

# BOOTSTRAP 4

## EINFÜHRUNG UND THEMING

# EINFÜHRUNGSTHEMEN

- **Einstieg in Bootstrap 4**  
Ressourcen und Download  
Quellen für Themes  
Browser-Kompatibilität  
Layout mit CSS Klassen  
Erweiterungen und Plugins
- **Umgang mit dem Grid-System**  
Gestaltungsprinzipien und  
12er-Raster  
Zwischenräume  
Automatisches Layout  
Responsive Klassen  
Ausrichtung und Sortierung  
Verschachtelte Darstellungen  
Flexbox horizontal und vertikal  
Suchmaschinen optimierter  
HTML-Code
- **Typografie**  
Größen einstellen  
Farbschemata anwenden  
Abstände  
Textausrichtung  
Schatten und Rahmen  
Sichtbarkeit und Textersetzung
- **Typische Inhalte verwenden**  
Texte und Überschriften  
Bilder und Medien-Elemente  
Formulare und Tabellen  
Icon Sets verwenden
- **Navigationen erstellen**  
Responsive Navbars  
Menüpunkte als Dropdown  
Tabbed Navigation
- **Komponenten verwenden**  
Alerts, Badges und Buttons  
Listen  
Cards und Jumbotron
- **Interaktive Komponenten**  
Pagination  
Progress  
Scroll Spy  
Modal Boxen, Popup-Fenster  
Karussell, Slider  
Akkordeon  
Tool-Tipps
- **Bootstrap und JavaScript**  
jQuery und popper.js  
Data Attribute, data-toggle  
Events und asynchrone  
Funktionen  
Defaults überschreiben  
Methoden der Komponenten
- **Theming**  
Überschreiben der CSS-  
Klassen  
Eigenes Bootstrap-CSS  
erstellen mit Online-Tools

# AUFBAUTHemen

- **Einführung**

Übersicht über die Projektstruktur  
npm Befehle einsetzen  
Einen eigenen Build erstellen  
Dateigrößen optimieren  
QUnit Test ausführen

- **SCSS Einführung**

Umwandlung von SCSS in CSS  
Verwendung von Variablen  
Verschachtelte Klassen  
Stile auslagern und importieren  
Mixins erstellen  
Erweitern und erben von Eigenschaften  
Operatoren anwenden  
@Rules und Direktiven  
Bedingte Stile  
Maps und Schleifen

- **Bootstrap und SCSS**

Variablen für Farben und Grautöne  
Variablen für Media Queries  
Wichtige SCSS Funktionen  
Bootstrap Optionen ändern

- **Anwendungsfälle**

Eigene Farben verwenden  
Default Variablen überschreiben  
Spaltenanzahl und Abstände im Grid ändern  
Media Queries ändern  
Eigene Komponenten erstellen



Richte deine Arbeitsumgebung ein.

Lies dazu das Readme auf <https://github.com/zenbox/workshop/> und folge den Anweisungen.

```
workshop/
├── node_modules/
│   └── bootstrap
│       ├── dist
│       ├── js
│       └── scss
├── assets
│   ├── css
│   └── scss
│       └── main.scss
├── index.html
├── package-lock.json
└── package.json
```

15 MIN



Richte deine Arbeitsumgebung ein.

Lies dazu das Readme auf <https://github.com/zenbox/workshop/> und folge den Anweisungen.

```
workshop/
├── node_modules/
│   └── bootstrap
│       ├── dist
│       ├── js
│       └── scss
├── assets
│   ├── css
│   └── scss
│       └── main.scss
├── index.html
├── package-lock.json
└── package.json
```

15 MIN

# LEARNING SASS BASICS

# IMPORTS

```
// _reset.scss
html, body, ul, ol {
  margin: 0;
  padding: 0;
}
```

```
// main.scss
@import 'reset';
```

```
$ sass --watch assets/scss:assets/css
-> main.css
```

# VARIABLES

```
// LAYOUT  
$default-padding: 0.5rem;  
$default-margin: 0;
```

```
// FONTS  
$font-size: 22px;  
$line-height: 1.8rem;
```

```
// COLORS  
$light: #ffe;  
$dark: #000;
```



# MIXINGS

```
@mixin transform($property) {  
  -webkit-transform: $property;  
  -ms-transform: $property;  
  transform: $property;  
}
```


```
.box { @include transform(rotate(30deg)); }
```

# MIXINGS II

```
@mixin corner-icon($name, $top-or-bottom, $left-or-right) {  
  .icon-#{$name} {  
    background-image: url("/icons/#{$name}.svg");  
    position: absolute;  
    #{ $top-or-bottom }: 0;  
    #{ $left-or-right }: 0;  
  }  
}  
  
@include corner-icon("mail", top, left);  
  
.icon-mail { background-image: url("/icons/mail.svg"); }
```

# NESTING

```
nav {  
  ul {  
    margin: 0;  
    padding: 0;  
    list-style: none;  
  }  
  li {display:inline-block;}  
  a {  
    display: block;  
    padding: 6px 12px;  
    text-decoration: none;  
  }  
}
```



```
nav ul {  
  margin: 0;  
  padding: 0;  
  list-style: none;  
}  
nav li {  
  display: inline-block;  
}  
nav a {  
  display: block;  
  padding: 6px 12px;  
  text-decoration: none;  
}
```

# NESTING WITH COMBINATORS

```
ul > {  
  li {  
    list-style-type: none;  
  }  
}  
  
h2 {  
  + p {  
    border-top: 1px solid gray;  
  }  
}  
  
p {  
  ~ {  
    span {  
      opacity: 0.8;  
    }  
  }  
}
```

```
ul > li { ... }  
  
h2 + p { ... }  
  
p ~ span { ... }
```

# NESTING WITH PARENT SELECTOR



```
.alert {  
  &:hover {  
    font-weight: bold;  
  }  
  
  [dir=rtl] & {  
    margin-left: 0;  
    margin-right: 10px;  
  }  
  
  :not(&) {  
    opacity: 0.8;  
  }  
}
```



```
.alert:hover { ... }  
.alert[dir=rtl] { ... }  
:not(.alert) { ... }
```

# MODULES

```
// _base.scss
$font-stack: Helvetica, sans-serif;
$primary-color: #333;

body {
  font: 100% $font-stack;
  color: $primary-color;
}

// main.scss
@use 'base';

.inverse {
  background-color: base.$primary-color;
  color: white;
}
```

# EXTEND

```
%message-shared {  
  border: 1px solid #ccc;  
  padding: 10px;  
  color: #333;  
}  
  
.message {  
  @extend %message-shared;  
}  
  
.success {  
  @extend %message-shared;  
  border-color: green;  
}
```

# OPERATORS

```
.container {  
  width: 100%;  
}  
  
article[role="main"] {  
  float: left;  
  width: 600px / 960px * 100%;  
}  
  
aside[role="complementary"] {  
  float: right;  
  width: 300px / 960px * 100%;  
}
```

CSS4: `calc(600px/960px*100%)`



# FUNCTIONS

```
@function pow($base, $exponent) {  
  $result: 1;  
  @for $_ from 1 through $exponent {  
    $result: $result * $base;  
  }  
  @return $result;  
}  
  
.sidebar {  
  float: left;  
  margin-left: pow(4, 3) * 1px;  
}
```

# IF ELSE

```
$light-background: #f2ece4;
$light-text: #036;
$dark-background: #6b717f;
$dark-text: #d2e1dd;

@mixin theme-colors($light-theme: true) {
  @if $light-theme {
    background-color: $light-background;
    color: $light-text;
  } @else {
    background-color: $dark-background;
    color: $dark-text;
  }
}

.banner {
  @include theme-colors($light-theme: true);
  body.dark & {
    @include theme-colors($light-theme: false);
  }
}
```

# EACH

```
$sizes: 40px, 50px, 80px;
```

```
@each $size in $sizes {
```

```
  .icon-#{ $size } {  
    font-size: $size;  
    height: $size;  
    width: $size;
```

```
  }
```

```
}
```

# EACH WITH MAPS

```
$icons: ("eye": "\f112", "start": "\f12e", "stop": "\f12f");

@each $name, $glyph in $icons {
  .icon-#{ $name }:before {
    display: inline-block;
    font-family: "Icon Font";
    content: $glyph;
  }
}
```

# FOR

```
$base-color: #036;
```

```
@for $i from 1 through 3 {
```

```
  ul:nth-child(3n + #{ $i }) {
```

```
    background-color: lighten($base-color, $i * 5%);
```

```
  }
```

```
}
```

# WHILE

```
// Divides `$value` by `$ratio` until it's below `$base`.  
@function scale-below($value, $base, $ratio: 1.618) {  
  @while $value > $base {  
    $value: $value / $ratio;  
  }  
  
  @return $value;  
}  
  
$normal-font-size: 16px;  
sup {  
  font-size: scale-below(20px, 16px);  
}
```



Arbeitsgruppen:

Erprobt verschiedene (mindestens 5) SASS Funktionalitäten anhand von selbst erdachten Anwendungsfällen und stellt sie anschließend kurz vor.

Zum Beispiel: **variables**, **mixings**, **nesting with parent**, **for** und **extend**.

```
workshop/
├── assets
│   ├── css
│   │   └── sass-examples.css
│   └── scss
│       └── sass-examples.scss
└── index.html
```

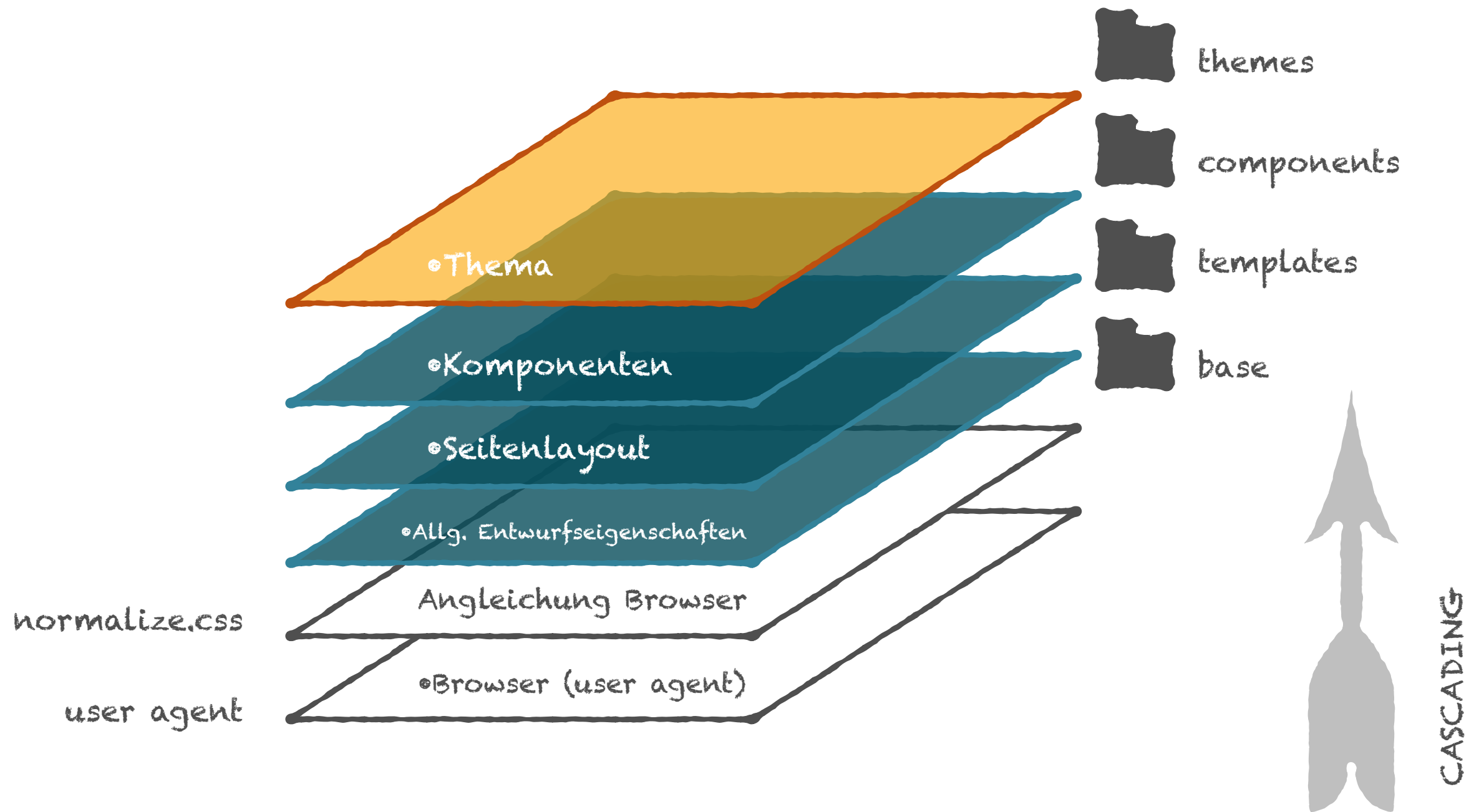
[HTTPS://SASS-LANG.COM/DOCUMENTATION/](https://sass-lang.com/documentation/)

20 MIN

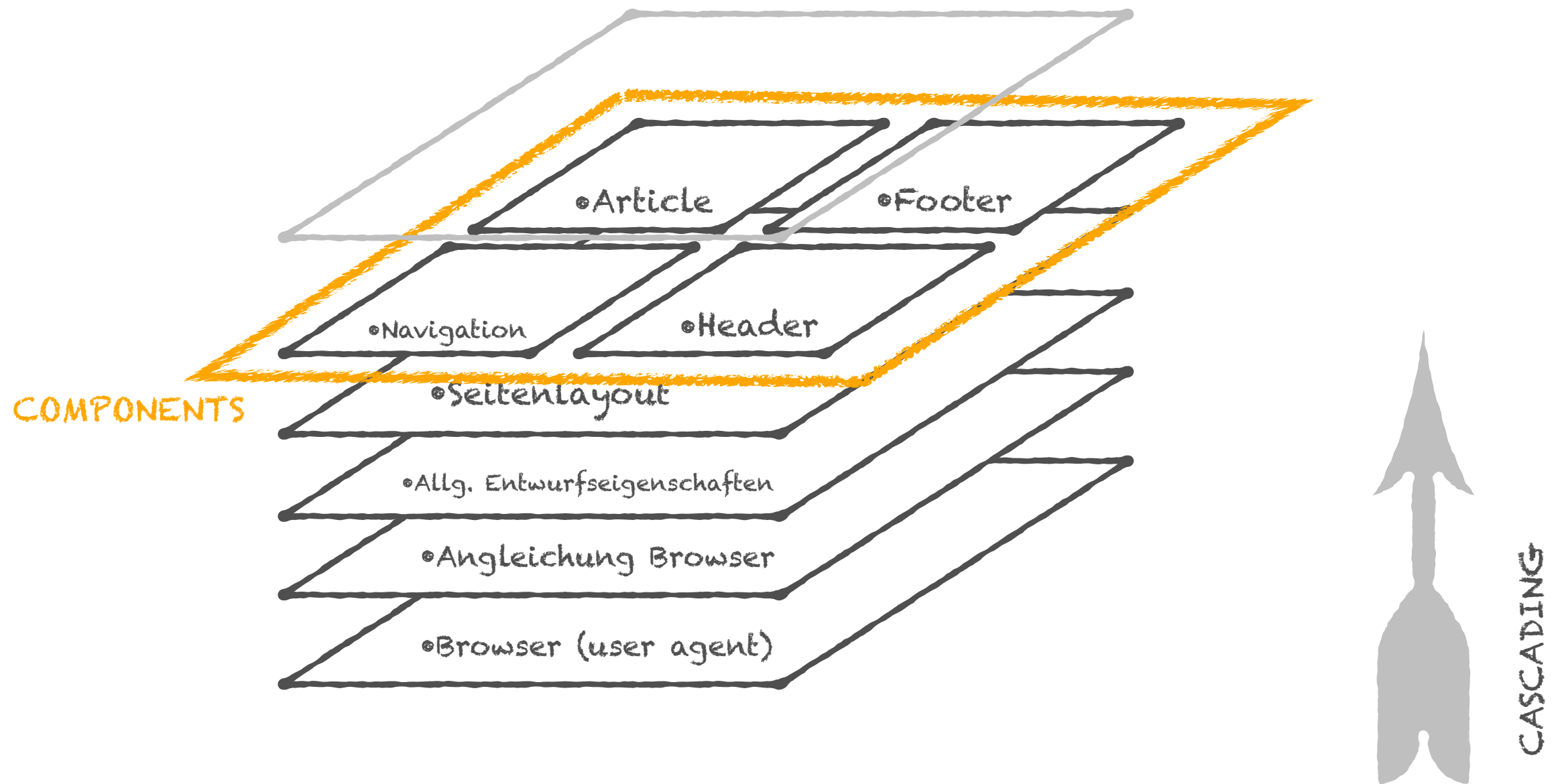
# CASCADING STYLESHEETS ARCHITECTURE



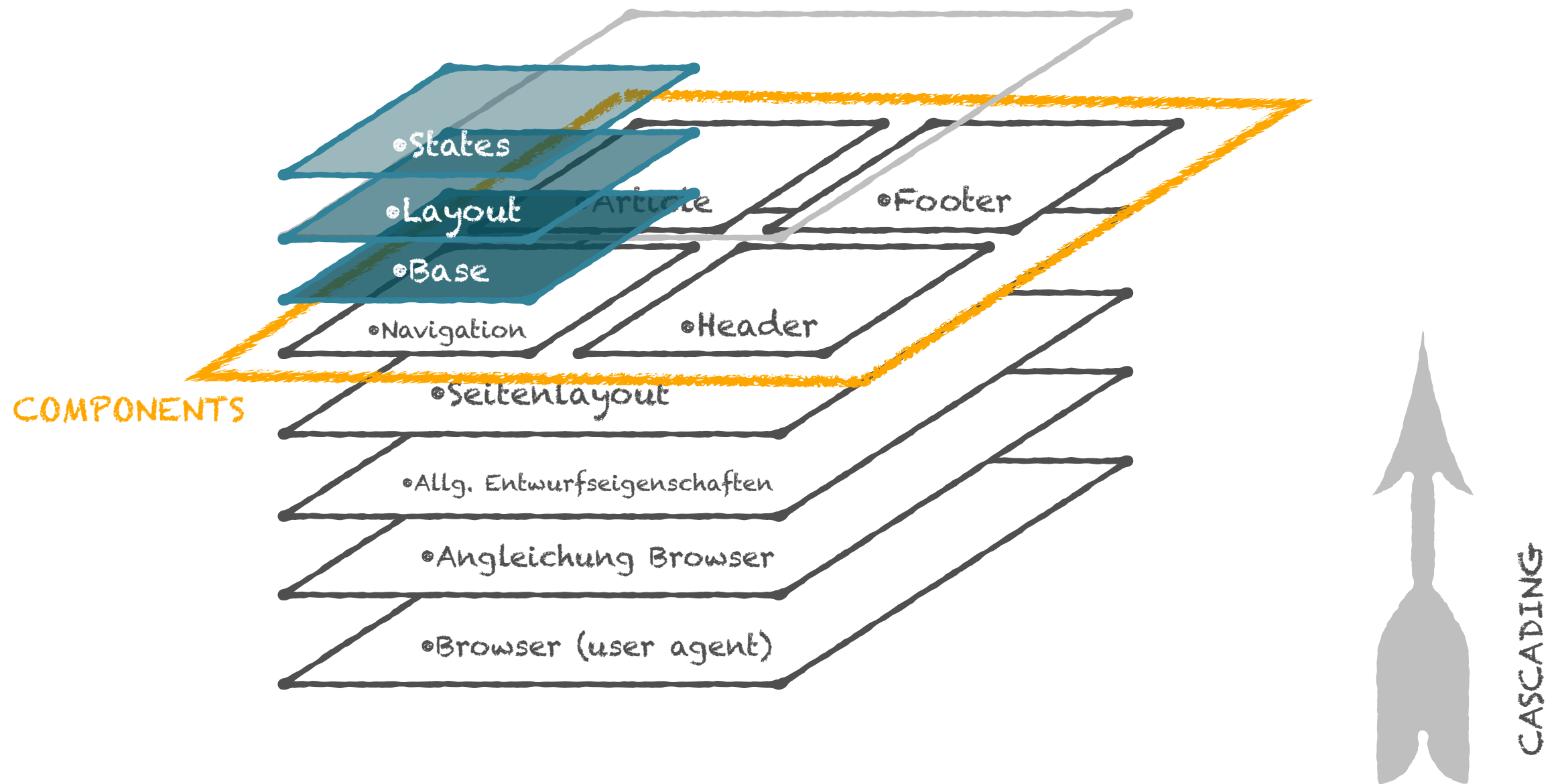
# CSS ARCHITECTURE



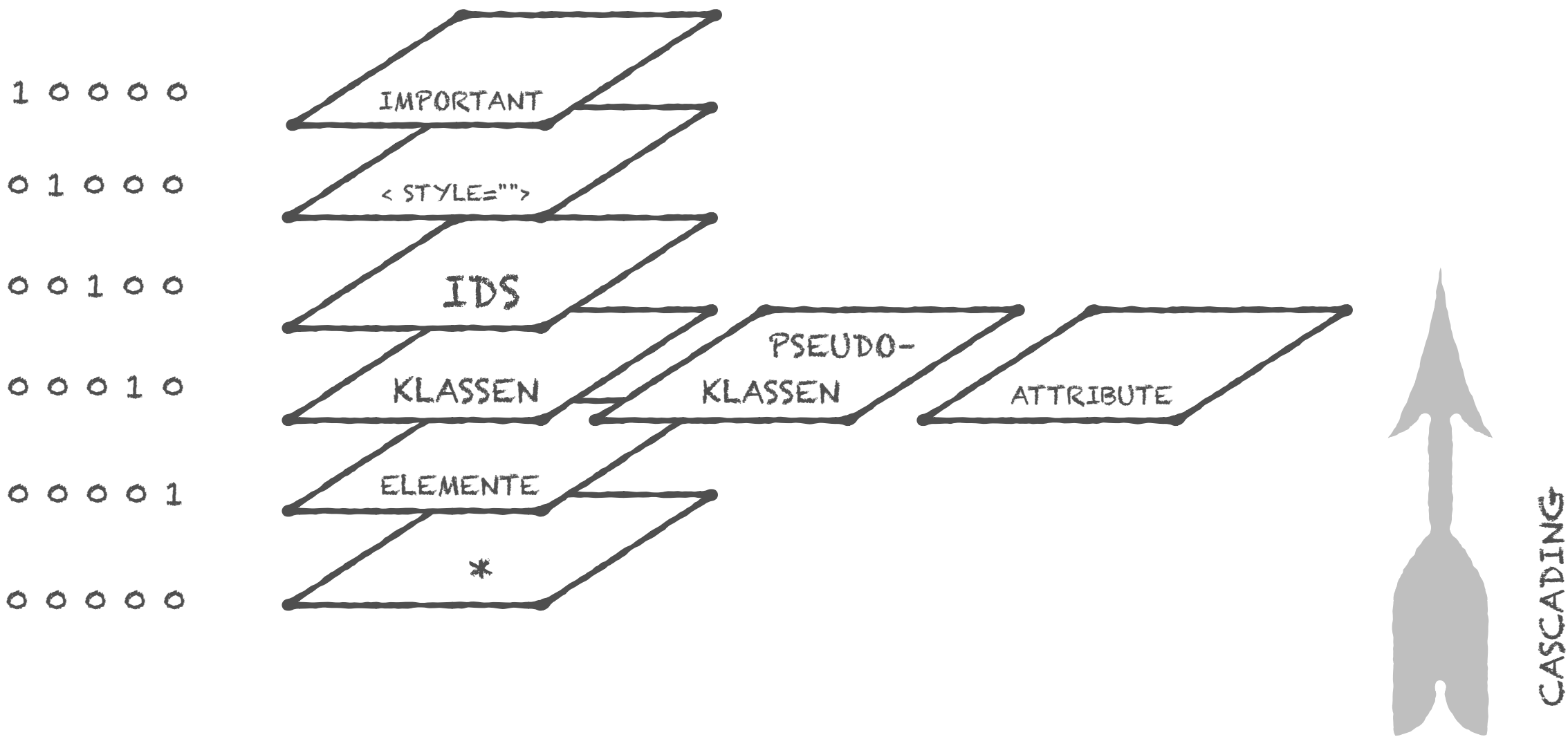
# CSS CASCADE



# CSS CASCADE



# CSS GEWICHTUNG



🕒 IMPORTANT  
ODER < STYLE="" >

🕒 PSEUDOKLASSEN

KLASSEN

🕒 PSEUDOKLASSEN

ELEMENTE

0  
0  
1  
0  
0  
0  
0  
0

0  
0  
0  
0  
0  
0  
0  
1

gen. Zustände

•Thema

States  
Layout  
Base

Komponenten

Seitenlayout

Allg. Zustände

Allg. Entwurf

Angleichung Browser

Browser (user agent)



CASCADING



Wende die CSS Architektur an.

```
workshop/
├── assets
│   ├── css
│   │   └── main.css
│   └── scss
│       ├── _variables.scss
│       ├── _mixins.scss
│       ├── _my-theme-base.scss
│       ├── _my-theme-template.scss
│       ├── _my-theme-component-1.scss
│       └── main.scss
└── index.html
```

15 MIN