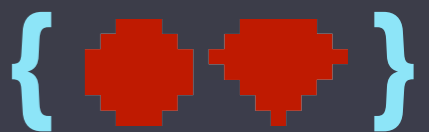


WAH LAU!

CSS CAN BE
TESTED TOO!

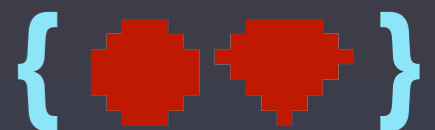


PRESENTED BY

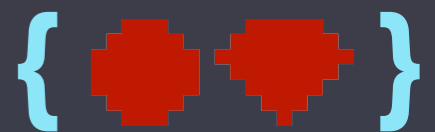
WINSTON TEO

@WINSTONYW

SOFTWARE ENGINEER @ { NEW CONTEXT }

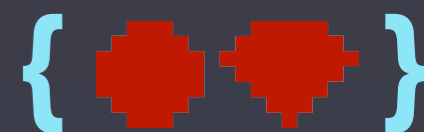


DO YOU
WRITE CSS?

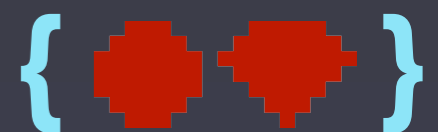




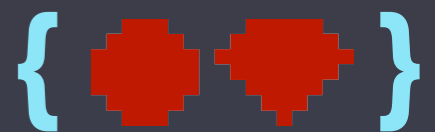
OR



WINSTON
LOVES CSS

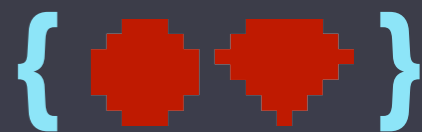


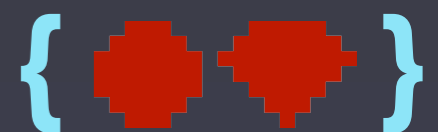
BUT
WE HAVE A PROBLEM



My stylesheets
are
always in a

MESS





BEAUTIFUL

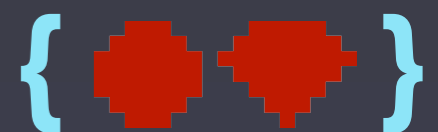


WEBSITE



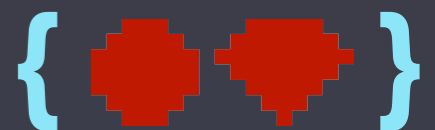
UGLY

CSS

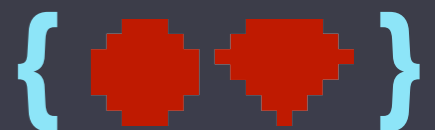


CSS FRAMEWORKS

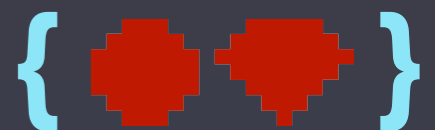
ARE GREAT WAYS TO WRITE CSS



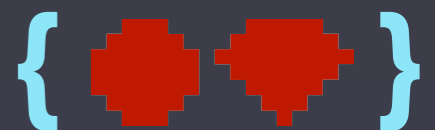
CSS FRAMEWORKS ARE NOT SOLUTIONS



CSS FRAMEWORKS ARE PRONE TO ABUSE



FORGOTTEN
\$VARIABLES
AND
@MIXINS



NESTINGS {

NESTINGS {


NESTINGS {

NESTINGS { AND MORE NESTINGS.. }

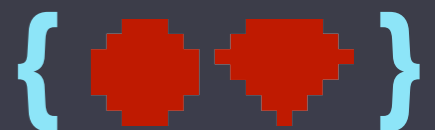
}

}

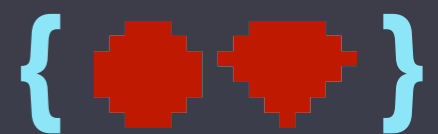
}

{  }

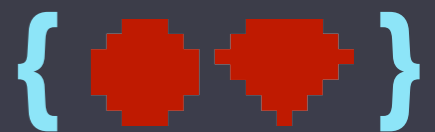
WHAT'S THE
SOLUTION?



TEST
YOUR
CSS!

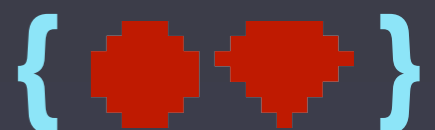


HUH?

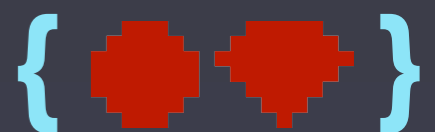


BENEFITS OF TESTING YOUR SOFTWARE

- ENSURES CODE QUALITY
- REDUCES COST OF CHANGE
- MAKES SOFTWARE ENGINEERING FUN



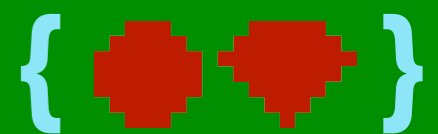
WHAT'S SO
FUN?



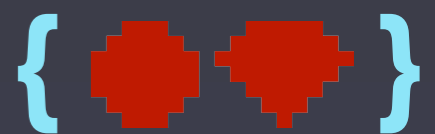
RED



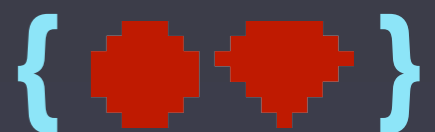
GREEN



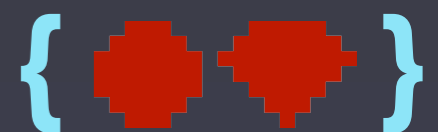
REFACTOR



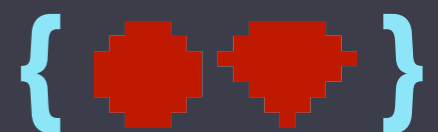
REFACTOR CSS WITH CONFIDENCE



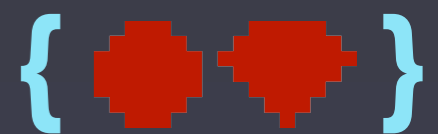
WE ARE ALREADY
TESTING
RUBY, HTML AND JS



WHY NOT CSS?



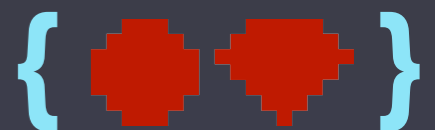
HOW?

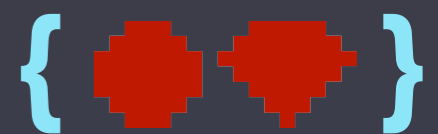


VISUAL INSPECTION

REQUIRES REVIEW OF EVERY SINGLE PAGE

- ADD/UPDATE SOME CSS
- REFRESH THE PAGE
- REPEAT
- CHECK ALL PAGES

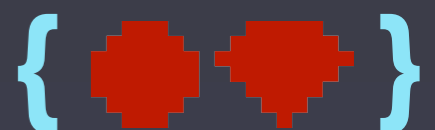




STYLE GUIDE

REQUIRES REVIEW OF ONLY ONE PAGE

- MODULAR APPROACH TO BUILDING CSS
- REUSE OF CSS STYLES



TWITTER

<http://twitter.github.com/bootstrap/>

Overview

Scaffolding

Base CSS

Components

Javascript plugins

Using LESS

Customize

Examples

Bootstrap

Bootstrap, from Twitter

Simple and flexible HTML, CSS, and Javascript for popular user interface components and interactions.

[View project on GitHub](#)

[Download Bootstrap \(v2.0.3\)](#)

[Upgrading from 1.4](#) [Download with docs](#) [Read the blog](#) [Submit issues](#) [Roadmap and changelog](#)

[Watch](#) 28,857 [Fork](#) 5,633 [Follow @twbootstrap](#) 23.4K followers [Tweet](#) 44.4K

Designed for everyone, everywhere.

Need reasons to love Bootstrap? Look no further.

Built for and by nerds

Like you, we love building awesome products on the web. We love it so much, we decided to help people just like us do it easier, better, and faster. Bootstrap is built for you.

For all skill levels

Bootstrap is designed to help people of all skill levels—designer or developer, huge nerd or early beginner. Use it as a complete kit or use to start something more complex.

Cross-everything

Originally built with only modern browsers in mind, Bootstrap has evolved to include support for all major browsers (even IE?!), and, with Bootstrap 2, tablets and smartphones, too.

12-column grid

Grid systems aren't everything, but having a durable and flexible one at the core of your work can make development much simpler. Use our built-in grid classes or roll your own.

Responsive design

With Bootstrap 2, we've gone fully responsive. Our components are scaled according to a range of resolutions and devices to provide a consistent experience, no matter what.

Styleguide docs

Unlike other front-end toolkits, Bootstrap was designed first and foremost as a styleguide to document not only our features, but best practices and living, coded examples.

Growing library

Despite being only 10kb (gzipped), Bootstrap is one of the most complete front-end toolkits out there with dozens of fully functional components ready to be put to use.

Custom jQuery plugins

What good is an awesome design component without easy-to-use, proper, and extensible interactions? With Bootstrap, you get custom-built jQuery plugins to bring your projects to life.

Built on LESS

Where vanilla CSS falters, LESS excels. Variables, nesting, operations, and mixins in LESS makes coding CSS faster and more efficient with minimal overhead.

HTML5

Built to support new HTML5 elements and syntax.

CSS3

Progressively enhanced components for ultimate style.

Open-source

Built for and maintained by the community via [GitHub](#).

Made at Twitter

Brought to you by an experienced [engineer](#) and [designer](#).

GITHUB

<https://github.com/styleguide/css>

github

[Signup and Pricing](#) [Explore GitHub](#) [Features](#) [Blog](#) [Login](#)

Overview

Styling (CSS)

Markup (HTML)

Behavior (JavaScript)

Ruby

0. Overview

1. Buttons

2. Forms

3. Source Code

4. Text Styling

5. Listings

6. Boxed Groups

7. Icons

1.1

buttons.scss

Classy buttons are big clickable buttons, great for form actions and primary page actions. This is used extensively around the site.

- **:hover** - Buttons love hover states.
- **:disabled** - For when you shouldn't be able to click on it.
- **.disabled** - Same as the disabled pseudo-class.
- **.primary** - For the main form action, use whenever creating something.
- **.primary:hover** - Hover state fo primary buttons.
- **.mousedown** - A brief moment of polish when the button is pressed.
- **.primary.mousedown** - Even primary buttons have a down state.
- **.danger** - This button is going to do something potentially bad or destructive.
- **.danger:hover** - Hover state for danger buttons.
- **.danger.mousedown** - Depressed danger button.
- **.danger:disabled** - Disabled danger button.

Button (button.classy)

Button (a.button.classy)

Button (button.classy)

Button (a.button.classy)

:hover

Button (button.classy)

Button (a.button.classy)

:disabled

disabled

BBC

<http://www.bbc.co.uk/gel/>

BBC News Sport Weather iPlayer TV Radio More Search

GEL

 Global Experience Language

Home | Philosophy | Web | TV | Mobile | Tablet | Accessibility

An introduction to GEL

GEL Authentic, Current, Pioneering, Modern British, Local/Global, Compelling, Distinctive, Joined-up, Universal, Best.

Building a Global Experience Language for the BBC

We are evolving a global experience language for the BBC's digital services. The GEL guidelines are a reference point for designing BBC services across Web, Mobile, IPTV and Tablet.

“This is probably the best job in the world!”
Neville Brody, Research Studios

Featured Pattern

Carousel

Download GEL Web Styleguide

Download GEL Assets

- Download core GEL components
- Download GEL toolbar guide

BUILDING BLOCKS

Typography

Large bold type should be used to establish a clear information hierarchy.

FOUNDATIONS

Universal Grid

Your starting point is a universal grid, divided into 61 x 16px vertical units.

FOUNDATIONS

The Masthead

The global masthead retains the current global navigation links with additional links.

PATTERNS

Overlay Panels

The overlay panel is an interaction pattern that is used to shift the user's focus from the main page to a single element or group of elements.

PATTERNS

Accordion

The accordion is used when a large amount of

BBC

<http://itservices.stanford.edu/service/web/design/styleguide/modern>

HTML & CSS Style Guide (Stanford Modern)

Home » University Web Resources » Stanford Self-help Web Design Resources » HTML & CSS Style Guide (Stanford Modern)

On this page:

[Validators](#) | [CSS](#) | [Icons](#) | [Paragraphs](#) | [Horizontal Rules](#) | [Links to Documents](#) | [Links that Open Windows](#) | [Images](#) | [Tables](#) | [Lists](#) | [Font Sizes](#) | [Headings](#)

[Documentation, Design, and Development \(DDD\)](#) has developed the current Stanford web templates to comply strictly with [XHTML 1.0](#) and [CSS \(Cascading Style Sheets\)](#) standards set by the [W3C \(World Wide Web Consortium\)](#). CSS allows Web authors to address the graphical layout of a page separately from its content and logical structure. This makes for cleaner, easier-to-update HTML code and the potential to view single HTML files in multiple formats, as appropriate for different contexts and devices.

Tools for Validating Code

- [W3C XHTML Validation Service](#)
- [W3C CSS Validation Service](#)
- [Total Validator \(Accessibility Validation\)](#)

Cascading Style Sheets

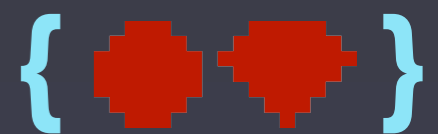
The [CSS 2.0 style sheets](#) that define layout, sizes, colors, etc. for the [web templates](#) are available for preview.

Sidebar Icons

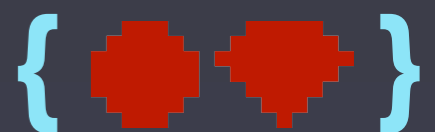
To invoke these styles you must label your navigation lists `<ul class="calendar">` or paragraphs `<p class="calendar">` with one of these specific class names:



ENOUGH?



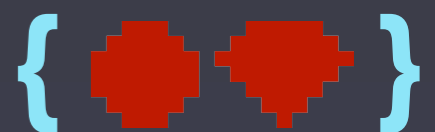
THAT'S NOT TESTING



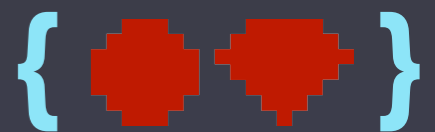
PROOF OF CONCEPT

CACTUS

CSS TESTING FRAMEWORK



GITHUB.COM/WINSTON/CACTUS



CACTUS GEM

TESTS YOUR CSS WITH JAVASCRIPT

- **ASSERT CSS STYLES**

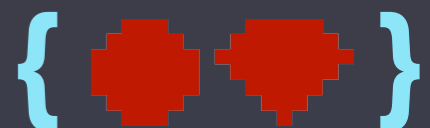
```
Cactus.expect(".header", "width").toEqual("800px");  
Cactus.expect(".header", "height").toEqual("200px");
```

- **PASSES AND FAILURES IN DEV AND TEST ENV**

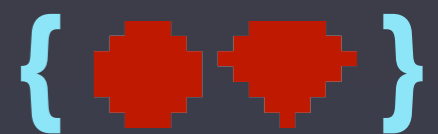
Cactus 48 specs, 0 failures

[Hide Passes](#)

Expected \$(' .alert')[0] margin-top to equal 10px. Got 10px.



DEMO TIME



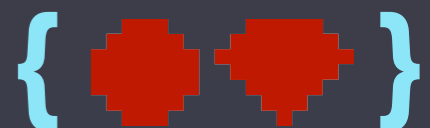
My Style Guide

.alert

.notice

```
/* assets/stylesheets/flash.scss */
```

```
.alert {  
  padding: 10px;  
  margin: 10px 0;  
  @include border-radius(10px);  
  font-size: 20px;  
  
  border: 1px solid red;  
  background: lighten(red, 40%);  
}  
  
.notice {  
  padding: 10px;  
  margin: 10px 0;  
  @include border-radius(10px);  
  font-size: 20px;  
  
  border: 1px solid green;  
  background: lighten(green, 40%);  
}
```



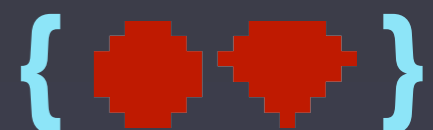

```
/* public/cactus_spec/application_spec.js */

//////////
// .alert

Cactus.expect(".alert").toHaveMargin("10px 0px");
Cactus.expect(".alert").toHavePadding("10px");
Cactus.expect(".alert").toHaveBorderWidth("1px");

Cactus.expect(".alert").toHaveBorderColor("#FF0000");
Cactus.expect(
  ".alert", "background-color"
).toHaveColor("#FFCCCC");

Cactus.expect(".alert p", "font-size").toEqual("20px");
```



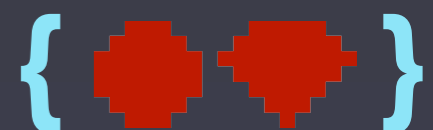
```
/* public/cactus_spec/application_spec.js */

//////////
// .notice

Cactus.expect(".notice").toHaveMargin("10px 0px");
Cactus.expect(".notice").toHavePadding("10px");
Cactus.expect(".notice").toHaveBorderWidth("1px");

Cactus.expect(".notice").toHaveBorderColor("#008000");
Cactus.expect(
  ".notice", "background-color"
).toHaveColor("#4DFF4D");

Cactus.expect(".notice p", "font-size").toEqual("20px");
```



```
/* assets/stylesheets/flash_refactored_1.scss */

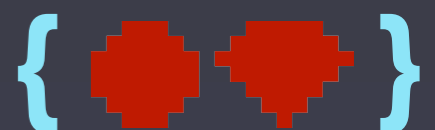
@mixin flash {
  padding: 10px;
  margin: 10px 0;
  @include border-radius(10px);
  font-size: 20px;
}

.alert {
  @include flash;

  border: 1px solid red;
  background: lighten(red, 40%);
}

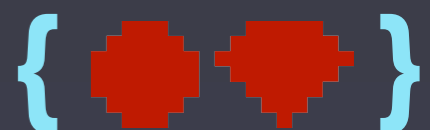
.notice {
  @include flash;

  border: 1px solid green;
  background: lighten(green, 40%);
}
```



```
/* assets/stylesheets/flash_refactored_2.scss */
```

```
.round-box {  
  padding: 10px;  
  margin: 10px 0;  
  border: 1px solid;  
  @include border-radius(10px);  
}  
.large-font {  
  font-size: 20px;  
}  
  
.alert {  
  @extend .round-box;  
  @extend .large-font;  
  
  border-color: red;  
  background: lighten(red, 40%);  
}  
  
.notice {  
  @extend .round-box;  
  @extend .large-font;  
  
  border-color: green;  
  background: lighten(green, 40%);  
}
```



```
/* public/cactus_spec/application_spec.js */

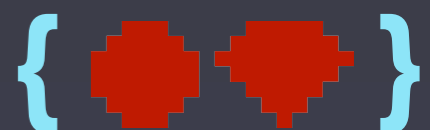
//////////
// Refactored

function test_round_box(selector) {
  Cactus.expect(selector).toHaveMargin("10px 0px");
  Cactus.expect(selector).toHavePadding("10px");
  Cactus.expect(selector).toHaveBorderWidth("1px");
}

function test_large_font(selector) {
  Cactus.expect(selector, "font-size").toEqual("20px");
}

test_round_box(".alert");
test_large_font(".alert p");
Cactus.expect(".alert").toHaveBorderColor("#FF0000");
Cactus.expect(".alert", "background-color").toHaveColor("#FFCCCC");

test_round_box(".notice");
test_large_font(".notice p");
Cactus.expect(".notice").toHaveBorderColor("#008000");
Cactus.expect(".notice", "background-color").toHaveColor("#4DFF4D");
```



```
/* views/application/index.haml */

.profile
  = image_tag "p1.jpg", alt: "Winston Teo"
  .name Winston Teo

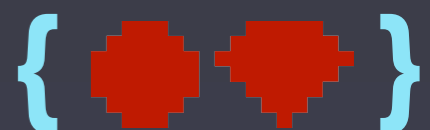
/* public/cactus_spec/application_spec.js */

Cactus.expect(".profile", "width").toEqual("200px");
test_round_box(".profile");
test_large_font(".profile .name");

/* assets/stylesheets/profile.scss */

.profile {
  width      : 200px;
  @extend .round-box;
  text-align: center;

  .name {
    @extend .large-font;
  }
}
```



```
/* views/application/index.haml */

.profile
  = image_tag "p1.jpg", alt: "Winston Teo"
  .name Winston Teo
  %p Smart and Handsome

/* public/cactus_spec/application_spec.js */

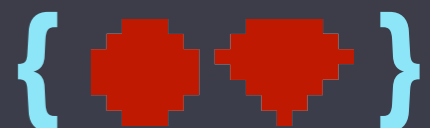
Cactus.expect(".profile", "width").toEqual("200px");
test_round_box(".profile");
test_large_font(".profile .name");
Cactus.expect(".profile p", "font-size").toEqual("12px");

/* assets/stylesheets/profile.scss */

.profile {
  width      : 200px;
  @extend .round-box;
  text-align: center;

  .name {
    @extend .large-font;
  }
}

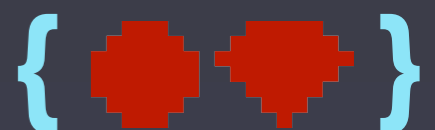
p { font-size: 12px;}
```



RUN SPECS

~ FAIL ~

MAKE SPECS PASS




```
/* views/application/index.haml */

.profile
  = image_tag "p1.jpg", alt: "Winston Teo"
  .name Winston Teo
  %p Smart and Handsome

/* public/cactus_spec/application_spec.js */

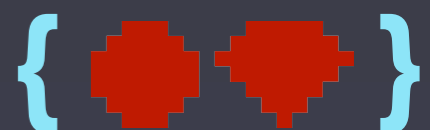
Cactus.expect(".profile", "width").toEqual("200px");
test_round_box(".profile");
test_large_font(".profile .name");
Cactus.expect(".profile p", "font-size").toEqual("12px");

/* assets/stylesheets/profile.scss */

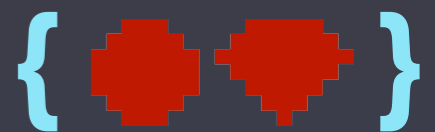
.profile {
  width      : 200px;
  @extend .round-box;
  text-align: center;

  .name {
    @extend .large-font;
  }

  p { font-size: 12px;}
}
```



AUTOMATED TESTS



```
/* spec/spec_helper.rb */

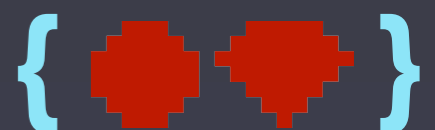
RSpec::Matchers.define :be_cactus do
  match do |actual|
    all(".cactus_fail").blank?
  end

  failure_message_for_should do |actual|
    message = "Oei! Something's wrong with the CSS on
              '#{actual.current_url} lah!'\n"
    all(".cactus_fail").each do |failure|
      message += "- #{failure.text}\n"
    end
    message
  end
end

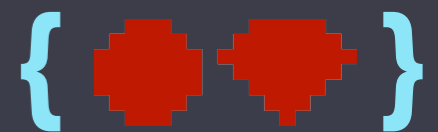
/* spec/requests/cactus_spec.rb */

require 'spec_helper'

describe 'rspec and capybara integration with cactus', js: true do
  it "is cactus-ready " do
    visit root_path
    page.should be_cactus
  end
end
```



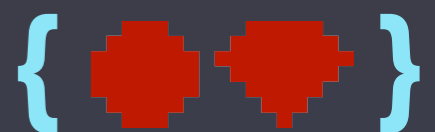
TDD?



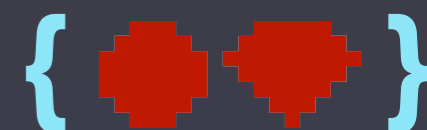
CACTUS WORKFLOW

THE NOT-SO-TDD WAY

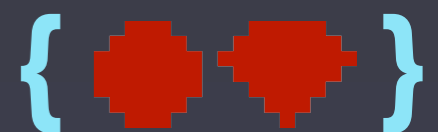
- ITERATE ON STYLEGUIDE
- LOCKDOWN STYLEGUIDE
- WRITE CACTUS SPECS AROUND STYLEGUIDE
- ~ REGRESSION TESTING



POSSIBILITIES



**ARE YOU
CONVINCED?**



**WAH LAU!
CSS CAN BE TESTED TOO!**

THANK YOU
@WINSTONYW

