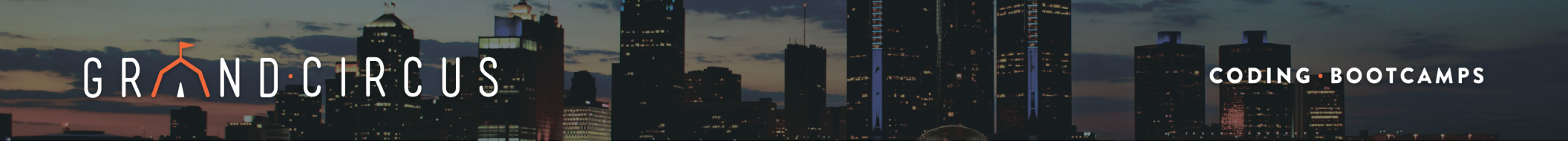


.NET C#
BOOTCAMP

FRONTEND TECHNOLOGIES

HTML

GOALS FOR TODAY

- The website as communication
- Your development toolchain
- Introduction to HTML

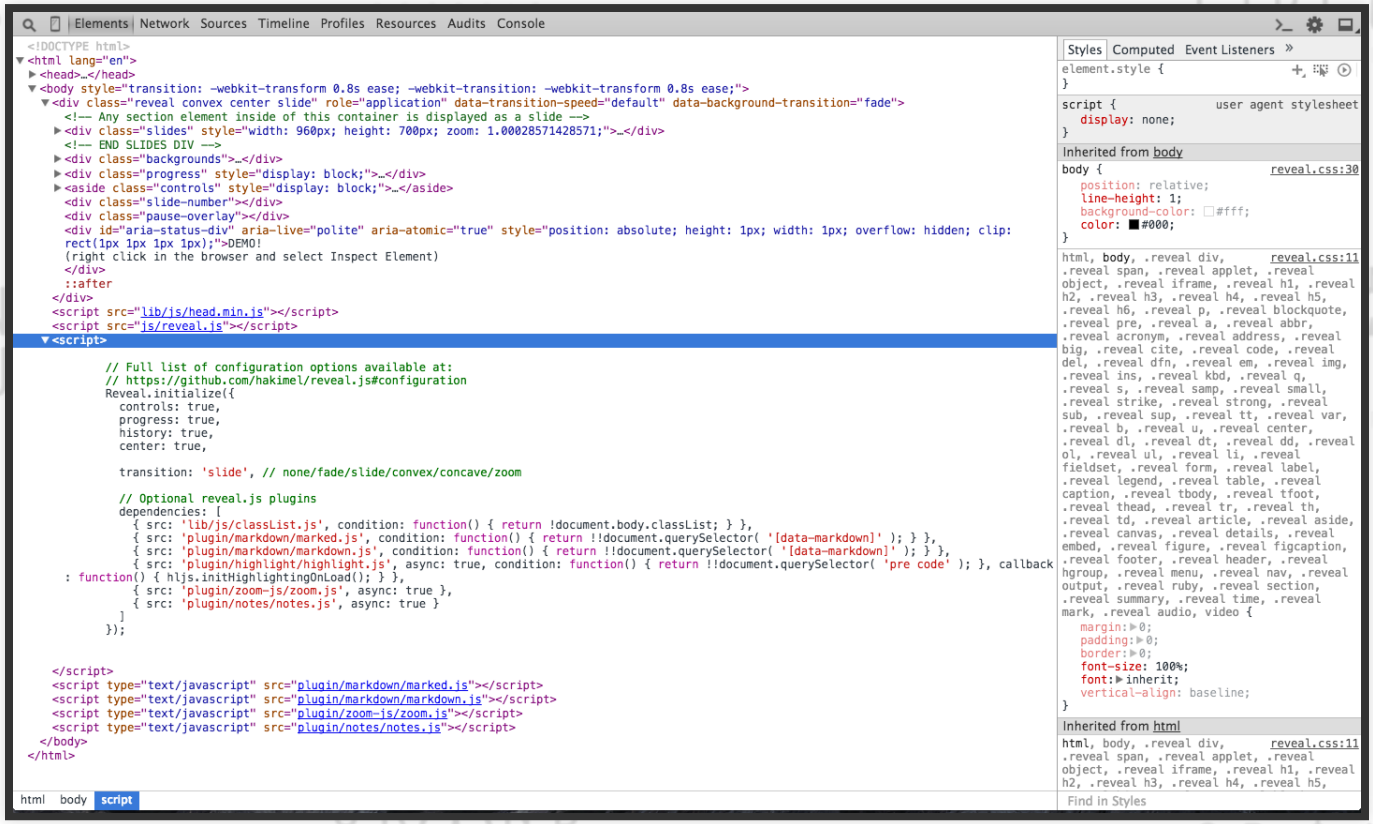
THE WEBSITE AS COMMUNICATION

Every website on the internet uses HTML & CSS.

...most of them use JavaScript as well, in one form or another.

DEMO!

(right click in the browser and select `Inspect Element`)


Elements Network Sources Timeline Profiles Resources Audits Console
<!DOCTYPE html>
<html lang="en">
<head>…</head>
<body style="transition: -webkit-transform 0.8s ease; -webkit-transition: -webkit-transform 0.8s ease;">

<!-- Any section element inside of this container is displayed as a slide -->
…
<!-- END SLIDES DIV -->
…
…
<aside class="controls" style="display: block;">…</aside>

DEMO!
(right click in the browser and select Inspect Element)

::after

<script src="lib/js/head.min.js"></script>
<script src="js/reveal.js"></script>
<script>
// Full list of configuration options available at:
// https://github.com/hakimel/reveal.js#configuration
Reveal.initialize({
controls: true,
progress: true,
history: true,
center: true,
transition: 'slide', // none/fade/slide/convex/concave/zoom
// Optional reveal.js plugins
dependencies: [
{ src: 'lib/js/classList.js', condition: function() { return !document.body.classList; } },
{ src: 'plugin/markdown/marked.js', condition: function() { return !!document.querySelector('[data-markdown]'); } },
{ src: 'plugin/markdown/markdown.js', condition: function() { return !!document.querySelector('[data-markdown]'); } },
{ src: 'plugin/highlight/highlight.js', async: true, condition: function() { return !!document.querySelector('pre code'); }, callback
: function() { hljs.initHighlightingOnLoad(); } },
{ src: 'plugin/zoom-js/zoom.js', async: true },
{ src: 'plugin/notes/notes.js', async: true }
]
});
</script>
<script type="text/javascript" src="plugin/markdown/marked.js"></script>
<script type="text/javascript" src="plugin/markdown/markdown.js"></script>
<script type="text/javascript" src="plugin/zoom-js/zoom.js"></script>
<script type="text/javascript" src="plugin/notes/notes.js"></script>
</body>
</html>
html body script
Styles Computed Event Listeners
element.style {
}
script { user agent stylesheet
display: none;
}
Inherited from body
body { reveal.css:30
position: relative;
line-height: 1;
background-color: #fff;
color: #000;
}
html, body, .reveal div, reveal.css:11
.reveal span, .reveal applet, .reveal
object, .reveal iframe, .reveal h1, .reveal
h2, .reveal h3, .reveal h4, .reveal h5,
.reveal h6, .reveal p, .reveal blockquote,
.reveal pre, .reveal a, .reveal abbr,
.reveal acronym, .reveal address, .reveal
big, .reveal cite, .reveal code, .reveal
del, .reveal dfn, .reveal em, .reveal img,
.reveal ins, .reveal kbd, .reveal q,
.reveal s, .reveal samp, .reveal small,
.reveal strike, .reveal strong, .reveal
sub, .reveal sup, .reveal tt, .reveal var,
.reveal b, .reveal u, .reveal center,
.reveal dl, .reveal dt, .reveal dd, .reveal
ol, .reveal ul, .reveal li, .reveal
fieldset, .reveal form, .reveal label,
.reveal legend, .reveal table, .reveal
caption, .reveal tbody, .reveal tfoot,
.reveal thead, .reveal tr, .reveal th,
.reveal td, .reveal article, .reveal aside,
.reveal canvas, .reveal details, .reveal
embed, .reveal figure, .reveal figcaption,
.reveal footer, .reveal header, .reveal
hgroup, .reveal menu, .reveal nav, .reveal
output, .reveal ruby, .reveal section,
.reveal summary, .reveal time, .reveal
mark, .reveal audio, video {
margin: 0;
padding: 0;
border: 0;
font-size: 100%;
font: inherit;
vertical-align: baseline;
}
Inherited from html
html, body, .reveal div, reveal.css:11
.reveal span, .reveal applet, .reveal
object, .reveal iframe, .reveal h1, .reveal
h2, .reveal h3, .reveal h4, .reveal h5,
Find in Styles

TERMINOLOGY

TERMINOLOGY

Some key terms or phrases that are used as a matter of course in the software industry. You may know some or all of these, or you may have heard the terms but be unclear about their actual meaning. They are common jargon among developers

WEB DEVELOPMENT

Web development is a broad term for the work involved in developing a web site for the Internet. In industry parlance, 'web development' usually refers to the more code-related tasks such as programming JavaScript, coding HTML and CSS. It can even extend to tasks related to the back end server infrastructure such as creating web services and handling business logic for a company or product.

WEB DESIGN

The process of planning & structuring a website; specifically, the visual aspects and assets for the site. Recently, this job description has also begun to include interaction design. That is, designing the user experience (UX), information architecture, and the flow of the application or site.

WEB SITE

A largely informational web page. While they may include dynamic elements and react to user inputs. The general purpose of a web site is to provide information about a person, business, product, or service.

WEB APPLICATION

A more recent term to indicate a web site whose sole purpose is not just informational, but rather functional. Web applications have become robust enough to do everything from our taxes, manage our personal calendars, or even do standard desktop publishing tasks.

YOUR DEVELOPMENT TOOLCHAIN

CHECKLIST

- Google Chrome
- Sublime Text / Atom.io

HTML 101

HTML

HYPERTEXT MARKUP LANGUAGE

- Developed by Tim Berners-Lee
- Developed in 1989 at CERN
- Originally developed as a way to share documents

HTML

STRUCTURE

```html
<html>
  <head>
    <title>My First Website!</title>
  </head>
  <body>

    <p>Look at the amazing content of my awesome website.</p>

    <img src="Cat-picture.jpg" alt="The internet is powered by cat pictures">

  </body>
</html>
```

CSS

PRESENTATION

```css
body {
  background-color: white;
  font-size: 18px;
  color: blue;
}

p {
  text-align: center;
  font-weight: bold;
}

img {
  border: 1px solid black;
  padding: 50px;
}
```

HTML

HYPERTEXT MARKUP LANGUAGE

HTML ELEMENTS

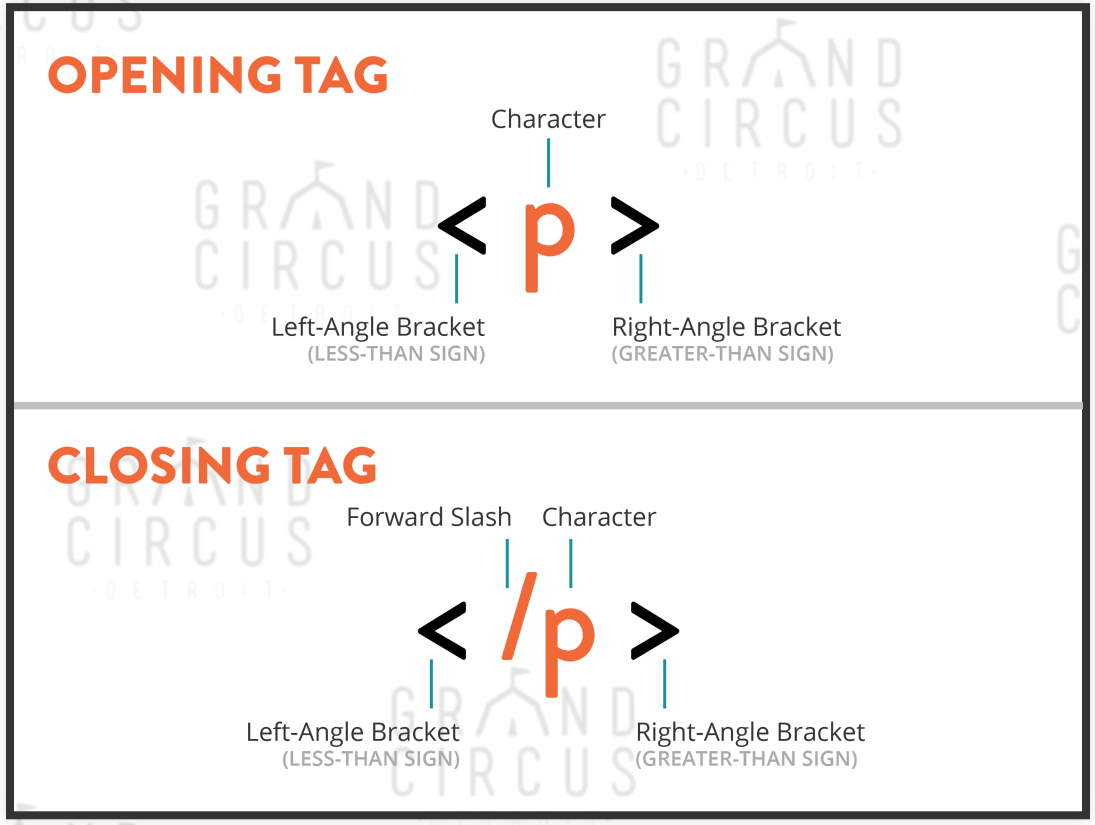
An element is an individual component of HTML

HTML TAGS

A tag marks the beginning and end of an element.

Tags are containers. They tell you something about the content between the opening & closing tags.

ANATOMY OF AN HTML ELEMENT



EXAMPLE

```
<p>This is a paragraph</p>
```

A paragraph elment consists of an opening `<p>` tag and a closing `</p>` tag and the content between the tags.

DIFFERENT KINDS OF ELEMENTS

Elements can be either container elements (they hold content) or stand-alone elements, sometimes called self-closing elements.

DIFFERENT KINDS OF ELEMENTS

Paragraph elements are containers

```
<p>Hi, I contain content</p>
```

Image elements are stand-alone

```
<img src="james.jpg" />
```

ATTRIBUTES

1. Provide additional information about HTML elements
2. Attribute tags include `class`, `id`, `style`, `language`, and `src` (source)
3. Attributes are positioned inside the opening tag, before the right bracket
4. Attributes are paired with values. Key / value pairs are an important concept in programming.
5. Selected from a pre-defined set of possible attributes depending on the element.

VALUES

1. Values are assigned to attributes
2. Values must be contained inside quotation marks

EXAMPLE ATTRIBUTES

```
<p id="trademark">An inline element<p>
```

EXAMPLE ATTRIBUTES

```
<div class="container">
  A bunch of stuff!
</div>
```

DOCTYPE

The first element on every HTML page. It tells the browser to expect HTML and what version to use.

HTML 5

```
<!doctype html>
```

HTML 4

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd" >
```

HTML TAG

After the doctype, all page content must be contained in the `<html>` tags

MAIN HTML TAGS

Tag	Description
`head`	Contains the page title and metadata
`body`	Contains all of the visible content
`title`	Optional tag. This is the name of your page. Nested in the `head` tag

NESTING

HTML elements 'nest' inside of one another. The element that opens first closes last.

NESTING EXAMPLE

```
<body>
  <div class="outer-div">
    <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod
    tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam</p>

    <a href="http://www.google.com">Google</a>

    <div class="inner-div">
      <ol>
        <li>Thing 1</li>
        <li>Thing 2</li>
        <li>Thing 3</li>
      </ol>
    </div>
  </div>
</body>
```

CONTENT TAGS

COMMON CONTENT TAGS

Tag	Description
`div`	defacto container element
`p`	used for body copy
`h1 thru h6`	designating titles/subtitles
`ol`	create a numbered list
`ul`	create an unordered list
`li`	list elements

SAMPLE HTML PAGE

```
<!DOCTYPE html>
<html>
<head>
  <title>My Totally Rad Website!</title>
</head>
<body>
  <h1>HELLO I AM JAMES AND THIS MY WEBSITE</h1>
  <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit,
    sed do eiusmod tempor incididunt ut labore et dolore magna
    aliqua. Ut enim ad minim veniam, quis nostrud exercitation
    ullamco laboris nisi ut aliquip ex ea commodo</p>

  <h4>My favorite things</h4>
  <ol>
    <li>Coding</li>
    <li>Board Games</li>
    <li>Quadcopters</li>
  </ol>
</body>
</html>
```

LINKS

ANCHOR ELEMENT

Links to other sites on the web (or within your project) are created using this element.

```
<a href="http://facebook.com">Facebook</a>
```

```
<a href="about.html">About Me</a>
```

LINK ELEMENT

Unlike the anchor element. The `<link>` specifies relationships between the current document and an external resource.

Most often this manifests as how CSS files are included with an HTML file.

```
<link src="main.css" rel='stylesheet' />
```

HTML COMMENTS

Like any other good coding language, HTML offers comments. They operat like comments in any other language. They are ignored by the browser engine.

```
<!-- Hello, I am a comment. -->
```

TABLES

TABLE ELEMENT

Tables are a way to represent complex information in a grid format. They are made of rows and columns.

Tables are compound elements (similar to lists) they are made up of several elements.

Element	Description
`<table>`	Table element
`<tr>`	Table row
`<td>`	Table cell
`<th>`	Table header cell (optional)

TABLE EXAMPLE

```
<table>
  <tr>
    <th>Business</th>
    <th>Numbers</th>
    <th>Synergy</th>
  </tr>
  <tr>
    <td>Cell 1</td>
    <td>Cell 2</td>
    <td>Cell 3</td>
  </tr>
  <tr>
    <td>Cell 4</td>
    <td>Cell 5</td>
    <td>Cell 6</td>
  </tr>
</table>
```

Business	Numbers	Synergy
Cell 1	Cell 2	Cell 3
Cell 4	Cell 5	Cell 6

STYLED TABLES

Id	Name	Email	Investments	Action
261	Alfred Alan	aalan@gmail.com	Stocks	
227	Alison Smart	asmart@biztalk.com	Residential Property	
246	Ally Emery	allye@easymail.com	Stocks	
212	Andrew Phips	andyp@mycorp.com	Stocks	
218	Andy Mitchel	andym@hotmail.com	Stocks	
221	Ann Melan	ann_melan@iinet.com	Residential Property	
243	Ben Bessel	benb@hotmail.com	Stocks	
232	Bensen Romanolf	benr@albert.net	Bonds	
233	Brad Cole	bradc@hotmail.com	Stocks	
241	Catherine Benchman	cathb@hotmail.com	Stocks	

74 items found, displaying 1 to 10.
[First/Prev] **1**, 2, 3, 4, 5, 6, 7, 8 [Next/Last]

Table caption

Header	Header	Header	Header	Header	Header
sample data	sample data	sample data	sample data	sample data	sample data
sample data	sample data	sample data	sample data	sample data	sample data
sample data	sample data	sample data	sample data	sample data	sample data
sample data	sample data	sample data	sample data	sample data	sample data
sample data	sample data	sample data	sample data	sample data	sample data
footer	footer	footer	footer	footer	footer

A NOTE ON TABLES

You may be thinking it now or you may think later that tables would be a great way to position content for site...

Don't do that. Just... don't. Trust me on this. It's not a good idea and it will give me an ulcer.

(srsly, don't)

WHAT'S WRONG WITH THIS CODE?

Look at the following examples and tell me what is wrong with the code.

```
<html>
    <head>
    <body>
    </head>

    </body>
</html>
```

```
<html>
  <head>
    <title>The Best Site Evar!!
  </head>
  <body>

    <p>Check out this riveting content!</p>

  </body>
</html>
```

```
<p style=hotStuff>Check out this riveting content!</p>
```

FOLDER STRUCTURE

This stuff is not exciting but it's *important*.

MAKING FOLDERS

Let's make sure you know how to make a folder in your operating system. This might seem like a silly thing to be sure we all know how to do but it's important because you'll making lots of folders in this class.

THE RULE OF THREES

In the beginning our projects will be very simple but as our projects grow in complexity it will be beneficial to organize them. My personal rule is that if i have 3 of the same kind of file (.html, .css, etc). I will make a folder for them.

RECAP

You should understand and be able to use:

- HTML elements
- Proper nesting
- HTML Comments
- Correct folder structure

CODE ALONG

LET'S MAKE OUR FIRST WEBSITE

1. Make a folder, name it `FirstWebsite`
2. Make a file, name it `index.html`
3. Follow Along

CSS

PART TWO: CSS

GOALS FOR TODAY

- CSS Styles
- Linking HTML and CSS
- Inline and Block Elements
- Character Codes

CSS 101

CASCADING STYLE SHEETS

(remember that 'cascading' part)

CSS IN THE WILD

```
.reveal.page .slides section {
             padding: 30px;
             min-height: 700px;
             -moz-box-sizing: border-box;
             box-sizing: border-box; }

             .reveal.page .slides section.past {
             z-index: 12; }

             .reveal.page .slides section:not(.stack):before {
             content: '';
             position: absolute;
             display: block;
             width: 100%;
             height: 100%;
             left: 0;
             top: 0;
             background: rgba(0, 0, 0, 0.1);
             -webkit-transform: translateZ(-20px);
             transform: translateZ(-20px); }
```

CSS RULES

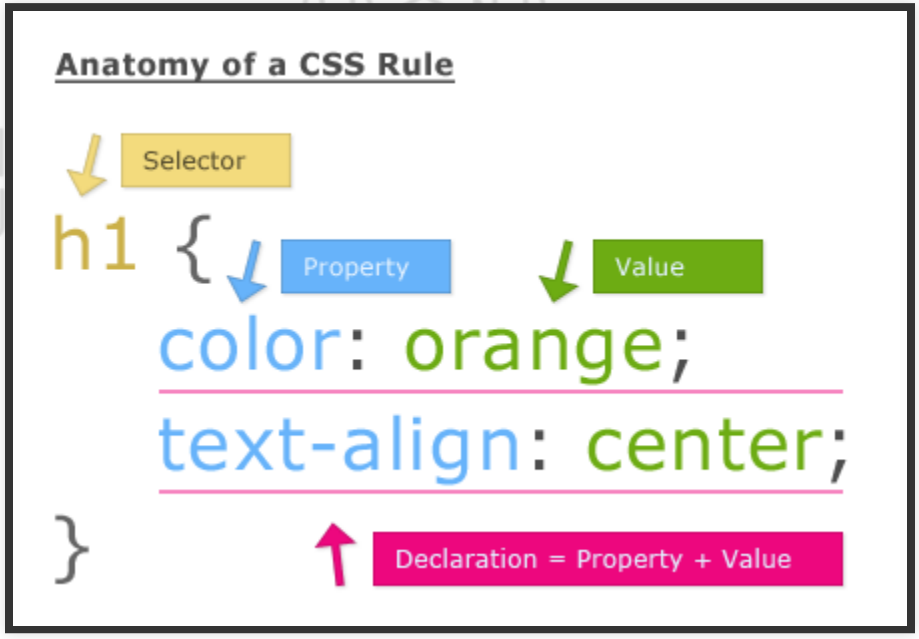
Individual components of CSS are called rules

CSS RULES

CSS rules are made up of two parts:

- One or more selectors
- One or more declarations
- The declaration must be inside curly braces that follows the selector

CSS SYNTAX


Anatomy of a CSS Rule
Selector
h1 {
Property
Value
color: orange;
text-align: center;
}
Declaration = Property + Value

CSS COMMENTS

Just like HTML, CSS offers comments:

```
/* This is a CSS comment,
           it can be multi-line */
```

CSS DECLARATIONS

- Declarations are made up of the property and value of the style you want to apply.
- They can be grouped together so that more than one declaration may be applied to a selected element.
- Declaration groups must be surrounded by curly brackets.
- Declarations must end in a semicolon.

```
.selector {
  background-color: red;
  color: white;
  border: 1px solid black;
  border-radius: 5px;
}
```

CSS SELECTOR

The selector instructs the browser to search the page for any HTML element that matches the given criteria. It applies any applicable declarations to that element.

CSS SELECTOR - ELEMENT

Elements can be selected by their element name. In this case, all elements of that element type will be selected and have the styles applied.

CSS SELECTOR - ELEMENT

```css
p {
  position: absolute;
  top: 0px;
  left: -100px;
}
```

CSS SELECTOR - CLASS

Elements can be selected based on HTML attributes such as class. In this case all elements that have a matching class attribute will be selected.

CSS SELECTOR - CLASS

```css
.timer {
  position: absolute;
  top: 0px;
  left: -100px;
}
```

CSS SELECTOR - ID

Elements can also be selected based on HTML attribute ID. In this case only one element would be selected, as HTML IDs are intended to be unique.

CSS SELECTOR - ID

```
#fluffy {
  position: absolute;
  top: 0px;
  left: -100px;
}
```

CSS SELECTOR - DESCENDENT SELECTORS

Selectors can be combined to become more specific. This example selects searches for any paragraph tag that is nested inside a div tag.

```
div p {
  position: absolute;
  top: 0px;
  left: -100px;
}
```

CSS SELECTOR - MULTIPLE

In addition a set of declarations can be applied to more than one selector by listing a number of comma-separated selectors.

```
.timer, img, div p, #kitty {
  position: absolute;
  top: 0px;
  left: -100px;
}
```

POP QUIZ HOT SHOT

Look at the following examples and tell me which Elements (if any) would be returned by the following selectors.

POP QUIZ

```
p {
  /* blah blah */
}
```

POP QUIZ

```
#fuzzy {
  /* blah blah */
}
```

POP QUIZ

```
.blue {
  /* blah blah */
}
```

POP QUIZ

```
.blue, #fuzzy {
  /* blah blah */
}
```

POP QUIZ

```
div p {
  /* blah blah */
}
```

QUESTIONS?

CSS PROPERTIES

CSS PROPERTIES

There are literally hundreds of css properties that are available for use. We don't have time to go over more than just a few. We will go over a few of the most common. However, the best strategy is to google for styling options as you're working.

COMMON CSS PROPERTIES

Property	Description
`background-color`	background color for an element
`color`	color of the *text* in an element
`font-family`	typeface for text
`font-size`	size for text (px, %, em, pt)
`font-weight`	used to bold text (if possible)
`text-decoration`	used for underline (mostly)
`height`	specifies the height of an element
`width`	specifies the width of an element

COLOR IN CSS

Method	Syntax	Description
color name	`white`	a list of 140 predefined colors
hexidecimal	`#FF0000`	RGB values in hex 00 - FF (0 - 255)
RGB	`rgb(255, 0, 187)`	RGB values in decimal numbers (0 - 255)
RGBA	`rgba(255, 0, 187, 0.5)`	RGB values with an added alpha (opacity) value

CSS UNITS

Method	Syntax	Description
em	`1em`	Scalable unit based on font size
pixels	`16px`	Fixed number of pixels
percent	`120%`	Percent value based on font size

FOLDER STRUCTURE

- Still not exciting
- Still important

PROJECT SETUP

Wherever you are going to keep your project work:

- Create a new folder called CSS-Exercise
- Inside that folder create two files called `index.html` & `main.css`

SET UP

LINKING HTML AND CSS FILES

Add this element to your HTML page's head

```
<link href="main.css" rel="stylesheet">
```

CODE ALONG

STYLING AN HTML PAGE

SET UP

1. Make a new folder
2. Create new files `index.html` and `styles.css`
3. Set up a basic HTML page (doctype, html, head, body)
4. `link` the css file to the html page
5. Test the linkage
6. Add some HMTL content (tags, classes, and ids)

SIMPLE CSS STYLES

Include at least one of each:

- class selector
- id selector
- child selector
- multiple selectors

SIMPLE CSS STYLES

Apply the following CSS properties:

- background-color
- color
- font-family
- font-size
- one or more additional properties of your choice

CSS POSITIONING

BLOCK VS. INLINE ELEMENTS

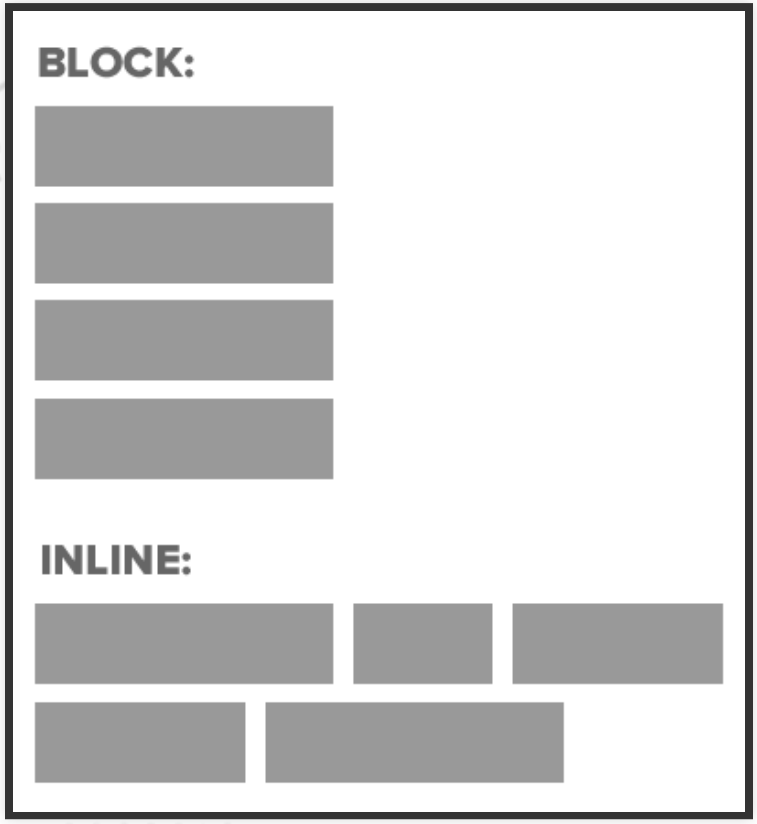
BLOCK ELEMENTS

Block elements each appear on a new line of a web page, like paragraphs. Spatially, what is happening is that the block element takes up all of the horizontal space it can. It stretches to fill all the space to the left and right of the element within its parent container.

INLINE ELEMENTS

Inline elements are rendered without starting a new line. They appear side by side until reaching the edge of its parent container. Then it will start a new line.

BLOCK VS. INLINE ELEMENTS



BLOCK VS. INLINE ELEMENTS

- *Inline element*: `<a> <img>`
- *Block element*: `<div> <p> <ul> <li> <table>`
  ...and just about everything else.

RIDDLE ME THIS

How could you make a list of anchor tags (inline element), display on a page as a verticle list?