegen
- **Schritt 3: Dateien verknüpfen**

# Bootstrap: Loslegen in 6 Schritten

- Schritt 1: Download
- Schritt 2: Dateistruktur anlegen
- **Schritt 3:** Dateien verknüpfen: (a) CSS

bootstrap.css im **head**  
VOR dem eigenen CSS

```
<!-- Bootstrap -->
<link rel="stylesheet" href="css/bootstrap.css">

<!-- Self-written styles -->
<link rel="stylesheet" href="css/styles.css">
</head>
```

# Bootstrap: Loslegen in 6 Schritten

- Schritt 1: Download
- Schritt 2: Dateistruktur anlegen
- **Schritt 3:** Dateien verknüpfen: (b) JS

.js am Ende des **body**s

```
</footer>

<!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
<script src="js/jquery-2.1.1.js"></script>

<!-- Include all compiled plugins (below), or include individual files as
needed -->
<script src="js/bootstrap.js"></script>
</body>
```



# Bootstrap: Loslegen in 6 Schritten

- Schritt 1: Download
- Schritt 2: Dateistruktur anlegen
- Schritt 3: Dateien verknüpfen
- **Schritt 4: Layout-Elemente einfügen**

z. B. Header, Primärnavigation,  
Hauptcontent, Marginalspalte  
und Footer

```
<body>
  <header class="main-header">
    <nav class="main-nav">
    </nav>
  </header>

  <main class="main">
    <section>
    </section>

    <aside class="main-aside">
    </aside>
  </main>

  <footer class="main-footer">
  </footer>
</body>
```



# Erforderlich: Viewport Meta Tag im head

```
<meta name="viewport" content="width=device-width,  
initial-scale=1">
```

Ermöglicht Responsive Web Design  
auf mobilen Geräten

# Bootstrap: Loslegen in 6 Schritten

- Schritt 1: Download
- Schritt 2: Dateistruktur anlegen
- Schritt 3: Dateien verknüpfen
- Schritt 4: Layout-Elemente einfügen
- **Schritt 5: Grid anlegen**

# Bootstrap: Loslegen in 6 Schritten

- Schritt 1: Download
- Schritt 2: Dateistruktur anlegen
- Schritt 3: Dateien verknüpfen
- Schritt 4: Layout-Elemente einfügen
- **Schritt 5:** Grid anlegen  
mit Hilfe der Bootstrap-Doku

# Bootstrap: Loslegen in 6 Schritten

- Schritt 1: Download
- Schritt 2: Dateistruktur anlegen
- Schritt 3: Dateien verknüpfen
- Schritt 4: Layout-Elemente einfügen
- Schritt 5: Grid anlegen
- **Schritt 6: Inhaltselemente anlegen und stylen**

# Bootstrap: Loslegen in 6 Schritten

- Schritt 1: Download
- Schritt 2: Dateistruktur anlegen
- Schritt 3: Dateien verknüpfen
- Schritt 4: Layout-Elemente einfügen
- Schritt 5: Grid anlegen
- **Schritt 6:** Inhaltselemente anlegen und stylen  
mit Hilfe der Code-Snippets aus der Bootstrap-Doku

# Bootstrap: Loslegen in 6 Schritten

- Schritt 1: Download
- Schritt 2: Dateistruktur anlegen
- Schritt 3: Dateien verknüpfen
- Schritt 4: Layout-Elemente einfügen
- Schritt 5: Grid anlegen
- Schritt 6: Inhaltselemente anlegen und stylen

# Bootstrap: Grid anlegen für main



# Bootstrap: Grid anlegen für main

- **container** und **row** sind bereits da





# Bootstrap: Grid anlegen für main

- **container** und **row** sind bereits da
- Section „Jump Marks“ und „Articles“: **col-sm-7**



# Bootstrap: Grid anlegen für main

- **container** und **row** sind bereits da
- Section „Jump Marks“ und „Articles“: **col-sm-7**
- Aside: **col-sm-4** und **col-sm-offset-1**



# Bootstrap: Inhaltselemente stylen

# Bootstrap: Inhaltselemente stylen

- Link „Einreichungsformular“: grüner Button
- Link „Tickets kaufen“: größer, orangener Button

# Bootstrap: Inhaltselemente stylen

- Link „Einreichungsformular“: grüner Button  
Link „Tickets kaufen“: größer, orangener Button
- Meta nav: „list-inline“ anordnen und „pull-right“ ausrichten

# Bootstrap: Inhaltselemente stylen

- Link „Einreichungsformular“: grüner Button  
Link „Tickets kaufen“: größer, orangener Button
- Meta nav: „list-inline“ anordnen und „pull-right“ ausrichten
- Main navigation: gewünschte „nav“ hinzufügen (z. B. „Pills“), ausrichten („justified“), größere Schrift (z. B. „lead“) „IAK 2014“ ist „active“

# Bootstrap: Inhaltselemente stylen

- Link „Einreichungsformular“: grüner Button  
Link „Tickets kaufen“: größer, orangener Button
- Meta nav: „list-inline“ anordnen und „pull-right“ ausrichten
- Main navigation: gewünschte „nav“ hinzufügen (z. B. „Pills“), ausrichten („justified“), größere Schrift (z. B. „lead“) „IAK 2014“ ist „active“
- Footer-Link: rechtsausgerichtetes Icon

Inhaltselemente: weitere Stylings



# Inhaltselemente: weitere Stylings

- Liste in Main aside: ohne Bulletpoints („unstyled“)

# Inhaltselemente: weitere Stylings

- **Liste in Main aside:** ohne Bulletpoints („unstyled“)
- **Jump Marks:** kleinere Textgröße

# Inhaltselemente: weitere Stylings

- **Liste in Main aside:** ohne Bulletpoints („unstyled“)
- **Jump Marks:** kleinere Textgröße
- **Headlines in Articles und Aside:** kleinere Textgröße

Bootstrap: Was noch zu sagen ist.

# Demonstration: JavaScript mit Bootstrap



# Warum Bootstrap?

- Gute Dokumentation

# Warum Bootstrap?

- Gute Dokumentation
- Große Community



# Warum Bootstrap?

- Gute Dokumentation
- Große Community
- Viele UI-Elemente

# Warum Bootstrap?

- Gute Dokumentation
- Große Community
- Viele UI-Elemente
- Responsive Web Design als Standard

# Warum Bootstrap?

- Gute Dokumentation
- Große Community
- Viele UI-Elemente
- Responsive Web Design als Standard
- Individuell anpassbar ("Customize")

Unser erstes Boilerplate: für euch



[github.com/developinguxd/boilerplates](https://github.com/developinguxd/boilerplates)

Das ist Bootstrap. Noch Fragen?

# Agenda

- ✓ Einführung
- ✓ HTML-Grundlagen
- ✓ CSS-Grundlagen
- ✓ Bootstrap
- Ausblick



# Tipps: die tägliche Arbeit



# Tipps: die tägliche Arbeit

- **Start:** Handscribbles

# Tipps: die tägliche Arbeit

- **Start:** Handscribbles
- **Entscheidung:** Ist die Projektart geeignet?

# Tipps: die tägliche Arbeit

- **Start:** Handscribbles
- **Entscheidung:** Ist die Projektart geeignet?
- **Doing:** Prototyping gemeinsam durch Kreation und Frontend-Entwicklung

# Tipps: die tägliche Arbeit

- **Start:** Handscribbles
- **Entscheidung:** Ist die Projektart geeignet?
- **Doing:** Prototyping gemeinsam durch Kreation und Frontend-Entwicklung
- **Ergänzende Dokumentation:** via Screenshots des Prototypen im gewohnten Programm

Weiterlernen: viele wichtige Themen

# Weiterlernen: viele wichtige Themen

HTML

CSS

Bootstrap

DevTools

# Weiterlernen: viele wichtige Themen

HTML

CSS

Bootstrap

DevTools

holder.js

jQuery

JavaScript

Sass

# Weiterlernen: viele wichtige Themen

HTML

CSS

Bootstrap

DevTools

holder.js

jQuery

JavaScript

Sass

GitHub

CodeKit

Webfonts

CodePen



A high-contrast, black and white close-up portrait of Steve Jobs. He is looking directly at the camera with a slight, thoughtful expression. He has a short beard and mustache, and is wearing his signature round, thin-rimmed glasses. The lighting is dramatic, with deep shadows on the right side of his face and forehead, and highlights on the left side of his face and nose. The background is dark and out of focus.

# STEVE JOBS

The Lost Interview

“I think everybody in this country should learn how to program a computer because it teaches you how to think.”



Developing UXD: Unsere Idee



# Developing UXD: Unsere Idee

# Developing UXD: Unsere Idee

- Fest definiertes, gemeinsames Toolset

# Developing UXD: Unsere Idee

- Fest definiertes, gemeinsames Toolset
- Moderne, erprobte Standards

# Developing UXD: Unsere Idee

- Fest definiertes, gemeinsames Toolset
- Moderne, erprobte Standards
- Vorauswahl der besten Lernressourcen

# Developing UXD: Unsere Idee

- Fest definiertes, gemeinsames Toolset
- Moderne, erprobte Standards
- Vorauswahl der besten Lernressourcen
- Lebendige Plattform



# Developing UXD: Unsere Idee

- Fest definiertes, gemeinsames Toolset
- Moderne, erprobte Standards
- Vorauswahl der besten Lernressourcen
- Lebendige Plattform
- Gemeinsam und Open Source

# Developing UXD: Unsere Idee

- Fest definiertes, gemeinsames Toolset
- Moderne, erprobte Standards
- Vorauswahl der besten Lernressourcen
- Lebendige Plattform
- Gemeinsam und Open Source
- Eigenentwicklungen

# Developing UXD: Unsere Idee

- Fest definiertes, gemeinsames Toolset
- Moderne, erprobte Standards
- Vorauswahl der besten Lernressourcen
- Lebendige Plattform
- Gemeinsam und Open Source
- Eigenentwicklungen
- Unterstützung durch Frontend-Community





developinguxd.com

@developinguxd

facebook.com/developinguxd

mail@developinguxd.com

# Developing UXD: Nächste Schritte

# Developing UXD: Nächste Schritte

- Die besten Lernquellen im Web



# Developing UXD: Nächste Schritte

- Die besten Lernquellen im Web
- Lernpläne

# Developing UXD: Nächste Schritte

- Die besten Lernquellen im Web
- Lernpläne
- Aufbau einer Community begeisterter UX-Designer

# Developing UXD: Nächste Schritte

- Die besten Lernquellen im Web
- Lernpläne
- Aufbau einer Community begeisterter UX-Designer
- Entwicklung von Tools für Workflow-Optimierungen

Vielen Dank! Happy coding!

# Euer Feedback: Wie wars?



developinguxd.com

@developinguxd

facebook.com/developinguxd

mail@developinguxd.com

Cornelia Erbs, Jeremias Dombrowsky  
Lessers Passage 10  
22767 Hamburg

+49 (0)151 23421980

mail@developinguxd.com  
<http://developinguxd.com/>

Alle Icons in dieser Präsentation kommen von  
<http://iconmonstr.com/>









Backlog: Sublime, HTML, CSS, JS

# Backlog

- **Sublime:** Shortcuts
- **Sublime:** Emmet Shortcuts
- **HTML:** Formulare
- **Holder.js:** Img placeholders
- **CSS:** BEM naming convention
- **jQuery und JavaScript:** Einbinden und laden

# Sublime Text 3: Emmet Shortcuts

# Sublime Text 3: Emmet Shortcuts

- **Selector-Syntax:** e.g. `article>h1+p` wird zu  
    `<article>`  
        `<h1></h1>`  
        `<p></p>`  
    `</article>`

# Sublime Text 3: Emmet Shortcuts

- **Selector-Syntax:** e.g. `article>h1+p` wird zu  
    `<article>`  
        `<h1></h1>`  
        `<p></p>`  
    `</article>`
- **Ctrl+w:** "Wrap with tag"

# Sublime Text 3: Emmet Shortcuts

- **Selector-Syntax:** e.g. `article>h1+p`Tab wird zu

```
<article>
  <h1></h1>
  <p></p>
</article>
```
- **Ctrl+w:** "Wrap with tag"
- **Cmd+Shift+k:** Tag ändern



# Sublime Text 3: Emmet Shortcuts

- **Selector-Syntax:** e.g. `article>h1+p` wird zu  
    `<article>`  
        `<h1></h1>`  
        `<p></p>`  
    `</article>`
- **Ctrl+w:** "Wrap with tag"
- **Cmd+Shift+k:** Tag ändern
- **Cmd+':** Element löschen

# HTML: Formulare

- **form**: Gruppiert zusammengehörige Formularelemente, darf auch andere Elemente enthalten.
- **input**: Kann von unterschiedlichem Typ sein: **text**, **email**, **number**, **tel**, **color**, **date**, **checkbox**, **radio**, ...
- **select**: Enthält **option**-Elemente, die ein Dropdown bilden.

# Sublime Text 3: Shortcuts

# Sublime Text 3: Shortcuts

- **TagnameTab**: Autocomplete (Mehrfach für Sprungmarken)

# Sublime Text 3: Shortcuts

- **TagnameTab**: Autocomplete (Mehrfach für Sprungmarken)
- **Cmd+p**: Fuzzy-Search für Dateien

# Sublime Text 3: Shortcuts

- **TagnameTab**: Autocomplete (Mehrfach für Sprungmarken)
- **Cmd+p**: Fuzzy-Search für Dateien
- **Cmd+Shift+p**: Zugriff auf alle Befehle

# Sublime Text 3: Shortcuts

- **TagNameTab**: Autocomplete (Mehrfach für Sprungmarken)
- **Cmd+p**: Fuzzy-Search für Dateien
- **Cmd+Shift+p**: Zugriff auf alle Befehle
- **Cmd+d**: Weitere Instanz des markierten Texts auswählen

# Sublime Text 3: Shortcuts

- **TagNameTab**: Autocomplete (Mehrfach für Sprungmarken)
- **Cmd+p**: Fuzzy-Search für Dateien
- **Cmd+Shift+p**: Zugriff auf alle Befehle
- **Cmd+d**: Weitere Instanz des markierten Texts auswählen
- Ausführliches Tutorial auf [scotch.io](https://scotch.io)



# Selektoren: BEM naming convention

# Selektoren: BEM naming convention

- **.page-header**: Einfacher Bindestrich bei zusammengesetzten Namen

# Selektoren: BEM naming convention

- **.page-header**: Einfacher Bindestrich bei zusammengesetzten Namen
- **.page-nav, .page-nav\_\_link**: Unterelemente eines "Moduls" haben den Modulnamen als Prefix.

# Selektoren: BEM naming convention

- **.page-header**: Einfacher Bindestrich bei zusammengesetzten Namen
- **.page-nav, .page-nav\_\_link**: Unterelemente eines "Moduls" haben den Modulnamen als Prefix.
- **.button, .button--small, .button--cancel**: Varianten sind als solche erkennbar.

# Holder.js: Img placeholders

# Holder.js: Img placeholders

- **Download:** <http://imsky.github.io/holder/>

# Holder.js: Img placeholders

- Download: <http://imsky.github.io/holder/>
- JavaScript im `<body>` laden:  
`<script src="js/holder.js"></script>`

# Holder.js: Img placeholders

- **Download:** <http://imsky.github.io/holder/>
- **JavaScript im <body> laden:**  
`<script src="js/holder.js"></script>`
- **Einsatz:**  
``  
``  
``  
``



Backlog: Bootstrap

# Bootstrap UI Editors

- Eine Reihe von Editoren: für unterschiedliche Ansprüche
- Siehe: <http://thedesignblitz.com/the-best-bootstrap-ui-editors/>

BootSwatchr

{wrap}bootstrap



I ♥  
BOOTSTRAP

Lavish

Bootstrap ThemeRoller



LayoutIt!

Pingendo

Kickstrap

Bootply

X-editable

 jetstrap

 DIVSHOT

PaintStrap



# Bootstrap: und andere Frameworks

- Bootstrap <http://twitter.github.io/bootstrap/>  
weit verbreitet, umfangreich
- Foundation <http://foundation.zurb.com/>  
sehr umfangreich, komplex
- Pure <http://purecss.io/> sehr schlank
- HTML Kickstart <http://www.99lime.com/elements/>  
umfangreich, leicht

Backlog: Argumentationshilfe

# Backlog: Argumentationshilfe

- Vorteile von HTML-Wireframes: für Kunden
- Vorteile von HTML-Wireframes: für Agenturen
- Grenzen des statischen Designs überwinden
- Wissen aufbauen
- Zusammenarbeit verbessern

## Vorteile: für Kunden

- Vermitteln Funktionalität einer Website von Anfang an realistisch
- Sind durch Interaktivität einfacher zu verstehen
- Produkt entwickelt sich vor den Augen des Kunden stetig weiter
- Kein Bruch zwischen Kurations- und Entwicklungsphase (keine Überraschungen)

Kunde

## Vorteile: für Agenturen 1/2

- Keine bzw. weniger Spezifikationen erforderlich
- Integrierte Arbeitsweise wird gefördert
- Animationen und Interaktivität werden direkt berücksichtigt
- Ansätze können direkt ausprobiert und dadurch Design-Entscheidungen leichter getroffen werden

UX

## Vorteile: für Agenturen 2/2

- Frühe Usertests sind möglich
- Browser-Spezifika funktionieren automatisch (Scrollbars, Button-Größen, Hover etc.)
- Code kann in Teilen weiter verwendet werden

UX



# Grenzen des statischen Designs überwinden

- Liquid und responsive Layouts
- Realistische Darstellung von Typografie
- Globale Anpassungen von Farben, Abständen, Größen
- Animationen und Interaktionen



# Wissen aufbauen

- Neue Denkweisen trainieren
- Neue Browserfeatures besser verstehen
- Wissen, was möglich ist
- Änderungen auf Live-Websites direkt ausprobieren (Developer Tools)
- Prototypen selbst bauen (z. B. mit Bootstrap) oder ändern/erweitern

# Zusammenarbeit verbessern

- Besseres Verständnis für das Medium und für Frontend-Developer
- Weniger Abhängigkeit von Frontend-Developern bei der Ideenfindung
- Implementierungsaufwände besser einschätzen können
- Gemeinsames (richtiges) Vokabular aufbauen
- Gemeinsam an Prototypen arbeiten

Kreation

Frontend

developinguxd.com

@developinguxd

facebook.com/developinguxd

mail@developinguxd.com