CSS Basics

Juan “Harek” Urrios
@xharekx33

What is CSS?

CSS refers to Cascading Style Sheets and is designed primarily to enable the separation of content from presentation.

While presentation will still be dependent on structure – this is pretty much unavoidable – separating it from content serves primarily to simplify any change from a slight design adjustment to a full-fledged redesign, and makes it a breeze to add or update things while maintaining presentational consistency throughout the site.

It will also allow us to optimize how a page is styled for different rendering methods, such as on-screen (and their variations...) , in print, on screen readers and tactile devices (used by blind users), etc.

IRON HACK

The power of CSS

A great way to witness the power of css is to visit www.csszengarden.com and see how the same exact markup can produce strikingly different results just by changing an external CSS file.

It's almost as if they were completely different sites.

IRON HACK

How to use CSS in your website

Include your CSS file (or files) in your document's <head>. Now when you load the page all the styles you define in that file will replace the browser's defaults.

FOLDERS
- MyApp
  - app
  - bin
  - config
  - db
  - lib
  - log
  - public
  - test
  - tmp
  - vendor
  - .gitignore
  - config.ru
  - Gemfile
  - Gemfile.lock
  - Rakefile
  - README.rdoc

```
<!DOCTYPE html>
<html lang="en">
  <head>
    ...
    <link rel="stylesheet" href="theme.css">
  </head>
  <body>
    ...
```

IRON HACK

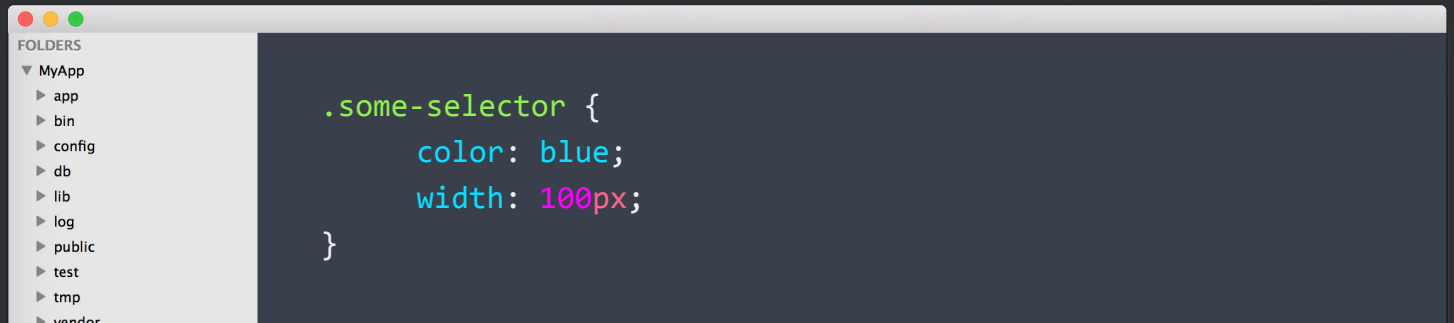
@xharekx33

CSS basic syntax

The syntax in CSS very simple, it consists of only 3 parts:

```
selector { property: value; }
```

For each selector you can add as many property/value pairs as needed. The selector targets the object that will be styled and the properties inside apply the actual styles.

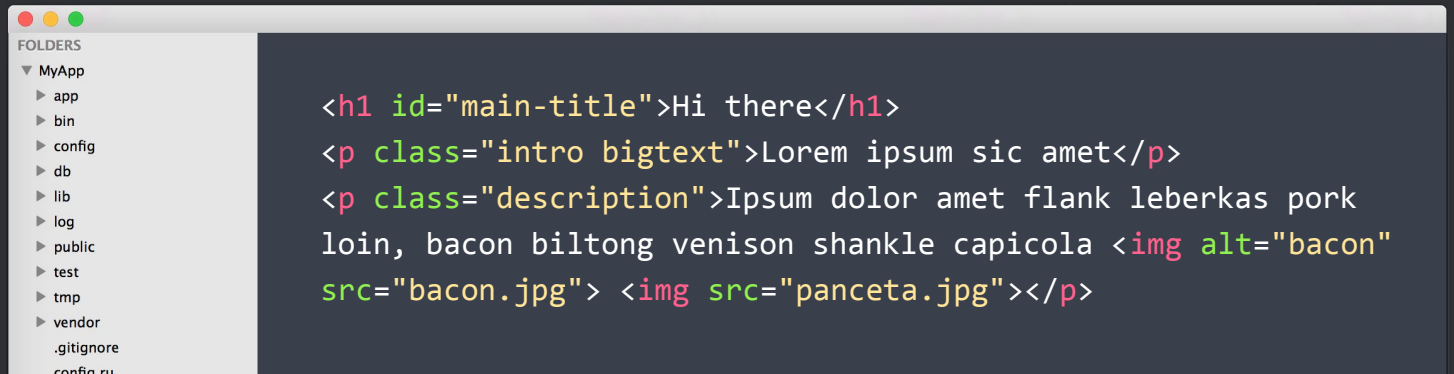


@xharekx33

Selectors

Different selectors allow you to:

- Target elements directly by their HTML tag: `p`, `h1`, `ul`, `div`...
- Target elements with a class or classes: `.class-name`, `.class.class2`
- Target the element with a certain id: `#element-id`
- Target the element with a specific attribute value: `.class[title="My title"]`

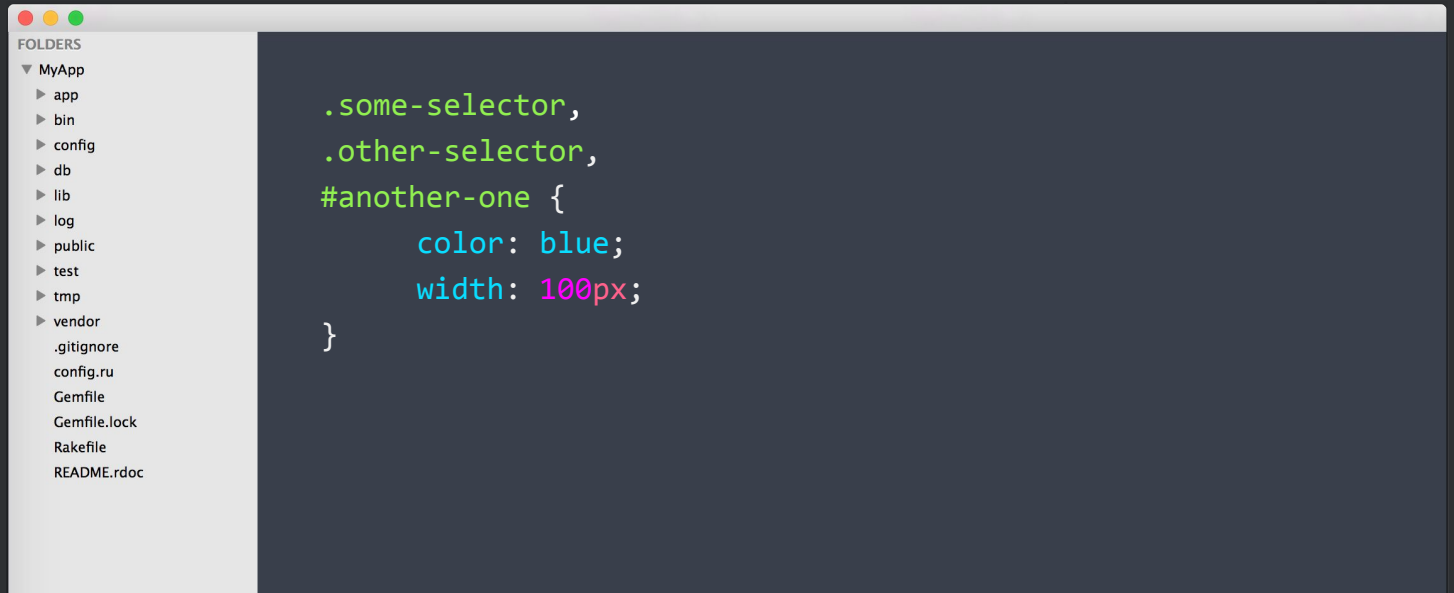


IRON HACK

@xharekx33

Chaining selectors

If you want to apply the same style to different elements, you can chain their respective selectors instead of repeating code.



Combining selectors

- Nested selectors: `.post a`, `.post p a`
- Direct children: `.post > a`
- Adjacent sibling: `h2 + p`
- General siblings: `h2 ~ p`

FOLDERS
- MyApp
  - app
  - bin
  - config
  - db
  - lib
  - log
  - public
  - test
  - tmp
  - vendor
  - .gitignore
  - config.ru
  - Gemfile
  - Gemfile.lock
  - Rakefile
  - README.rdoc

```
<h1 id="main-title">My site name</h1>
<h2>Some tagline</h2>
<article class="post">
  <h1><a href="permalink.html">Title for my post</a></h1>
  <p class="awesome-p">Lorem ipsum sic amet <a href="url.com">uber<a>
rosae</p>
  <p>Ipsum dolor amet flank leberkas pork loin, bacon biltong venison
shankle capicola</p>
  <a href="permalink.html">Read more...</a>
</article>
```

IRON HACK

@xharekx33

Pseudo-classes

A CSS pseudo-class is a keyword added to selectors that specifies a special state of the element to be selected, in relation to factors like the history of the navigator, the status of its content, or the position of the mouse:

`a:hover`, `a:visited`, `input:checked`, `input:focus`, `p:first-child`, `li:last-of-type`...

See them all here: https://developer.mozilla.org/en-US/docs/Web/CSS/pseudo-classes

IRON HACK

@xharekx33

Pseudo-elements

CSS pseudo-elements are added to selectors but instead of describing a special state, they allow you to style certain parts of a document separately, or insert content for certain situations:

`blockquote::before`, `p::first-letter`, `::selected`, `p::selected`...

See them all here: https://developer.mozilla.org/en-US/docs/Web/CSS/Pseudo-elements

IRON HACK

Selectors practice

Practice your selectors with CSS Diner

Inheritance

When you define styles for some element, all elements nested inside will inherit properties assigned to it, unless they are modified explicitly.

These properties tend to be those that deal with color, typography, list styles, text alignment and indentation and visibility, such as: `color`, `cursor`, `font-family`, `font-weight`, `line-height`, `list-style`, `text-align`, `visibility`...

An partial and slightly outdated list can be found here: http://stackoverflow.com/questions/5612302/which-css-properties-are-inherited

The CSS Box model

In a document, each element is represented as a rectangular box. Each of these rectangular boxes is described using the standard box model that consists of content, padding, border and margin.

The CSS Box model: content area

The content area is the area containing the real content of the element. The CSS properties `width`, `min-width`, `max-width`, `height`, `min-height` and `max-height` control the its size. It can also have a `background` (color or image)

The CSS Box model: content area

FOLDERS
- MyApp
  - app
  - bin
  - config
  - db
  - lib
  - log
  - public
  - test
  - tmp
  - vendor
  - .gitignore
  - config.ru
  - Gemfile
  - Gemfile.lock
  - Rakefile
  - README.rdoc

```
.box {
      width: 300px;
      height: 100px;
      background: white url("img.png") no-repeat left top;
}
```

IRON HACK

@xharekx33

The CSS Box model: padding area

The padding area extends the content area (and its background), separating the content from the border. Its size is controlled using `padding-top`, `padding-right`, `padding-bottom`, `padding-left` or the shorthand `padding`.

The CSS Box model: padding area

FOLDERS
- MyApp
  - app
  - bin
  - config
  - db
  - lib
  - log
  - public
  - test
  - tmp
  - vendor
  - .gitignore
  - config.ru
  - Gemfile
  - Gemfile.lock
  - Rakefile
  - README.rdoc

```
.box {
    ...
    padding: 10px 20px 10px 20px;
}
```

The CSS Box model: border area

The border area add a border of a certain `border-width` to the padding area. Its width can also be defined using the shorthand `border` property (along with the style and color of the border)

The CSS Box model: border area

FOLDERS
- MyApp
  - app
  - bin
  - config
  - db
  - lib
  - log
  - public
  - test
  - tmp
  - vendor
  - .gitignore
  - config.ru
  - Gemfile
  - Gemfile.lock
  - Rakefile
  - README.rdoc

```
.box {
      ...
      border: 2px solid red;
}
```

IRON HACK

@xharekx33

The CSS Box model: margin area

The margin area extends the border area with an empty area used to separate the element from its neighbors. Its size is controlled using the `margin-top`, `margin-right`, `margin-bottom`, `margin-left` or the shorthand `margin`.

The CSS Box model: margin area

FOLDERS
- MyApp
  - app
  - bin
  - config
  - db
  - lib
  - log
  - public
  - test
  - tmp
  - vendor
  - .gitignore
  - config.ru
  - Gemfile
  - Gemfile.lock
  - Rakefile
  - README.rdoc

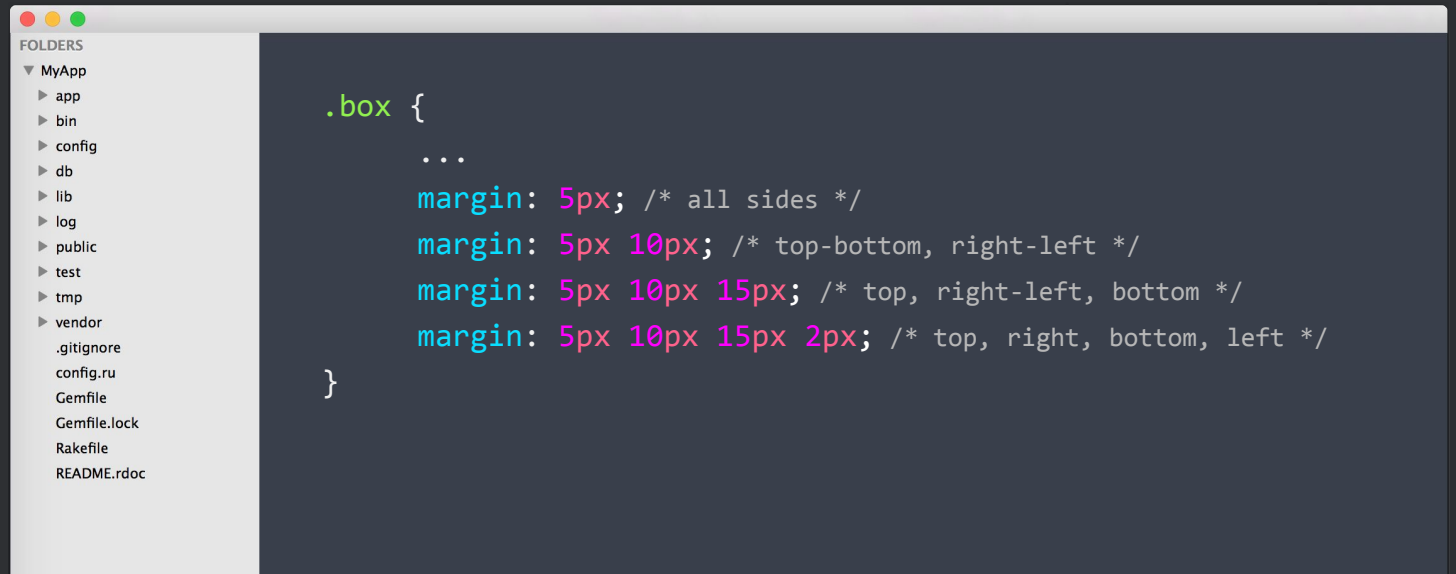
```
.box {
      ...
      margin: 5px 10px 5px 10px;
}
```

IRON HACK

@xharekx33

Margin and padding: shorthand notation

When using shorthand notation for the `padding` and `margin` properties, values for each side can be assigned like this:




```
.box {
    ...
    margin: 5px; /* all sides */
    margin: 5px 10px; /* top-bottom, right-left */
    margin: 5px 10px 15px; /* top, right-left, bottom */
    margin: 5px 10px 15px 2px; /* top, right, bottom, left */
}
```

IRON HACK

@xharekx33

Positioning elements

The `position` property specifies the type of positioning method used for an element. Possible values are: `static`, `relative`, `fixed` and `absolute`

FOLDERS
- MyApp
  - app
  - bin
  - config
  - db
  - lib
  - log
  - public
  - test
  - tmp
  - vendor
  - .gitignore
  - config.ru
  - Gemfile
  - Gemfile.lock
  - Rakefile
  - README.rdoc

```
.box {
      ...
      position: absolute;
      top: 10px;
      left: 50%;
      z-index: 20;
}
```

@xharekx33

Exercise

Get the HTML code from CSSPlayground and:

- Follow the instructions for each div
- Use the correct selector that indicates their color to change their background
- Add a margin of 10px to all boxes
- Add a red border to blue divs
- Add a blue border to red divs
- Use shorthand notation to give the direction h3 a padding of 20px on all sides and see what happens.

IRON HACK

Floated elements

Floated elements are taken out from the normal flow and placed along the left or right side of their container, where text and inline elements will wrap around them.

This elements won't be included in their container's content height unless a clearfix is applied.

To stop content from flowing alongside floated elements we will have to clear them

Floated elements

FOLDERS
- MyApp
  - app
  - bin
  - config
  - db
  - lib
  - log
  - public
  - test
  - tmp
  - vendor
  - .gitignore
  - config.ru
  - Gemfile
  - Gemfile.lock
  - Rakefile
  - README.rdoc

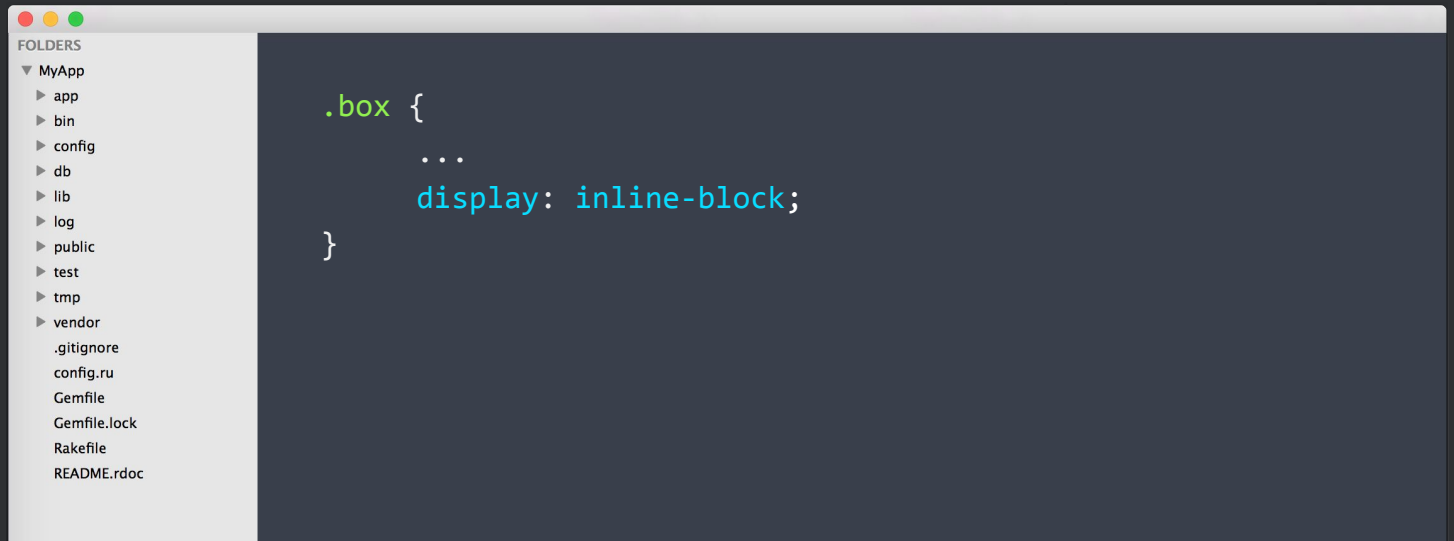
```
.floated {
      float: left;
}
.cleared {
      clear: both;
}
.container:after { /* one of many clearfixes... */
      content: "";
      display: table;
      clear: both;
}
```

IRON HACK

@xharekx33

Display properties

The `display` property can be used to override the default type of rendering box used for an element. Possible values include: `none`, `inline`, `block`, `inline-block` and `list-item` among others.



@xharekx33

Typography

The typography (the font used) in an element can be adjusted using:

FOLDERS
- MyApp
  - app
  - bin
  - config
  - db
  - lib
  - log
  - public
  - test
  - tmp
  - vendor
  - .gitignore
  - config.ru
  - Gemfile
  - Gemfile.lock
  - Rakefile
  - README.rdoc

```
.cool-fonts {
        font-size: 2em;
        line-height: 1.5;
        text-align: center;
        color: black;
        font-family: myFont, "Helvetica Neue", Arial, sans-serif;
}

@font-face {
    font-family: myFont;
    src: url(myfont.woff) format(woff),
         url(myfont.ttf) format(ttf);
}
```

IRON HACK

@xharekx33

Box-shadow

The box-shadow property casts a drop shadow from the frame of almost any element. Multiple box shadows can be applied to the same element (the first specified shadow will be on top)

FOLDERS
- MyApp
  - app
  - bin
  - config
  - db
  - lib
  - log
  - public
  - test
  - tmp
  - vendor
  - .gitignore
  - config.ru
  - Gemfile
  - Gemfile.lock
  - Rakefile
  - README.rdoc

```
.div-with-shadow {
      box-shadow: inset 3px 3px 5px 5px red, -1px 0 5px
olive;
}

/* inset x-offset y-offset blur spread color */
```

IRON HACK

@xharekx33

Text-shadow

The `text-shadow` property adds shadows to text. Multiple shadows can be applied to the same text too as it happens with box-shadow.

FOLDERS
- MyApp
  - app
  - bin
  - config
  - db
  - lib
  - log
  - public
  - test
  - tmp
  - vendor
  - .gitignore
  - config.ru
  - Gemfile
  - Gemfile.lock
  - Rakefile
  - README.rdoc

```
.text-with-shadow {
      text-shadow: 3px 3px 5px red;
}

/* x-offset y-offset blur color */
```

@xharekx33

Border-radius

The `border-radius` property defines how rounded the corners of an element are.

FOLDERS
- MyApp
  - app
  - bin
  - config
  - db
  - lib
  - log
  - public
  - test
  - tmp
  - vendor
  - .gitignore
  - config.ru
  - Gemfile
  - Gemfile.lock
  - Rakefile
  - README.rdoc

```
.rounded-rectangle {
      border-radius: 3px 3px 5px 5px;
}

/*  top-left, top-right, bottom-right, bottom-left */
```

IRON HACK

@xharekx33

Exercise part 2

Continue working on the code from the previous exercise:

- Float the box class to the left and see what happens
- Select the green class and give it a green background
- Give #top a z-index to it appears on top of the other elements
- Give top a box-shadow.

IRON HACK

@xharekx33

CSS units

You can specify sizes and dimensions using relative units or absolute units

FOLDERS
- MyApp
  - app
  - bin
  - config
  - db
  - lib
  - log
  - public
  - test
  - tmp
  - vendor
  - .gitignore
  - config.ru
  - Gemfile
  - Gemfile.lock
  - Rakefile
  - README.rdoc

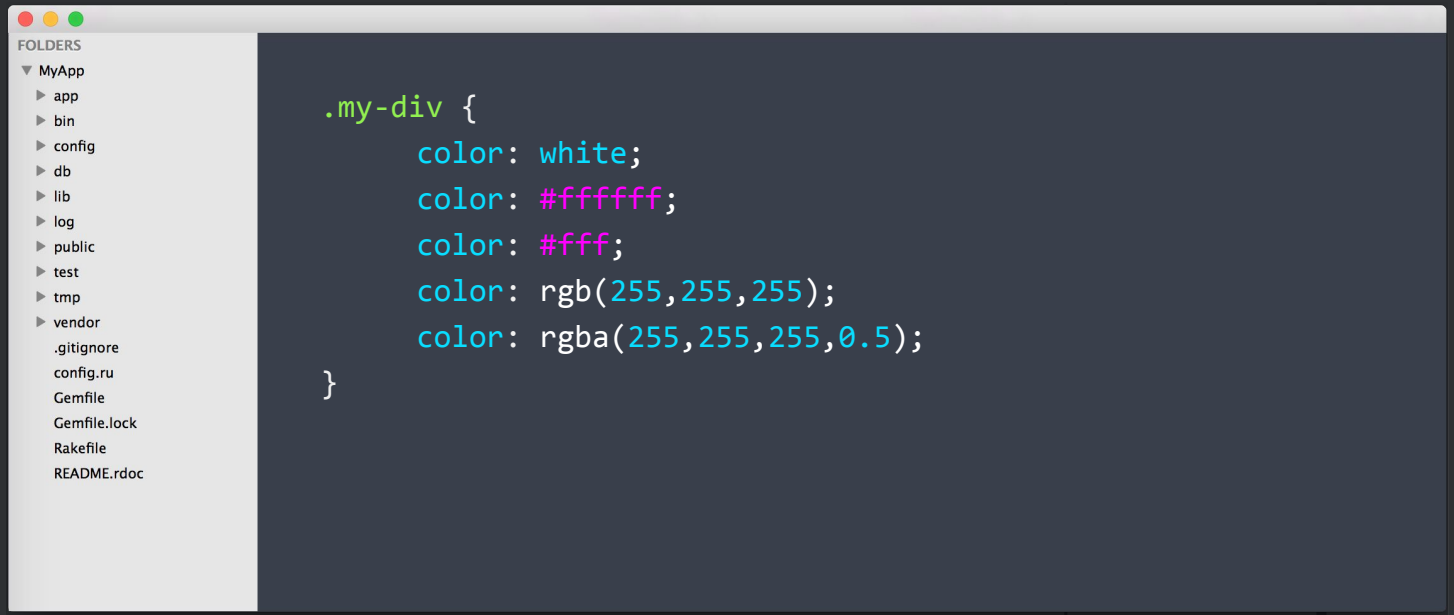
```
html { /*  font-size is 16px by default */
}

.my-div {
      width: 50%; /*  relative to its parent element */
      font-size: 20px; /*  absolute size */
      font-size: 200%; /*  relative to its parent element */
      font-size: 2em; /*  relative to its parent element */
      font-size: 2rem; /*  relative to the root element */
}
```

IRON HACK

@xharekx33

CSS Colors

You can specify colors by name, hex or rgb/a value.



IRON HACK

@xharekx33

Best practices

- Indent your css.
- Each selector in a separate line.
- Space before the opening brace. Closing brace in its own line.
- Space between property and value.
- End all declarations with a semicolon.
- If a value is 0, don't include the unit.
- Use dashes and not underscores in class names

IRON HACK

@xharekx33