European SharePoint Conference

Make SharePoint SASSY

Webinar

Stefan Bauer – n8d

Stockholm
NOVEMBER 9-12TH -2015

STEFAN Bauer
Est. 2015
★ N 8 D ★
WEB DESIGN & DEVELOPMENT

Information Architect
Vienna / Austria

n8d.at/blog

@StfBauer

Make SharePoint SASSY!

- I will explain what SASS is and how you can use it
- How to use SASS to brand SharePoint without requiring lengthy deployments.
- How to create simple Rich Text Editor Styles using mixins and includes.
- How to apply a Grid layout and make it Responsive.
- How to structure your branding correctly to make it more maintainable.
- How CSS 4 fits into the picture and does it make SASS obsolete?

BRANDING CHALLENGES

INTRODUCTION TO SASS

STRUCTURE YOUR BRANDING

Changed SharePoint Development

2003

Inject CSS
154 files

Search & Destroy

2007 / 2010
2013

Golden Age
Of
Master Page

2013 / 2016
Office 365

CSS or JS

Cloud or On-Premises

Old Workflow

1. Create a farm solution
2. Add Master Page
3. Add CSS File
4. Deploy
5. Customize CSS File on Server
6. Copy final CSS File to Farm Solution
7. Deploy Solution
8. Repeat step 1 - 8

PROS:

- Worked pretty well
- Centralized Design Files
- Ghosting

CONS:

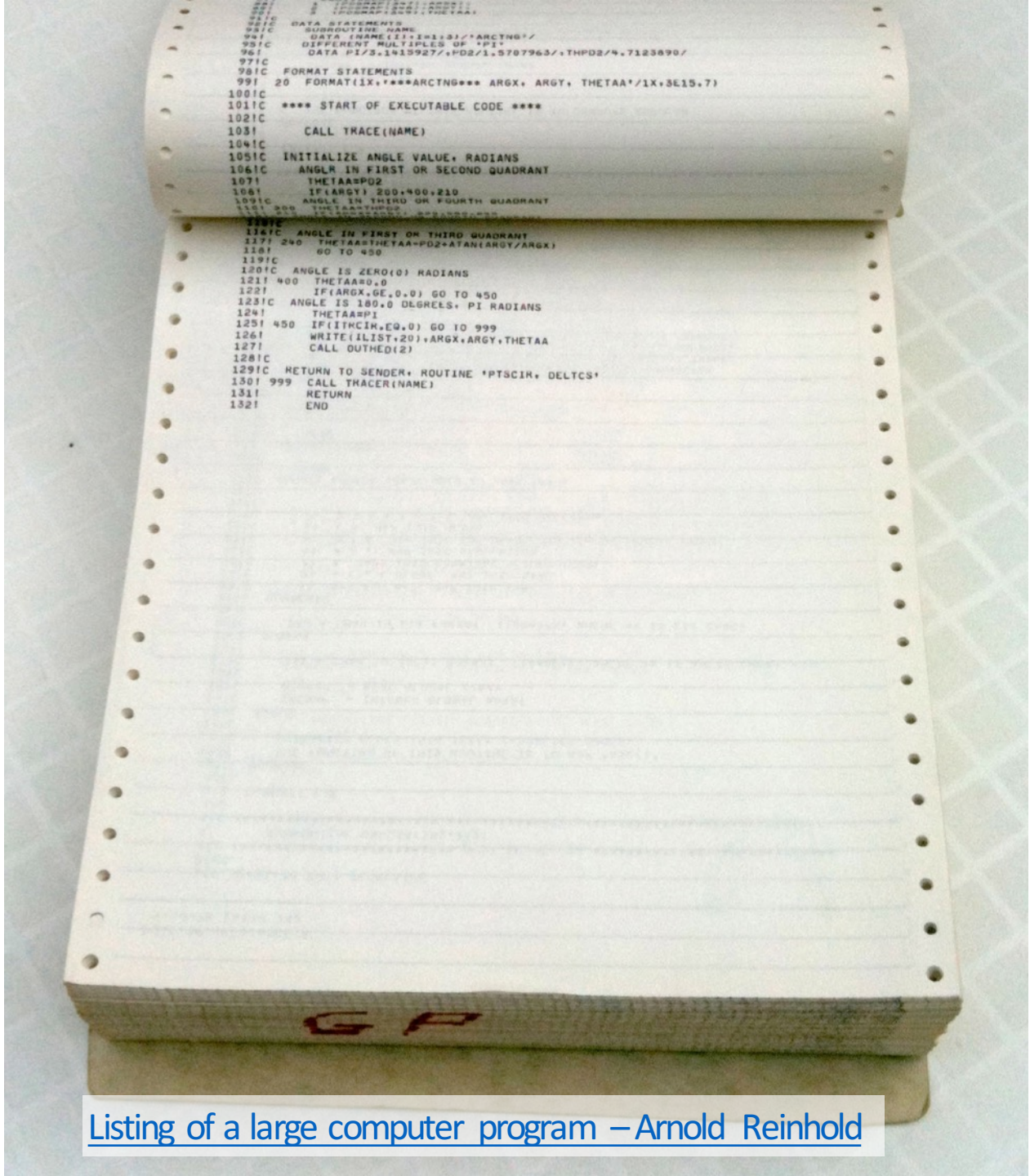
- Development on the Server
- Legacy farm solution
- One single CSS File
- Not cloud ready

Modularize CSS

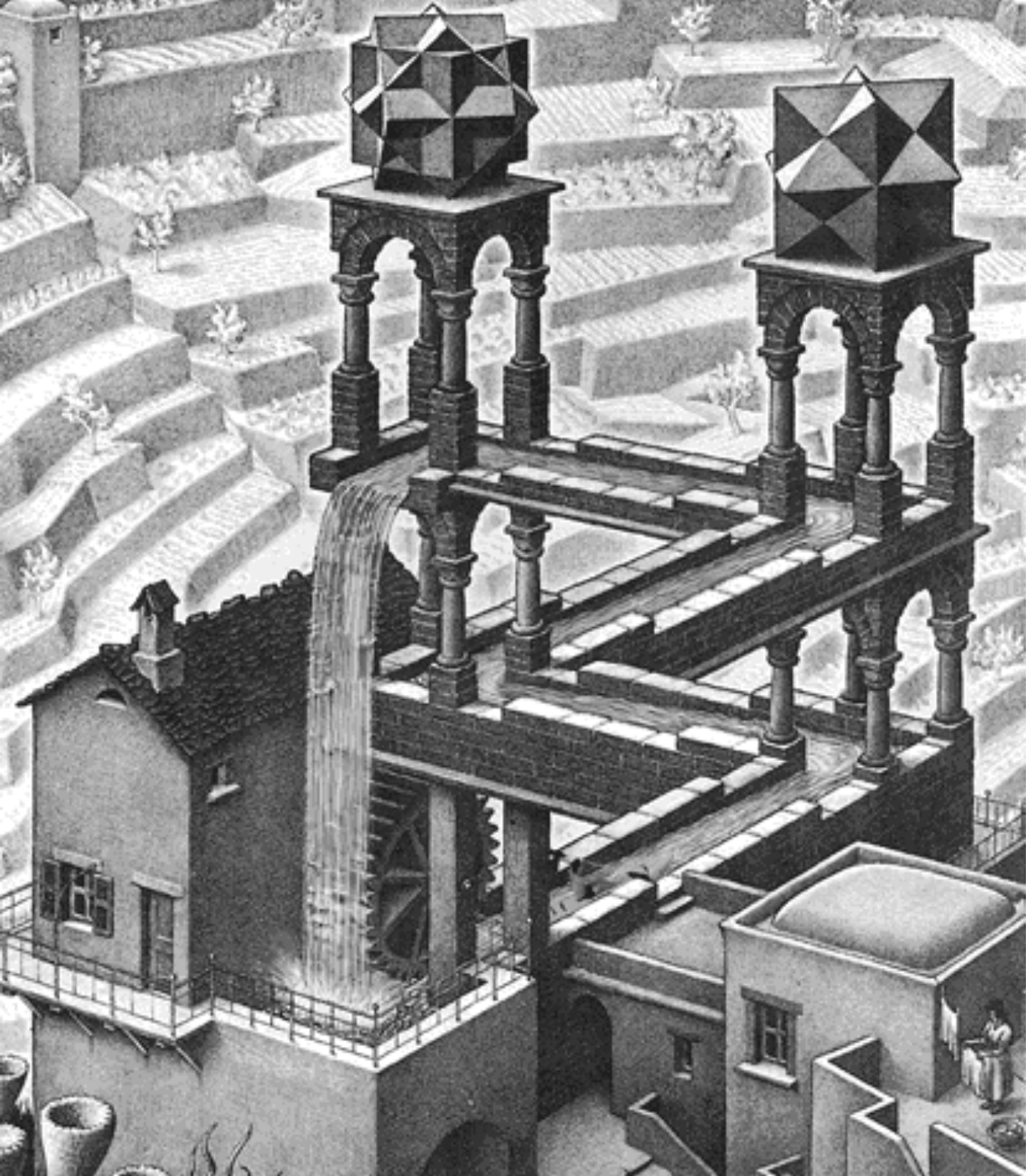
Find and identify UI elements faster

Divide complexity into smaller parts

Avoid development conflicts



Listing of a large computer program – Arnold Reinhold



Avoid Repeating Tasks

Reusable components

More maintainable code

rce: http://my.n8d.at/1MVjnEV

Why is CSS the way it is?

CSS stops short of even more **powerful features** that programmers use **in** their **programming languages**: macros, variables, symbolic constants, conditionals, expressions over variables, etc. That is because these things **give power-users a lot of rope**, but **less experienced users will unwittingly hang themselves**; or, more likely, be so scared that they **won't even touch CSS**. It's a balance. And for CSS the balance is different than for some other things.

– Bert Bos – A list apart

CSS Spaghetti Code

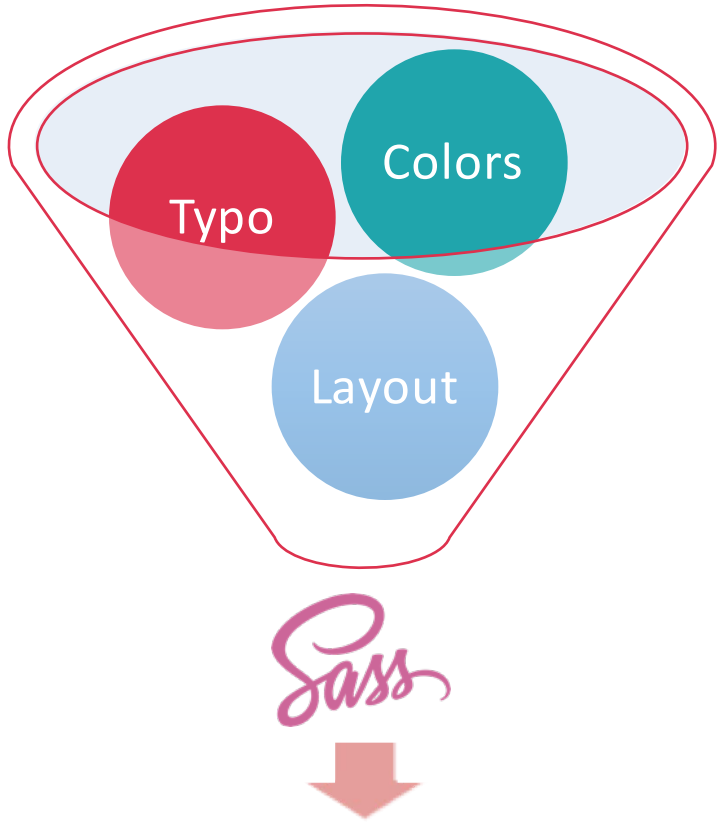
BRANDING CHALLENGES

INTRODUCTION TO SASS

STRUCTURE YOUR BRANDING

What is SASS

Sass (short for Syntactically Awesome Style Sheets)



A layer between style sheets and CSS

faster, more efficient, and easier to maintain.

SASS – Choose your style

```
#main p
    color: #00ff00;
    width: 97%;
    .redbox
        background-color: #ff0000;
        color: #000000;
```

SASS – Syntax

The old school RUBY based syntax

File extension: .sass

```
#main {
    p {
        color: #00ff00;
        width: 97%;
        .redbox {
            background-color: #ff0000;
            color: #000000;
        }
    }
}
```

SCSS – Syntax

Like you writing native CSS

File extension: .scss

SASS – for developer

Hack CSS

Benefit of a programming language

Calculates the headline font based on a dynamic ratio. Create a font shorthand definition by merging all the font properties like this.

CSS:

font: italic small-caps bold 1em/140% Helvetica, sans-serif;

```
@mixin headlines(){

    & h1, & h2, & h3, & h4{

        @if ($sp-hl-padding){
            padding: $sp-hl-padding;
        }

        @if ($sp-hl-colors and length($sp-hl-colors) == 1){
            color: $sp-hl-colors;
        }
        margin: 0px;
    }

    @for $i from 1 through $sp-hl-count{
        h#{$i} {
            $font-size : 100%+(($sp-hl-count - $i+1)*$sp-hl-factor);
            font: fontShortHand($font-size);
            @if ($sp-hl-colors
                    and
                length($sp-hl-colors) >= $i){
                color: nth($sp-hl-colors, $i);
            }

        }
    }
}
```

SASS - Variables

```scss
// Variables
$green: #2ECC40;
$font-main: "Lobster", sans-serif;
$def-border: 1px lime solid;

// Usage
.box{
  background-color: $green;
  font-family: $font-main;
  border: $def-border;
}
```

```css
.box {
  background-color: #2ECC40;
  font-family: "Lobster", sans-serif;
  border: 1px lime solid;
}
```

W3C - CSS Variables – Specification

```
:root {
  --main-color: #06c;
  --accent-color: #006;
}
/* make use of variables. */
#foo h1 {
  color: var(--main-color);
}
```

CSS – Variables

Post CSS – JavaScript

SASS – Nesting < Parent

```scss
a {
    color: green;
    &:hover {
        color: blue;
    }
    &:visited {
        color: green;
    }
}

.container {
    width: 800px;
    & &-item {
        display: block;
        width: 200px;
        height: 200px;
        float: left;
    }
}
```

```css
a {
    color: green;
}

a:hover {
    color: blue;
}

a:visited {
    color: green;
}

.container {
    width: 800px;
}

.container .container-item {
    display: block;
    width: 200px;
    height: 200px;
    float: left;
}
```

SASS – Variables - Data types

- **Numbers**
  1.2, 13, 10px
- **Strings** with or without quotes
  "foo", 'bar', baz
- **Colors**
  blue, #04a3f9, rgba(255, 0, 0, 0.5)
- **Booleans**
  true / false
- **Nulls**
  null – what else

- **List of values**
  separated by spaces or commas
  1.5em 1em 0 2em
  Helvetica, Arial, sans-serif
- **Maps from one value to another**
  (key1: value1, key2: value2))

SASS - @extend

Define element once – reuse it with overrides

```
.box {
    width: 10em;
    height: 5em;
    backgroud-color: blue;
}

.box-green {
    @extend .box;
    background-color: green;
}

.box-yellow{
    @extend .box;
    background-color: yellow;
}
```

```
.box, .box-green, .box-yellow {
  width: 10em;
  height: 5em;
  backgroud-color: blue;
}

.box-green {
  background-color: green;
}

.box-yellow {
  background-color: yellow;
}
```

SASS – Placeholder @extend

“%” acts like a new pseudo selector in addition to class and id css selector

Extend element templates without extending compiling the selector itself

SASS – Placeholder @extend

```scss
%box {
    width: 10em;
    height: 5em;
}

.box-blue {
    @extend %box;
    background-color: blue;
}

.box-green {
    @extend %box;
    background-color: green;
}

.box-yellow{
    @extend %box;
    background-color: yellow;
}
```

```scss
.box-blue, .box-green, .box-yellow {
  width: 10em;
  height: 5em;
}

.box-blue {
  background-color: blue;
}

.box-green {
  background-color: green;
}

.box-yellow {
  background-color: yellow;
}
```

SASS - @mixin / @include

@mixin

- Re-use bigger chunks of CSS
- Acts like functions
- Pass parameter
- Default values for parameter

@include

- Extend a selector with CSS

SASS - @mixin

CSS 3 transition mixin and browser fallback definitions

```
@mixin transition($transition-property, $transition-time, $method, $delay: 0s) {
    -webkit-transition: $transition-property $transition-time $method $delay;
       -moz-transition: $transition-property $transition-time $method $delay;
        -ms-transition: $transition-property $transition-time $method $delay;
         -o-transition: $transition-property $transition-time $method $delay;
            transition: $transition-property $transition-time $method $delay;
}
```

SASS - @include

```
.box{
    width: 200px;
    height: 200px;
    background-color: $c-red;
    @include transition(background-color, 0.5s, ease-in-out);
    &:hover{
        background-color: $c-red-light;
    }
}
```

@import – CSS vs SASS

CSS

- **Each file will be requested separately**
- **Slow down UI**
- **Don’t use it**

SASS

- **Compiles partials it into CSS file**
- **Supports CSS import**
- **Allows to structure the design**

@import – CSS vs SASS

Filename: _yourvariables.scss

Use in SASS File:

@import ‘yourvariables’

@import ‘core/mixins’

Don’t forget the underscore!

ONE FILE TO **RULE** THEM ALL,
ONE FILE TO **FIND** THEM,
ONE FILE TO **BRING** THEM ALL,
AND IN THE SASS WAY **MERGE** THEM.

-J.R.R TOLKIEN

Why SASS?

- CSS with superpower!
- Clean structured code
- Reusable components

- **Consistency**

- Many Extensions

Organizing CSS:

- *OOCSS*
  Object oriented CSS
- *SMACSS*
  Scalable and Modular Architecture for CSS
- *BEM*
  Block Element Modifier

Blog post:

Organizing CSS: OOCSS, SMACSS, and BEM

- CodeKit (Paid)
- Compass.app (Paid, Open Source)
- Hammer (Paid)
- Koala (Open Source)
- LiveReload (Paid, Open Source)
- Mixture (Free)
- Prepros (Paid)
- Scout (Open Source)

SASS in Visual Studio
VS 2013 – Updated 2

gem install sass

Sassmeister

Tryout & Learn

http://sassmeister.com

Yeoman – yeoman.io

Web related template engine

Platform

Task Runner

Like MS Build

Bower

Like NuGet

Runs on all major platforms (Windows, Mac, Linux)

Yeoman – recommended templates

- **generator-webapp – deprecated**
  Template for web applications – ~~grunt~~ gulp

- **generator-gulp-webapp**
  Template for web applications – gulp

- **generator-angular**
  Template for angularjs

- **YO OFFICE! - Microsoft Office Project Generator**
  Template for writing Microsoft Office add-ins (aka apps)

Yeoman – generator-webapp

- Webserver (Node.js)
- Integrated SASS support
- Auto-compile SASS to CSS
- Auto reload web page
- CSS can be integrated into SharePoint on-premises or Office 365

Work Locally

Inject CSS and JS

No deployment downtime

Use JSOM / Rest API

Like an Add-in (App)

Full SASS Support

Works for Office 365 and on-premises

Work Locally

DEMO

- Integrate Yeoman into SharePoint
- Font reset in SharePoint
- Rich Text Editor mixin

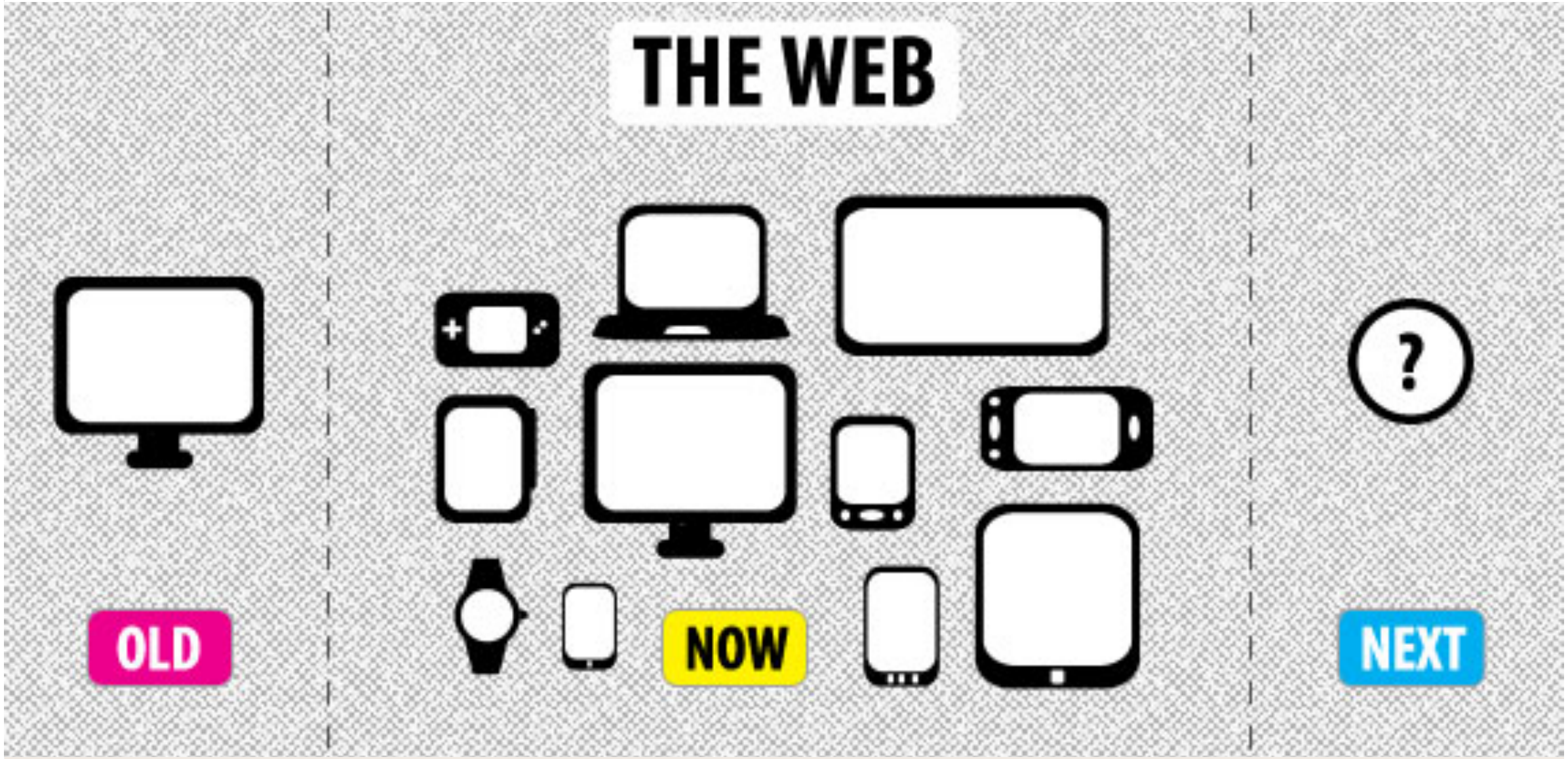
Work Locally – more information

Getting started

How I develop in SharePoint and Office 365 now

How to use CSS and JavaScript files from Yeoman directly in SharePoint

Make SharePoint Responsive



http://www.zeendo.com/info/wp-content/uploads/2013/03/responsive-web-design.jpg

Responsive Web Design

1. Define and create grid layout
2. Define media breakpoints
3. Define how the content looks


DER STANDARD
Angst vor politischem Patt nach Italien-Wahl
Doppelt hält besser: Oscars für Waltz und Haneke
Streit um Herkunftskennzeichnung von Fleisch
Pseudofragen
Nachrichten in Echtzeit auf
derStandard.at
BLAHA INSTINCT BEGEGNUNGSRÄUME BÜRO.
blaha
DER STANDARD KOMPAKT
Doppelter Erfolg: Oscars für Waltz und Haneke
Angst vor politischem Patt nach Italien-Wahl
Linke musste um Mehrheit zittern
Erfolg für Grillo, Berlusconi stark
Exotische Spekulationen
Salzburg arbeitete mit Geld des Bundes
Nachrichten in Echtzeit auf
derStandard.at

Susy Grid

Your Layout - our math

- Calculate grid
- Easy to use
- Built on SASS
- Easy SharePoint integration

- Susy Grid / Tutorials

- SASS Mediaqueries
  Project on github

DEMO

BRANDING CHALLENGES

INTRODUCTION TO SASS

STRUCTURE YOUR BRANDING

We’re not designing pages,
we’re designing systems of components.

Stephen Hay

Responsive deliverables should look a lot like fully-functioning Twitter Bootstrap-style systems custom tailored for your clients' needs. –

Dave Rupert

Beware of MICRO BRANDING

JSLink, Display Templates and Script Embed Web Parts allow you to add additional styles to SharePoint The problem is how the styles.

Modern Style Guides

- HTML based style documentation
- Dynamic and extendable
- Allow to build new UI Components
- Allows teams to work better together

Periodic Table of the Elements

Atomic Web Design – Brad Frost

SEARCH THE SITE

LABEL

ENTER KEYWORD

INPUT

SEARCH

BUTTON

SEARCH THE SITE

ENTER KEYWORD

SEARCH

pattern lab

documentation | about | resources | demo | on github

pattern lab

create atomic design systems

http://patternlab.io

download | view on github | view demo

Recap – SASS

- **SASS** helps you to write better CSS
  - Keep an eye on the source code
  - Well structured

- Supports **DRY** (Don’t repeat yourself) your development

Recap - Create your own style guide

- Sketch in **HTML first**
- Create reusable components
- Document all design assets
- Add things you need rather than use a full version of framework

Resources used in demo

- How I develop in SharePoint and Office 365 now
- Apply grid system to SharePoint using Susy
- Pure CSS Burger
- SharePoint Font Reset
- Responsive SharePoint Project on GitHub

Have fun making SharePoint