

Make your code delicious

@sturobson

About Me.

Stuart Robson

- front-end developer
- all the buzzwords
- rwdcalc.com
- SassnotSASS.com



Making your code delicious

- Helps you care about your work.
- Makes you aware of what you're doing.
- Gives understanding as to what you're doing.
- Helps others comprehend what you've done.

Boilerplates, Frameworks, Systems & Libraries

HTML5 Boilerplate

HTML5 ★ BOILERPLATE

[SOURCE CODE](#)

[DOCS](#)

[OTHER PROJECTS](#)

The web's most popular front-end template

HTML5 Boilerplate helps you build **fast**, **robust**, and **adaptable** web apps or sites. Kick-start your project with the combined knowledge and effort of 100s of developers, all in one little package.

[Download v4.2.0](#)

[Get a custom build](#)

[See the CHANGELOG](#)

Twitter Bootstrap

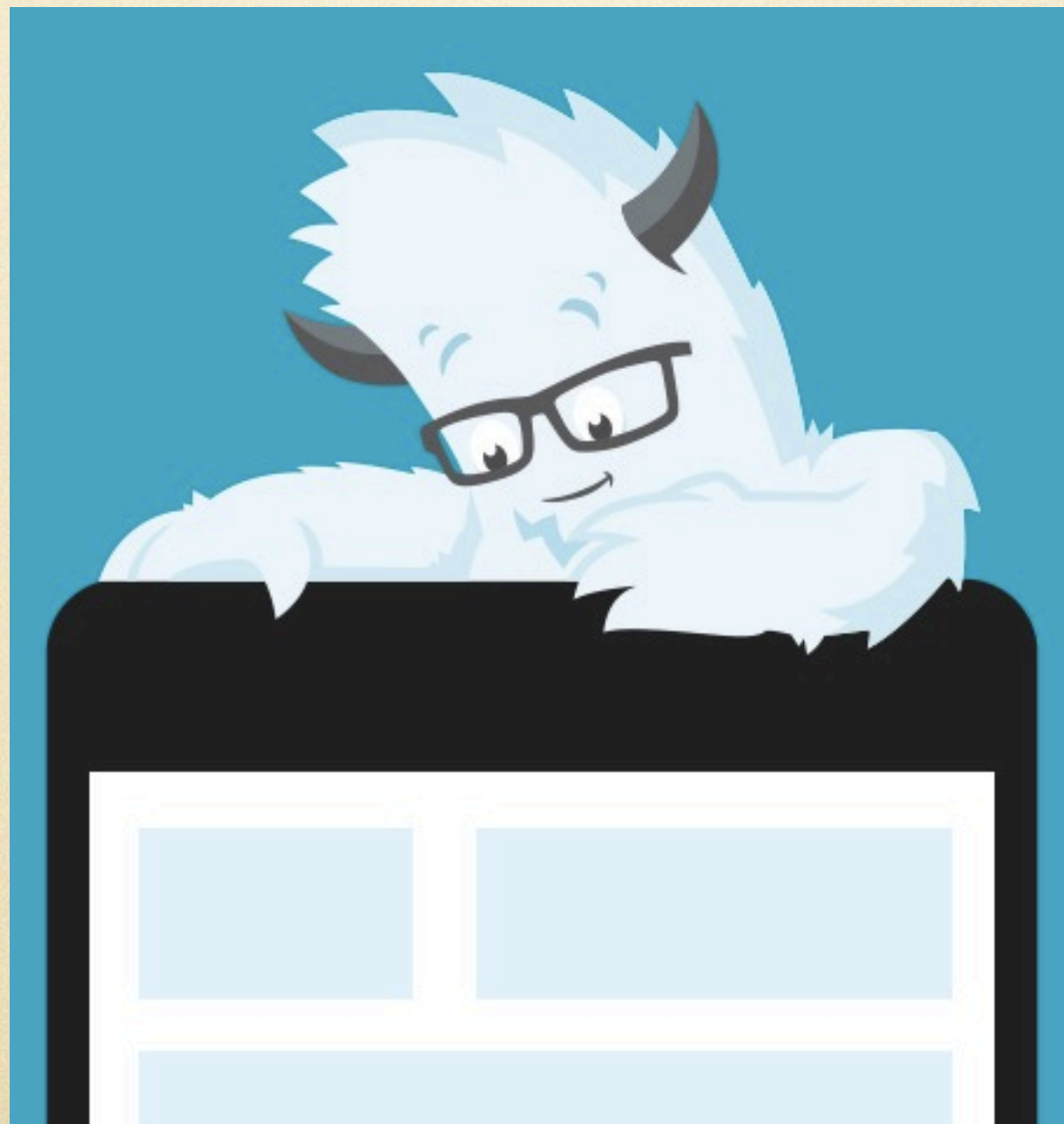
Bootstrap

Sleek, intuitive, and powerful front-end framework for faster and easier web development.

[Download Bootstrap](#)

[GitHub project](#) [Examples](#) [Extend](#) [Version 2.3.2](#)

Zurb Foundation



Rock Hammer

Rock Hammer

A curated project library for [Hammer For Mac](#). Rock Hammer contains baseline typography, plus styling for common HTML elements including images, forms and tables, as well as navigation, responsive modules and widgets

[Download the complete Rock Hammer](#)

[Rock Hammer on Github](#)

Inuit.CSS

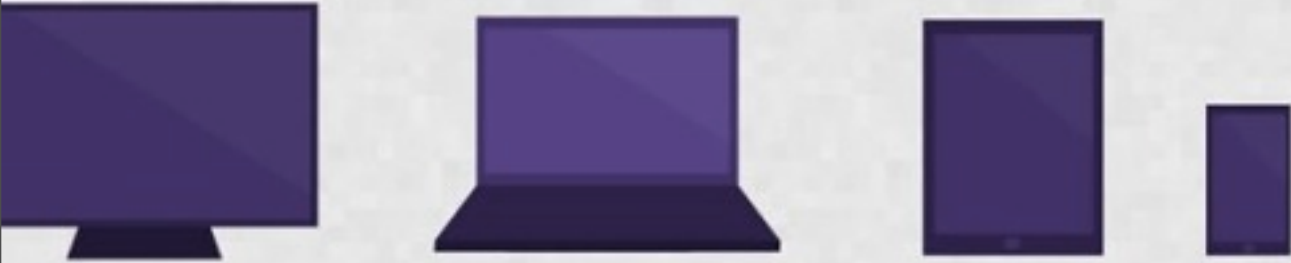


inuitcss

inuit.css – a powerful, scalable, Sass-based, BEM, OOCSS framework.

And many others

Responsive Grid System for your Next Project



Bo
basic

HTML5
Ultra-Lean HTML5,
for Rapid W

Responsive

MIT Open Source

Download

unsemantic

Download Here

What is it?

Un-se-man-tic [ˈʌnsəˈmɑntɪk]

adjective

1. A *unsemantic* word that *developers* use to outgrow the work of their peers.

noun

2. "Now, it's also a CSS framework. I mean hey... "If the shoe fits, wear it." "

Successor to 960.gs

SE-On?

Responsive Demo

Adapt.js Demo

presentational elements, and markup, Starkers is the perfect

- fo Skeleton
- What & Why
- Grid
- Typography
- Buttons
- Forms
- Media Queries
- Support
- Examples
- Download
- License & Log

A Beautiful B
Responsive, I
Development

Gumby

Introducing Gumby 2

Now more flexible than ever

You simply **have** to try it.

Typo-tactically Awesome. Brilliantly Flexible. Incomparably Simple. Unbelievably Robust.

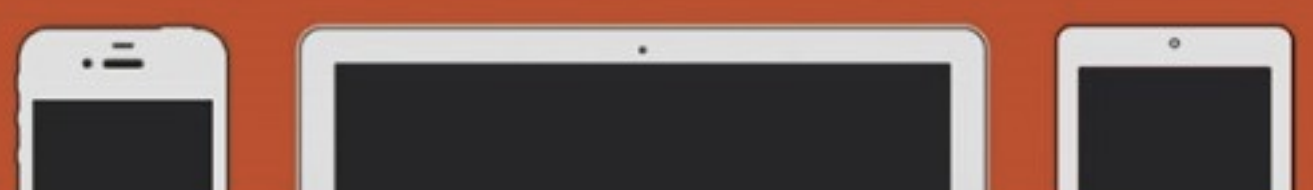
Download Gumby 2

Read the Documentation

BASE

Super Simple Respons
built to work on mobile devices, tablets, net

DOWNLOAD LATEST BUILD VIEW DEMOS AND DOCS



One% CSS G
12 Columns Fluid CSS Grid

(beg), borrow, (steal) code.

Look at everything

```
<!DOCTYPE html>
<!--[if lt IE 7]>      <html class="no-js lt-ie9 lt-ie8 lt-ie7"> <![endif]-->
<!--[if IE 7]>        <html class="no-js lt-ie9 lt-ie8"> <![endif]-->
<!--[if IE 8]>        <html class="no-js lt-ie9"> <![endif]-->
<!--[if gt IE 8]><!--> <html class="no-js"> <!--<![endif]-->
```

```
<!-- The page header typically contains items such as your site heading, logo and possibly the main site navigation -->
<!-- ARIA: the landmark role "banner" is set as it is the prime heading or internal title of the page -->
<header role="banner">
```

```
@mixin font-size($font-size: 16){
  font-size : #{$font-size}px;
  font-size : #{$font-size / 10}rem; }
```

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

Side Nav

Side nav, like you'll see on site or a page.

Link 1
Link 2
Link 3
Link 4

Browser support

inuit.css is a modern framework for modern browsers. It takes advantage of [normalize.css](#) and global `box-sizing:border-box;` (optional). As such, inuit.css is intended for **IE8** and above only. The last release to support IE7 was **v4.1.5**.

- _brand.scss
- _clearfix.scss
- _debug.scss
- _helper.scss
- _mixins.scss
- _normalize.scss
- _pull.scss
- _push.scss
- _reset.scss
- _shared.scss
- _widths.scss

If you like it, use it.

Write your own code

Making it your own

```
<!--[if lt IE 9]> <html class="no-js lt-ie9" lang="en"> <![endif]-->
<!--[if gt IE 8]><!--> <html class="no-js" lang="en"> <!--<![endif]-->
```

```
.an-example {
    // put the IE9 and 'other' browser CSS declarations here

    .lt-ie9 & {

        // put the IE8 and below CSS declarations here
        /* IE8 and below */

        // if you need IE7 as well then prefix the CSS with a *
        /* IE7 and below */

        // if you need IE6 as well then prefix the CSS with a _
        /* IE6 */

    }
}
```

```
// -----
// El Captain Bizarro CSS Reset
// -----

html {
    font-size: $doc-font-size+px;
    -webkit-text-size-adjust: 100%;
    -ms-text-size-adjust: 100%;
}

@-webkit-viewport { width : device-width; }
@-moz-viewport { width : device-width; }
@-ms-viewport { width : device-width; }
@-o-viewport { width : device-width; }
@viewport { width : device-width; }

*, *:before, *:after {
    -webkit-box-sizing: border-box;
    -moz-box-sizing: border-box;
    box-sizing: border-box;
}

article, aside, details, figcaption, figure, footer, header, hgroup, main, nav, section, summary {
    display: block;
}
```

```
<!-- start coding -->
<div id="wrapper">

    <header role="banner">

        <nav role="navigation">

            </nav> <!-- /nav -->

        </header> <!-- /header -->

        <section role="main">

            </section> <!-- /main -->

        <footer role="contentinfo">

            </footer> <!-- /footer -->

        </div> <!-- /#wrapper -->
```


Create your own 'bits'

```
@mixin mq($point, $IE9: false, $query1: min, $query2: width) {  
  
  @if $IE9 == true{  
    // if true then give an IE conditional class 'fallback' for oldIE  
    .lt-ie9 & {  
      @content;  
    }  
    @media screen and (#{ $query1 }-#{ $query2 }: $point / $doc-font-size +em) {  
      @content;  
    }  
  }  
  @else {  
    // if not true (suggest writing - false - for clarity no 'fallback'  
    @media screen and (#{ $query1 }-#{ $query2 }: $point / $doc-font-size +em) {  
      @content;  
    }  
  }  
}
```

```
// box-sizing with IE fallback  
  
@mixin box($width: 'null', $padding: 'null') {  
  padding: 0% + $padding;  
  width: 0% + $width;  
  .lt-ie8 & {  
    padding: 0% + $padding;  
    width: 0% + $width - $padding - $padding;  
  }  
}
```

```
@mixin font-size($size, $margin: yes, $line-height: $doc-line-height) {  
  @if $margin == yes {  
    font-size: 0px + $size;  
    font-size: 0rem + $size / $doc-font-size;  
    line-height: ceil($size / $doc-line-height) * ($doc-line-height / $size);  
    margin-bottom: 0px + $line-height;  
    margin-bottom: 0rem + ($line-height / $doc-font-size);  
  }  
  @if $margin != yes {  
    font-size: 0px + $size;  
    font-size: $size / $doc-font-size + rem;  
    line-height: ceil($size / $line-height) * ($line-height / $size);  
  }  
  
  @if $line-height != $doc-line-height {  
    font-size: 0px + $size;  
    font-size: 0rem + $size / $doc-font-size;  
    line-height: $line-height;  
    margin-bottom: 0px + $line-height;  
    margin-bottom: 0rem + ($line-height / $doc-font-size);  
  }  
  @else {  
    font-size: 0px + $size;  
    font-size: $size / $doc-font-size + rem;  
    margin-bottom: 0px + $margin;  
    margin-bottom: 0rem + $margin / $doc-font-size;  
  }  
}
```


Easier to take away

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


Sass

Documentation

Read existing documents

Principles of writing consistent, idiomatic CSS

The following document outlines a reasonable style guide for CSS development. These guidelines strongly encourage the use of existing, common, sensible patterns. They should be adapted as needed to create your own style guide.

This is a living document and new ideas are always welcome. Please contribute.

1. General principles

"Part of being a good steward to a successful project is realizing that writing code for yourself is a Bad Idea™. If thousands of people are using your code, then write your code for maximum clarity, not your personal preference of how to get clever within the spec." - Idan Gazit

- Don't try to prematurely optimize your code; keep it readable and understandable.
- All code in any code-base should look like a single person typed it, even when many people are contributing to it.
- Strictly enforce the agreed-upon style.
- If in doubt when deciding upon a style use existing, common patterns.

Contents

- CSS document anatomy
 - General
 - One file vs. many files
 - Table of contents
 - Section titles
- Source order
- Anatomy of rulesets
- Naming conventions
 - JS hooks
 - Internationalisation
- Comments
 - Comments on steroids
 - Quasi-qualified selectors
 - Tagging code
 - Object/extension pointers
- Writing CSS
- Building new components
- OOCSS
- Layout
- Sizing UIs
 - Font sizing
- Shorthand
- IDs

Write your own rules

Always a space after a property's colon

```
display: block;
```

not

```
display:block;
```

To aid in writing your CSS without referring to your HTML add the code snippet within this CSS Comment for ease and clarity.

```
/*  
 * @description: /  
 *  
 * @author: / Stuart Robson  
 * @version: / 0.1  
 * @date: / dd/mm/yyyy  
 *  
 *  
 * @example: /  
 *  
 *  
 * @type: /  
 * @notes: /  
 * @tags: / ^ ^ ^ ^ ^  
 */
```

Nesting Media Queries with Nested Elements

It's therefore only progression to suggest where you would have elements nested within elements that also require media queries. For consistency rather than writing media queries after 'every' element in a nested group of elements. Group the media queries at the end of the element. Like this -

```
.element {  
  background: blue;  
  display: block;  
  margin: 0 auto;  
  width: 44.15467%;  
  h1 {  
    color: green;  
    font-size: 32px;  
    font-size: 2rem;  
  }  
  .box {  
    width: 80%;  
    margin: 0 auto;  
  }  
  @media (min-width: 700px) {  
    width: 100%;  
    h1 {  
      font-size: 48px;  
      font-size: 3rem;  
    }  
    .box {  
      width: 95%;  
    }  
  }  
  @media (min-width: 900px) {  
    h1 {  
      font-size: 32px;  
      font-size: 2rem;  
    }  
  }  
}
```

Pseudo & Combinator Selectors

After this we place the Pseudo selectors followed by the combinator selectors, like this

```
.element {  
  background-color: #090000;  
  
  @extend .rounded-border;  
  
  @include data-module;  
  
  cursor: pointer;  
  height: 300px;  
  margin: 0 auto;  
  padding: 0;  
  width: 400px;  
  
  &:hover {  
    text-decoration: underline;  
  }  
  
  & > li {  
    margin: 10px;  
  }  
  
  & + ul {  
    margin-left: 10px;  
  }  
}
```


ITERATE

Thanks

Making your code delicious

t: @sturobson

w: alwaystwisted.com



