

anatom5
perception marketing

anatom5 perception marketing GmbH

+49 211 420 944 1

info@anatom5.de

www.anatom5.de

twitter.com/anatom5

Samstag, 03.09.2016

The new discipline of an frontend Architect.

Worum geht es dabei? Was steckt hinter dem Konzept?

anatom5 Kurzprofil

- 01 14 Jahre Erfahrung → Agentursitz Düsseldorf
- 02 5 feste Mitarbeiter + großes Experten-Netzwerk
- 03 Design für Web, Print und Mobile → Medienkonvergenz
- 04 Schwerpunkte → Barrierefreiheit für Web, Mobile und PDF
- 05 6 Auszeichnungen für beste Barrierefreiheit (BIENE-Award)
- 06 Kunden → Landkreise, Städte, Ministerien, kult. Einrichtung
- 07 Contao → von kleinen Seiten bis hin zu Portalen

Marc Reimann

- 01 MacKP / MacKPWD
- 02 Webdesign seit 2005
- 03 Contao seit 2008
- 04 Contao Yoda
- 05 Mitglied der C-C-A
- 06 Mitglied im MM-Team

Kleine Historie der Frontend Entwicklung

Damals

- Editor
 - HTML
 - Grafiken
 - CSS
 - JavaScript
 - Später ein wenig PHP
- FTP (Filezilla)
- Fertig :-)

Heute (als Auszug!)

- Grafiken
 - SVG? / Font? / Background-
img? / Responsive
<picture> / Sprite?
- Browsertesting
 - IE8? / Android (old?) /
IE9/10/11/Edge / Firefox /
Chrome / Safari / Opera /
usw.
- Auflösung? Smartphone bis TV /
Mediaqueries (ein paar?)
 - PS mit Desktop / Tablet /
Smartphone / und?
- Statische vs. Dynamische Seite
(CMS)
- IDE
- CSS- , JavaScript-, PHP-
Framework?
- Präprozessoren
 - SASS / Less
 - Variablen
 - Funktionen
 - Mixins
- Grunt? / Gulp?
- Autoprefixer
- PHP
- Dokumentation vs Zeit
- Geschwindigkeit der Seite
- Hochladen: Wo? Wie? Wann?

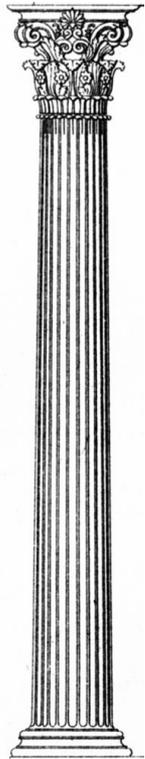
„One of our strongest skills is that we can spend an hour with a new framework or Gulp plug-in, and identify its strengths and weaknesses and possible use in our project.“

Von Micah Godbolt in 'Frontend Architecture for Design Systems'

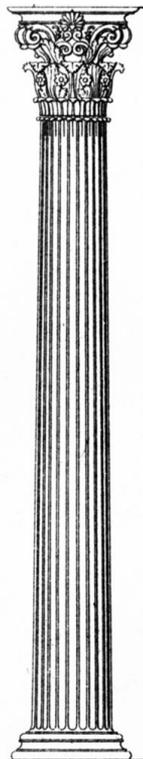
Ordnung ist das halbe Leben!

- Modular content
 - Besser wenige kleine Komponenten als viele große
- Comprehensive testing
 - Testen wie Entwickler (früh / direkt / vor Livegang!)
- Streamlined processes
 - Git flow mit kleineren besser zu überblickenden Bereichen / automatische Prozesse für Dokumentation und Verarbeitung
- Exhaustive documentation
 - Viele Arbeiten Hand in Hand (Backend-Developer / Frontend-Developer / Designer / Marketing Manager / usw.)
→ Dokumentation muss alle zusammen bringen!

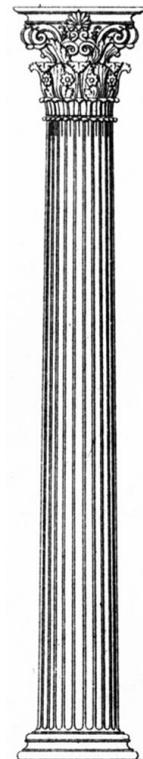
The Pillars of Frontend Architecture



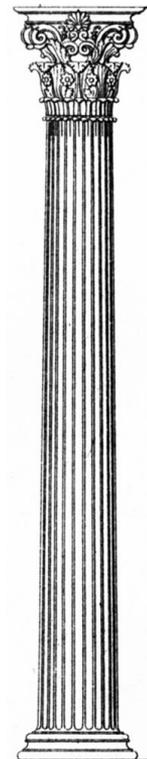
1. the Code Pillar



2. the Process Pillar



3. the Testing Pillar



4. the Documentation Pillar

1. The Code Pillar

Wie setzen wir unser HTML, CSS und JavaScript um?

2. The Process Pillar

The development process – Tools und Prozesse, die wir nutzen, um die Produktivität und Effizienz zu erhöhen.

3. The Testing Pillar

Wird immer wichtiger, je größer die Seite ist,
an der wir arbeiten.

4. The Documentation Pillar

Nicht erst nachdem das Projekt fertig ist!
Dokumentiert wird immer!

4. The Documentation Pillar

Nicht erst nachdem das Projekt fertig ist!
Dokumentiert wird immer!

1. The Code Pillar

„Good code never happens by accident“

Von Micah Godbolt in 'Frontend Architecture for Design Systems'

1. The Code Pillar: HTML

Procedural Markup:
100% Automation, 0% Control

- HTML-Code wird generiert (z.B. Contao)
- Frontend-Team muss damit leben
- „div soup“

```
<div id="header" class="clearfix">
  <div id="header-inner" class="clearfix">
    <div id="nav-header" role="navigation">
      <div class="nav-header-main">
        <ul class="menu">
          <li class="first">
            |
          </li>
        </ul>
      </div>
    </div>
  </div>
</div>
```

1. The Code Pillar: HTML

Static Markup:
0% Automation, 100% Control

- Meist nur noch bei Statischen Seiten zu finden
- HTML ist sehr minimal und semantisch gut
- CSS wurde zu einem 'specificity nightmare'

```
<header>
  <nav>
    <ul>
      <li>
        ...
        <ul>
          <li>
            ...
          </li>
        </ul>
      </li>
      <li>
        ...
      </li>
    </ul>
  </nav>
</header>
```

```
header > nav > ul > ul > li > a { color: green; }
header > nav > ul > ul > ul > li > a { color: red; }
```

1. The Code Pillar: HTML

Balance Between Control and Automation?

Modular Markup!

1. The Code Pillar: HTML

Modular Markup: 100% Automation, 100% Control

- In einem CMS nicht immer zu 100% sinnvoll
- HTML und CSS durch Klassen trennen
- Spezifität von Selektoren im CSS möglichst gering halten

```
<nav class="nav">
  <ul class="nav__container">
    <li class="nav__item">
      <a href="#" class="nav__link">Link</a>
      <ul class="nav_container--secondary">
        <li class="nav_item--secondary">
          <a href="#" class="nav__link--secondary">Link</a>
        </li>
      </ul>
    </li>
  </ul>
</nav>
```

```
.nav__link { color: green; }
.nav__link--secondary { color: red; }
```

1. The Code Pillar: HTML

Modular Markup:

- OOCSS
(Object-Oriented CSS)
- SMACSS
(Scalable and Modular Architecture for CSS)
- BEM
(Block Element Modifier)

1. The Code Pillar: HTML

Modular Markup:

- Alle sind gut und haben Ihren Sinn.
- Keins passt zu 100% zu dir/euch und dem Projekt...
- Such dir das Richtige für dich, dein Team und das Projekt!
- Denkt euch etwas eigenes aus. Es ist egal. Hauptsache ihr bleibt einmal dabei!

1. The Code Pillar: CSS

Specificity Wars and the Pains of Inheritance!

```
<div class="main">
  <h2>My Headline in the main Content ist red</h2>
</div>
<div class="sidebar-right">
  ⚡<h2>Headlines in the sidebar are green</h2>
  <div class="news">
    <h2>News in Important red!</h2>
  </div>
</div>
```

```
.main h2 { color: red; }
.sidebar-right h2 { color: green; }
.sidebar-right .news h2 { color: red; }
```

1. The Code Pillar: CSS

Specificity Wars and the Pains of Inheritance!

- **Single Responsibility Principle**
 - Jede Klasse sollte für einen Grund genutzt werden.
 - D.h. NICHT, für alles eine Klasse (padding-20, font-size-23, color-green)

```
<div class="main">
  <h2 class="main-header">My Headline in the main Content ist red</h2>
</div>
<div class="sidebar-right">
  <h2 class="sidebar-right-header">Headlines in the sidebar are green</h2>
  <div class="news">
    <h2 class="news-header">News in Important red!</h2>
  </div>
</div>
```

```
.main-header { color: red; font-size: 2em; }
.sidebar-right-header { color: green; font-size: 1.5em; }
.news-header { color: red; font-size: 1.5em; }
```

1. The Code Pillar: CSS

Component Modifiers?

- **Modifizieren von CSS je nach Ort der Nutzung**
 - News: Im Hauptbereich? In der Sidebar? Im Footer?

```
<div class="main">
  ...
</div>
<div class="sidebar-right">
  <div class="news news--nested">
    <h2 class="news-header">News ...</h2>
  </div>
</div>
```

```
.sidebar-right .news--nested {
  font-size: 2em;
}
.news-nested .news-header {
  font-size: 4em;
}
```

1. The Code Pillar: CSS

- ID → Klasse
- Versuch die Spezifität von einem Selektor bei 1 zu halten
- Theming über eine extra Klasse (Farbe)
- Vermeide Selektoren mit Bezug zur Umgebung (main .selector, footer .selector)
- Vermeide Verschachtelungen im CSS (Präprozessor!!!)

1. The Code Pillar: CSS

weitere Infos

- <https://erdmann-freunde.de/css-kurs-fuer-com>
- <http://oocss.org/>
- <https://smacss.com/>
- <http://getbem.com/>
- ...

1. The Code Pillar: J S

- Such dir das passende Framework
- Keep your Code clean
- Erstelle wiederverwendbare Funktionen
- Mit Klassen arbeiten:
js- → js-accordion-open / js-accorion-closed

1. The Code Pillar: Allgemein

Schau dir das Layout an und erstelle
Einheiten → Katalogisiere die Einheiten!

Ansätze:

- Atomic Design

<http://bradfrost.com/blog/post/atomic-web-design/>

– Brad Frost

- Webseiten modular entwickeln

<https://speakerdeck.com/flocke/webseiten-modular-entwickeln>

– Jens Grochtdreis

1. The Code Pillar: Regeln

- 1) Regeln sollen nur einen Fall beschreiben und so einfach wie möglich sein.
- 2) Erstelle dir deine eigenen Regeln und schreib sie auf!

1. The Code Pillar: Regeln

Beispiele:

- 1) Ein Selektor für alles
- 2) Ein Selektor für eine Einheit (und deren Elemente)
- 3) Single Source of Truth
- 4) Für eine Einheit immer eine .css / .scss Datei
- 5) SCSS-Dateien in passenden Ordnern ablegen
- 6) ...

2. The Process Pillar

- Ticket System, um den Fortschritt von dem Projekt zu erkennen und zu verfolgen
- Erstelle eine Entwicklungsumgebung, wo du Änderungen testen kannst
- Baue dir Build Prozesse, um Code zu validieren, compilieren und zu testen
- Push getesteten Code in die Entwicklungsumgebung
- Getesteter und abgenommener Code dann erst in die Live-Umgebung

2. The Process Pillar: Workflow

Alter Workflow

- Anforderungen
- Wireframe
- Design | Development
- Frontend-Entwicklung

Alle Gruppen arbeiten sehr getrennt voneinander

2. The Process Pillar: Workflow

Neuer Workflow

- Anforderungen (Designer / Frontend-Entwickler / Kunde / usw.)
- Prototype (Designer / Frontend-Entwickler / Kunde / usw.)
- Development (Designer / Frontend-Entwickler / Kunde / usw.)

Alle Gruppen arbeiten zusammen an den unterschiedlichen Prozessen und sollten mit einbezogen werden.

2. The Process Pillar: Workflow

Task-Runners

Gulp

- <http://gulpjs.com/>
- Ist ein wenig schneller

Grunt

- <http://gruntjs.com/>
- Hat(te) die meisten fertigen Module

Welchen?

Egal! Such dir einen Passenden aus.
Beide helfen dir beim Automatisieren.

2. The Process Pillar: Workflow

- Dateien
/source + gulpfile.js + package.json + path-config.json
- 2 Dinge:
 - 1) npm installieren
 - 2) Im Projektordner: npm install --save-dev
- Task ausführen: gulp -serv (Phpstrom macht es einfacher)
- Von /source wird jetzt alles verarbeitet :)

3. The Testing Pillar

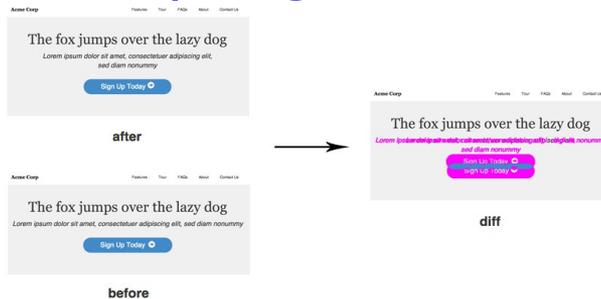
- Test driven development – Unit Testing (erst den Test schreiben, danach den Code) → QUnit
<https://qunitjs.com/>
- Performance Testing: Performance Budget – WebPageTest API
<https://css-tricks.com/use-webpagetest-api/>

3. The Testing Pillar

- **Visual Regression Testing**
Automatische visuelles Testen - nicht mehr per Hand alle Seiten durchklicken
 - Page-based diffing
 - Component-based diffing
 - CSS unit testing
 - Headless Browser driven
 - Desktop browser driven
 - GUI-based comparison and change approval
 - Command-line comparison and change approval

3. The Testing Pillar

- <http://screengui.de/31> „Erfolgsfaktor Layout-Testing“
 - WebdriverCSS
<https://github.com/webdriverio/webdrivercss>



- Galen Framework
<http://galenframework.com/>

Galen Report

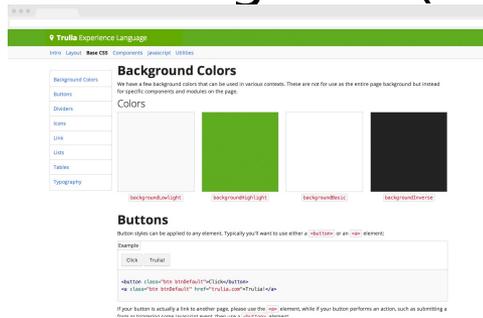
Passed	Failed	Warnings	Total	Started	Duration	Test
13	0	0	13	2015-01-04 21:04:30	19s	Home page on Safari on Mac browser
13	0	0	13	2015-01-04 21:04:49	28s	Home page on Opera browser
13	0	0	13	2015-01-04 21:05:17	24s	Home page on Firefox browser
13	0	0	13	2015-01-04 21:05:42	37s	Home page on Chrome browser
8	5	0	13	2015-01-04 21:06:20	25s	Home page on IE 11 browser

4. The Documentation Pillar

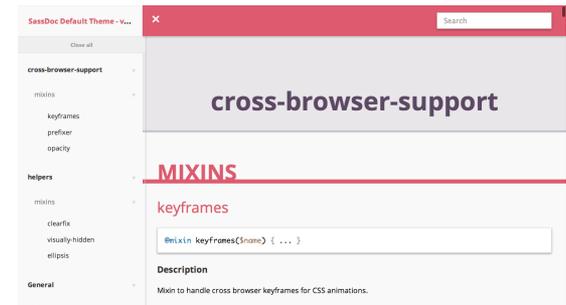
- Alles
- Sofort
- Im Code
- Extern
- Für alle
 - Designer
 - Frontend-Developer
 - Backend-Developer
 - Kunden
 - Für die Zukunft!

4. The Documentation Pillar

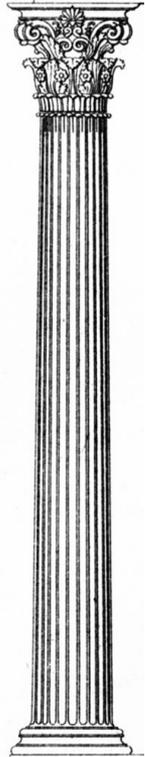
- Statische Dokumentation (Living-Styleguide)
 - Hologramn (<https://trulia.github.io/hologram/>)



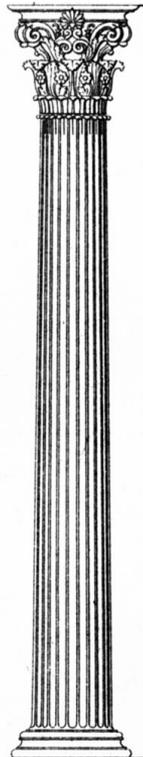
- SassDoc (<http://sassdoc.com/>)



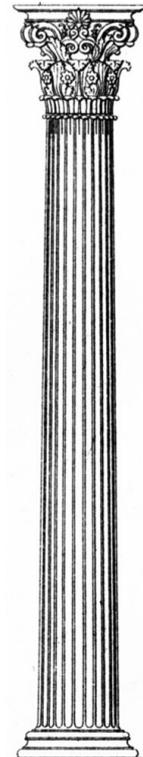
The Pillars of Frontend Architecture



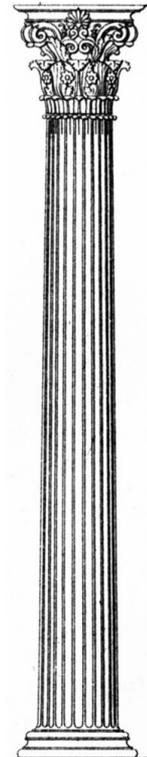
1. the Code Pillar



2. the Process Pillar



3. the Testing Pillar



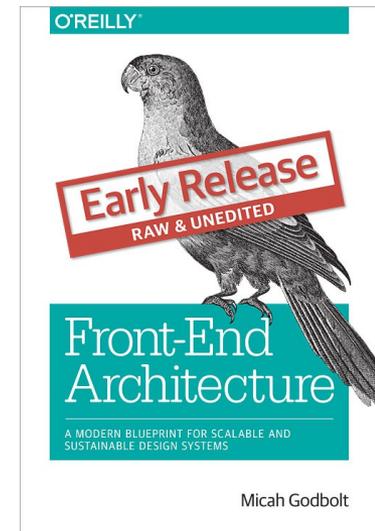
4. the Documentation Pillar

Frontend Architecture for Design Systems

Von Micah Godbolt

<http://frontendarchitecture.info/>

(nicht mehr erreichbar?)



<https://github.com/micahgodbolt/front-end-architecture>

Weiteres

Vortrag 'Code Patterns for Pattern-Making'
von Miriam Suzanne auf dem CascadiaFest 2016

<http://2016.cascadiafest.org/speakers/miriam-suzanne/>

Contao?

- Code ist vorgegeben?
 - Einiges schon gut!
 - Anderes nicht so...
 - Wie können wir das erreichen, was wir wollen?

Contao?

- Für alles ein eigenes Template?
 - /templates allgemeine für jedes Projekt
 - /templates/theme spezifisch für das Projekt
- Erdmann & Freunde: CSS-Kurs
<https://erdmann-freunde.de/css-kurs-fuer-contao.html>
- Contao CSS class replacer extension von Toflar
<https://github.com/Toflar/contao-css-class-replacer>

Contao?

- <http://nutshell.erdmann-freunde.de/>



Contao?

- Wie seht ihr das?
- Wie macht ihr das aktuell?

anatom5
perception marketing

anatom5 perception marketing GmbH

+49 211 420 944 1

info@anatom5.de

www.anatom5.de

twitter.com/anatom5

http://downloads.anatom5.de/folien/2017_contao_konferenz_frontend-architecture.pdf