44-563: Unit 01

Developing Web Applications and Services

Welcome - First Week

- Introduction & overview
- Important information / first day
- Weekly plan - how to be successful!
- Initial content:
  - MON: Static pages & HTML5
  - WED: CSS
  - FRI: Lab
  - **No Class next Monday**

Introduction & Overview

Instructors & Assistants

- Dr. Nathan Eloe
- Dr. Tianyang Wang
- Dr. Michael Rogers
- Dr. Denise Case

- Naga Sravanthi Modali
- Bhavishya Arelli
- Kranthi Kumar

Lots of Acronyms

HTML

XML

REST

SOAP

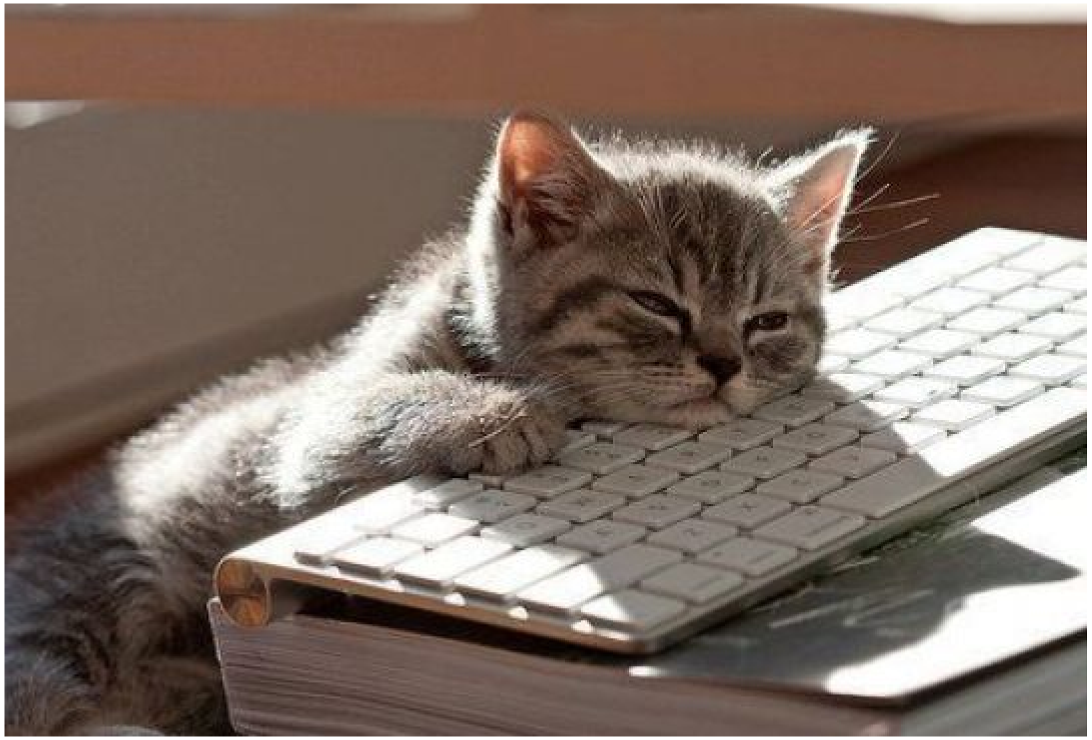
JSON

JS

CSS

REST

HTTP



Hang in there - during our 16 weeks, soon you'll find you will be speaking the language of web development!

Course Overview

- Web dev is constantly evolving... **initiative** & **willingness to explore**
- **Hands-on, problem-solving** approach
- Pair programming & **teamwork**
- **Collaborative** & **creative** assignments
- 4 weeks on **client-side** development (Exam 1)
- Remaining on **server-side** development (Exam 2)
- Team **project &** comprehensive **final exam**
- Lots of opportunities for points! Come, participate, and explore - our goal is to make it a fun, creative, useful course for all backgrounds.

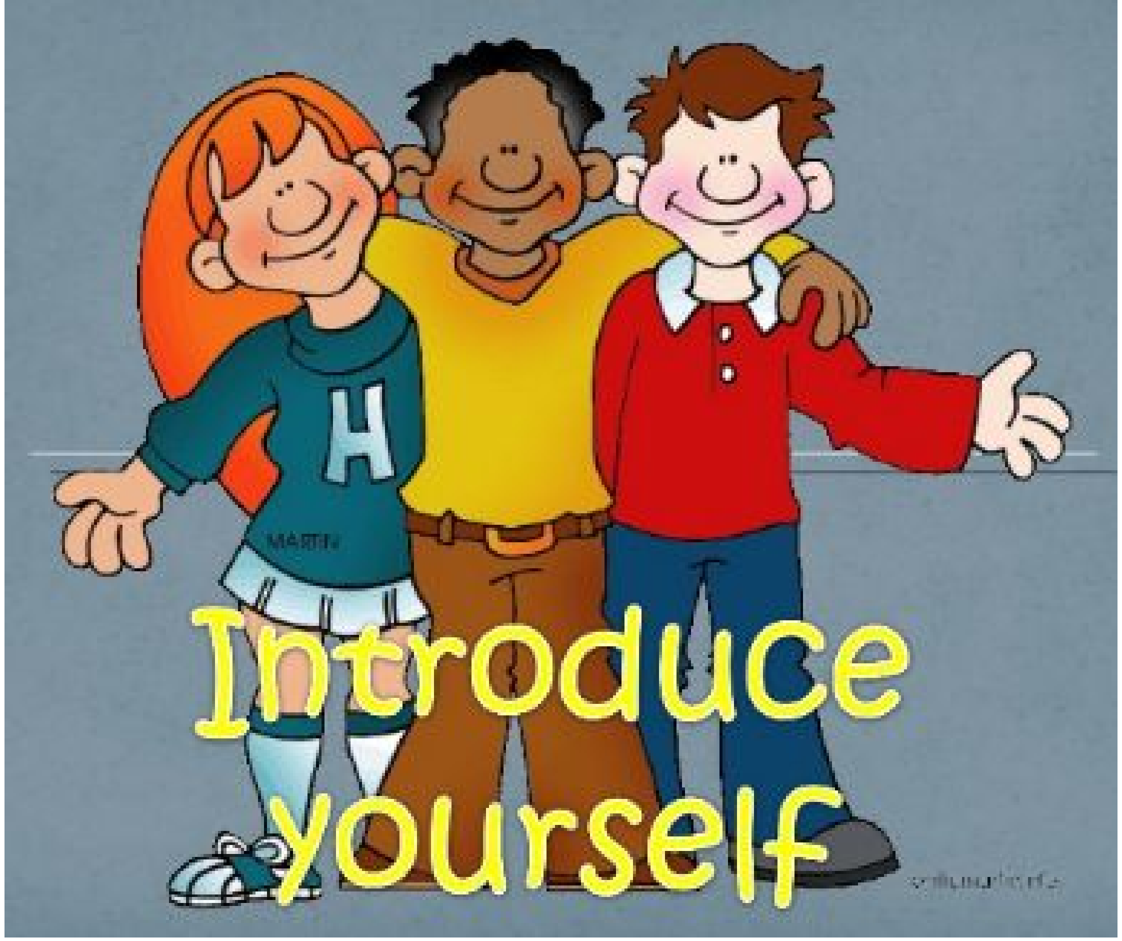
Use this course to develop your brand

Develop your **network**, your **brand**, your "**T**" (broad knowledge in many areas, with a few subjects where you go deep).

If you don't type, practice at www.typing.com! If you like design instead of development, explore more in art and color.

Our goals are to **understand** & **discuss** concepts & choices; and **implement some basic** functional requirements

Create, **share**, & **publish** your work, develop/extend your portfolio (try Cloud9, GitHub pages, & more)!



https://image.slidesharecdn.com/introduce-yourself-110518050006-phpapp02/95/introduce-yourself-1-728.jpg?cb=1305694929

Introductions: Share your name, undergraduate or graduate student, and what area(s) do you plan to focus on? (project management, design, development, data, security...)

Important Information

Syllabus is our guide.

Professionalism is one of our key goals.

⭑**Email: Always begin subject line with course number & section (e.g. 44-563-03) & helpful subject**

⭑**Academic Integrity** and your personal brand is crucial.

⭑**Always** cite your sources (we practice this a lot).

Any questions? Then, we'll continue to the course site.

Important information

Open our course site:

1. Syllabus
2. Academic Integrity Policy.
3. MS-ACS Attendance policy
4. General Information and Student Agreement.
5. **Due First Day of Class: Tonight**, after reading 1-4, take the Student Agreement Quiz (must pass with 100% before any points can be earned)

Important information

Open our course site:

6. In class: take **self-assessment pre-survey** (get your unique code-word - it improves the validity of our results, but doesn't compromise confidentiality) Takes ~10 min - then we'll continue.

Please take time to complete the survey now.

We will allow about 10 minutes.

http://idolosol.com/images/10-minute-timer-1.jpg

Important Information

Surveys completed?

Collaboration & all external resources:

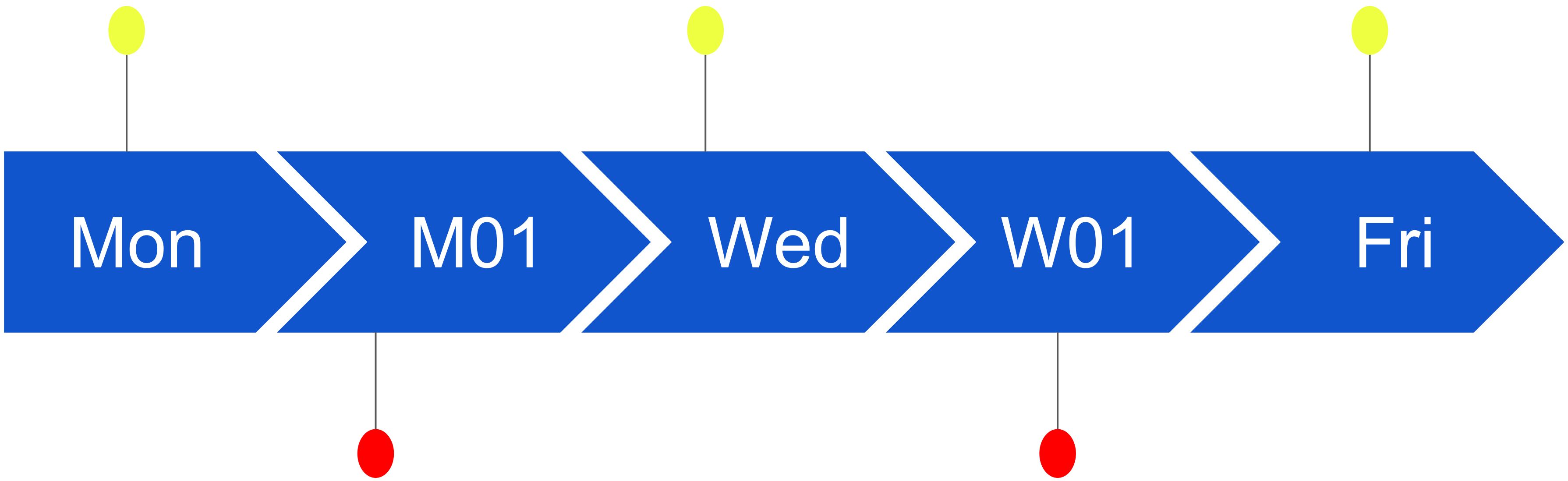
- **Always credit your sources. All sources. Every sentence, every image, every class, every time.**
- Do not compromise assessments.
- Do not plagiarize.
- Do not represent a submission as your own if it is not. You must be able to reproduce any submission at any time.

Weekly Plan

Start on in-class exercises

Expand content / in-class exercise

Short quiz, 10-minute workshop, lab time for assignments



Submit exercise or installation discussion (due Wed morning)

TODAY is the first

Submit exercise

(due Friday morning)

In some cases, the submission will be a **screenshot** posted to a discussion

Monday: Easy Points

Easy points (tasks are required for upcoming assignments).

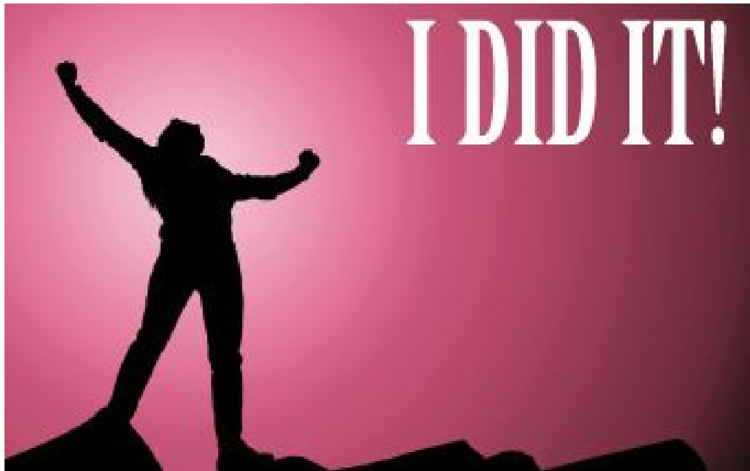
Often involve installations. **Embed screenshots.** Show running on **your desktop.** Use Canvas files to host your screenshots.

Good way to bring up assessment scores - these are the **curving** requested at course end. Don't miss them!

Submit via **course site Monday evening** (so you have time to seek help if problems arise: the actual due date may be later and assignment-specific)**.**

Typically take only 10-15 minutes.

Post problems and ask for assistance **BEFORE** the deadline.



Wednesday: Exercises

Easy points for completing in-class activities (generally) - often work towards the more interesting coding assignments.

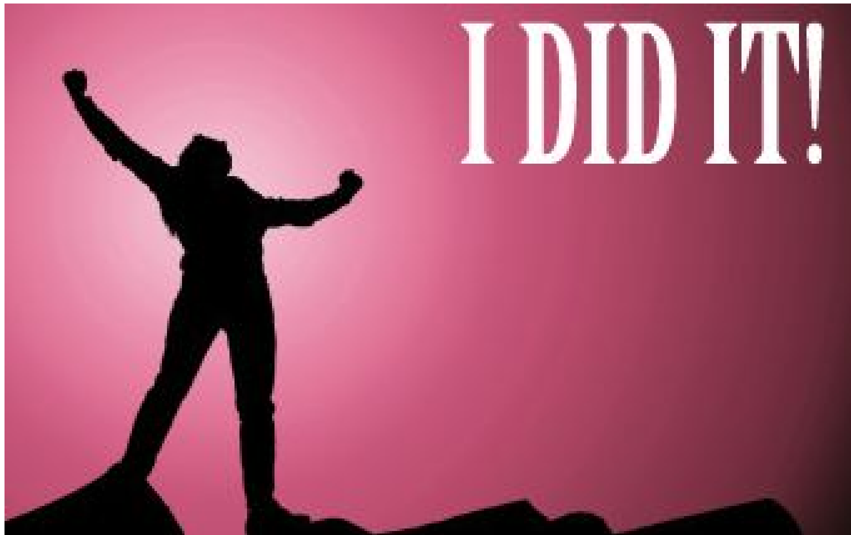
Usually involve a small **unique**, **creative** implementation.

Your own work, on your own system (embed screenshot on your desktop).

Good way to bring up assessment scores. Don't miss these!

Submit to **course site Wednesday evenings** (ideally: the same previous slide)

Post problems and ask for assistance
BEFORE the deadline.



Friday: Quiz, Workshops, Labs

Fridays are **fun days** - free from lectures!

5-minute **reinforcement quiz**.

You get 3 tries - **don't miss these**!

1. On your own - practice for an exam.
2. You can discuss with a friend.
3. All together for third try (with instructor).

Later: 10-minute topic **workshops**.

Lab time to work on more challenging **coding assignments (e.g. A01)**

You will always have *at least one* coding partner (even if submit separately).

May involve checking code into GitHub or BitBucket repositories.

The more you try, the more you learn. Have fun!

Due Tonight!

- ❑ Read Syllabus
- ❑ Read Academic Integrity Policy
- ❑ Review Attendance Policy

- ❑ Submit Student Agreement Quiz with 100% tonight!

Always bring

1) your **laptop** and

2) a bound **notebook** for taking notes, turning in quizzes, etc.

Any other questions about the course?

Now for the good stuff

What **three** main technologies are **foundational** to web development?

Foundational Technologies For Web Development

HTML

CSS

JavaScript

HTML (structure)

Simple Static Web Page

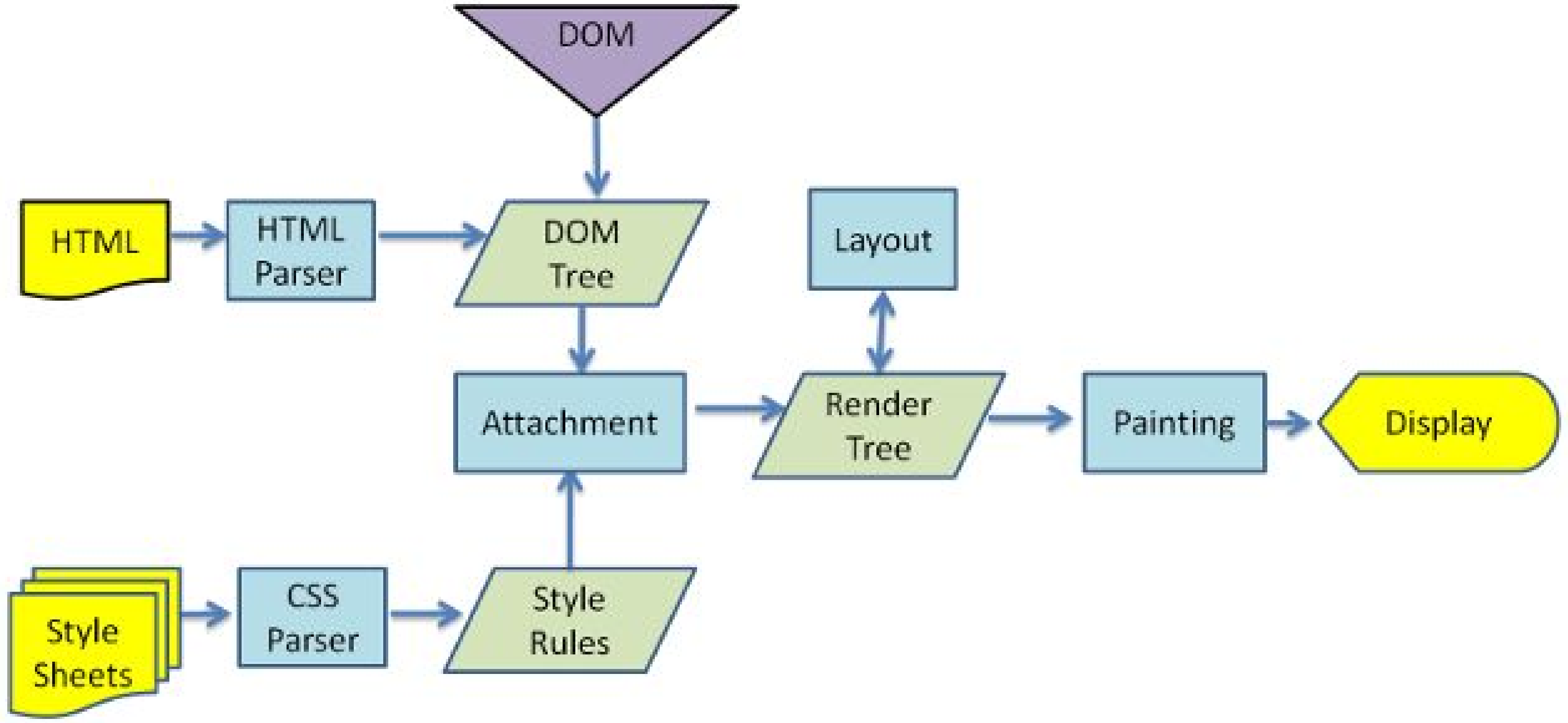
index.html

```
<!DOCTYPE html>
<html>
  <head>
     <title>Name</title>
  </head>
  <body>
     <h3>About Me</h3>
      Try adding more headings
      And see what happens
      with h1 or h2 instead.
  </body>
</html>
```

Guess: What appears in the browser?

What's Happening?

1. The **browser** program reads the **first line** and knows the file is **HTML**.
2. It **parses** the text into HTML **tags**, **text**, and **style** information.
3. It creates a **DOM** (document object model) **tree**, then a **render tree**.
4. It then **traverses these trees** and **paints** & **displays** result to user.



http://www.html5rocks.com/en/tutorials/internals/howbrowserswork/

What is HTML?

What is HTML?

HyperText Markup Language (HTML) is the standard markup language used to create web pages.

With CSS and JavaScript, HTML is a cornerstone technology used to create web pages and GUIs for mobile and web applications. Browsers read HTML files and **render** them into visible or audible web pages.

HTML describes the **structure** of a website.

https://en.wikipedia.org/wiki/HTML

Programming vs Markup

What other languages have you heard about?

1. Are they **programming** languages (e.g. Java, C#, C++, C, python, etc)
2. Or **markup** languages that describe or annotate information (e.g. HTML, CSS, Markdown, etc)

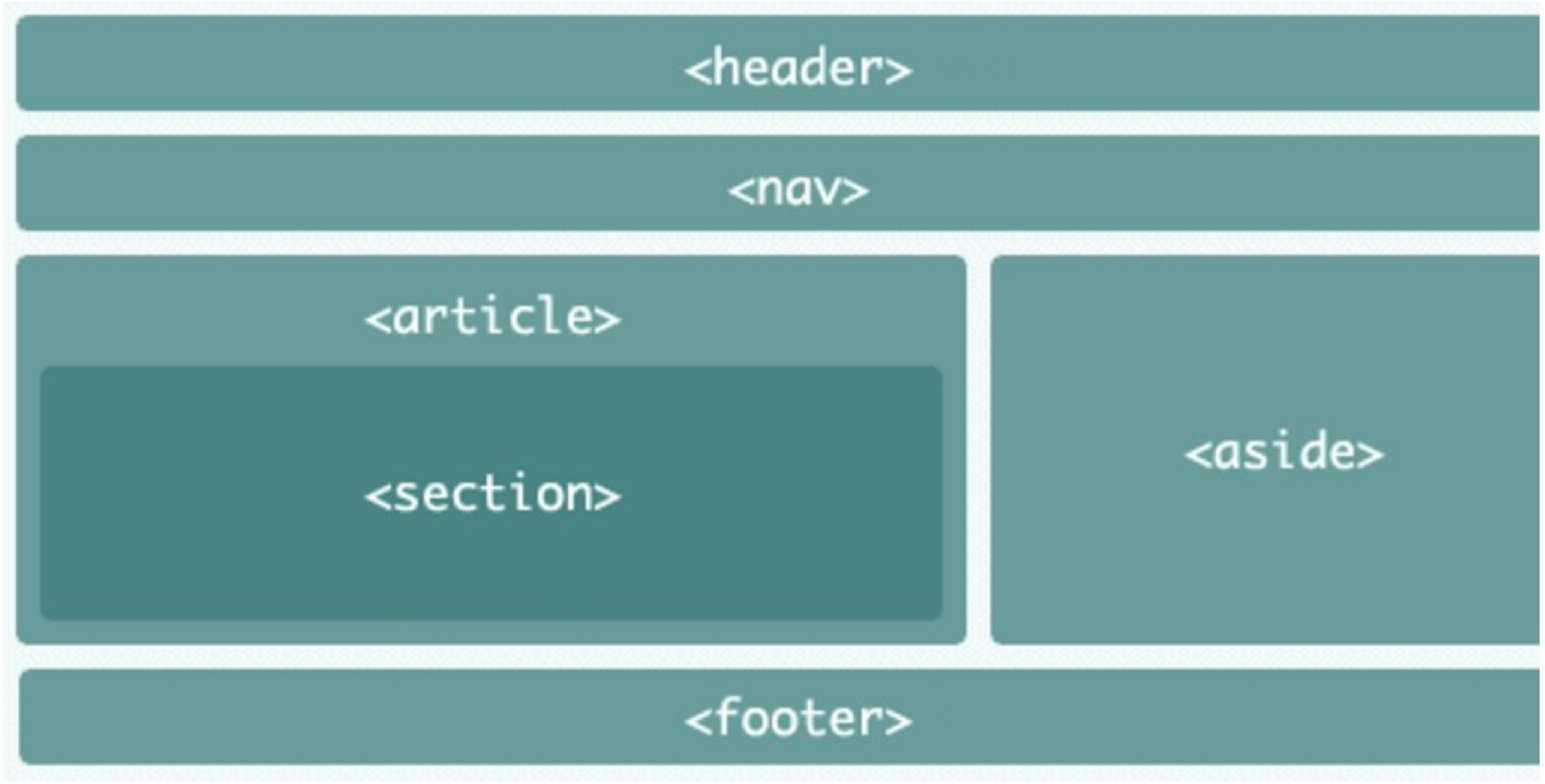
Hands on: Make a web page

1. Start / Notepad or editor.
2. Copy (CTRL-C) & paste (CTRL-V) the following into a file (or get from here).
3. Save as **index.html** in your Documents folder.
4. In File Explorer, right-click index.html & open with Chrome.

index.html

```
<!DOCTYPE html>

<html>
<head>
	<title>Your Name Here</title>
</head>
<body>
<h1>My First Heading</h1>
<p>My first paragraph.</p>
</body>
</html>
```

Semantic HTML tags



Use these tags often!

http://www.htmlgoodies.com/tutorials/html5/new-tags-in-html5.html#fbid=0pqVY796LM-

Other Famous HTML Tags

Page Structure: <doctype>,<html>, <head>,<meta>,<title>,<link>, <script>,<body>

Headings: <h1> … <h6>

Paragraphs, etc.: <p>, <br>, <hr>

Formatting: <strong>,<em>,<i>,<sub>,<sup>

Lists: <ol>, <ul>, <li>

Links: <a href="">

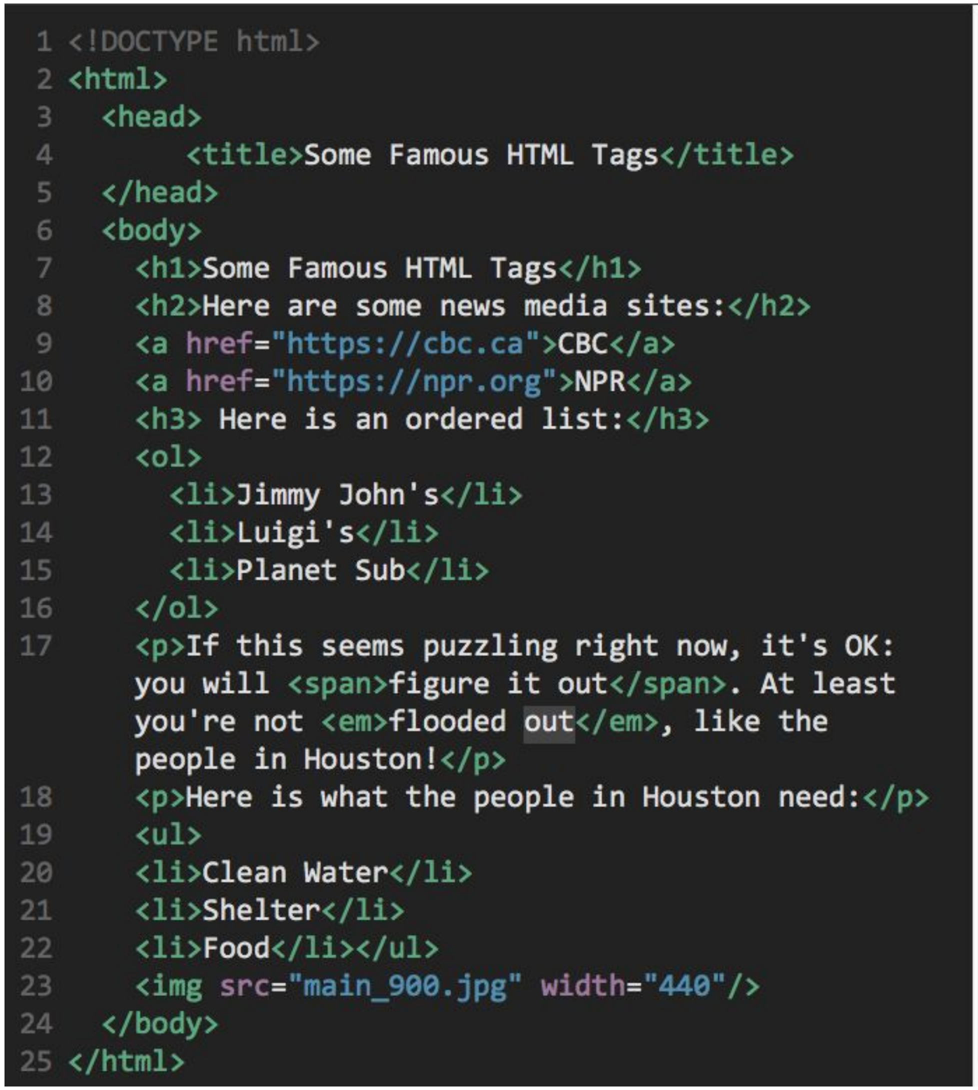
Images: <img src="">

Tables: <table><th>,<tr>,<td> (table header, table row, table data cell, respectively)

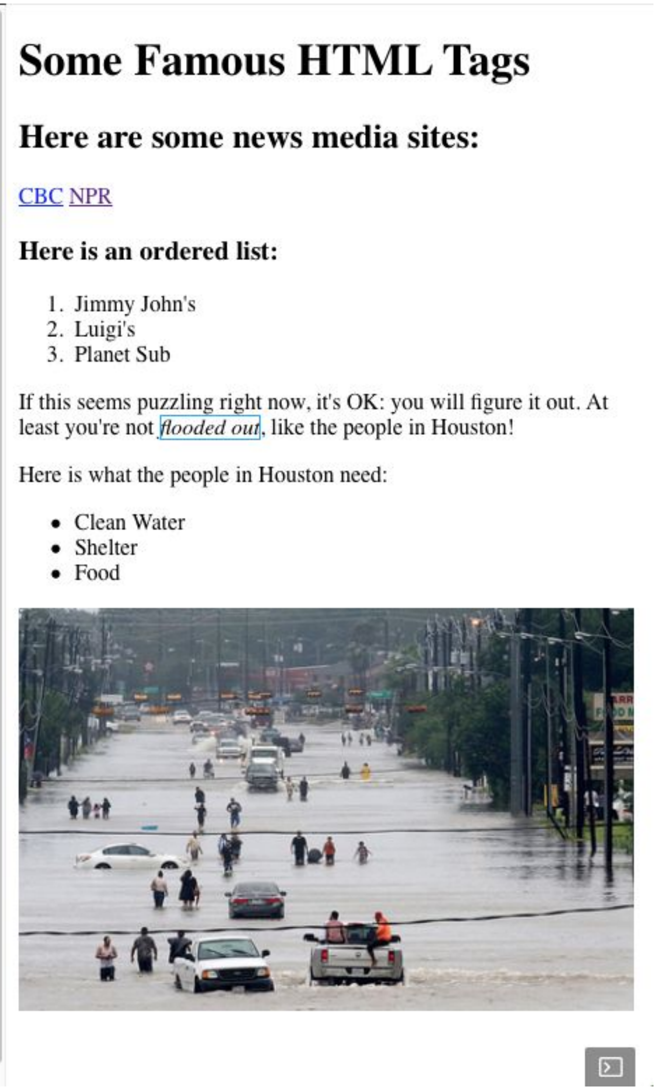
Comments: <!-- Who needs comments, anyways? -->

Almost all tags have an **ending** tag, e.g., <h1> First **</h1>**.

Famous HTML Tags in Use



```
<!DOCTYPE html>
<html>
  <head>
       <title>Some Famous HTML Tags</title>
  </head>
  <body>
    <h1>Some Famous HTML Tags</h1>
    <h2>Here are some news media sites:</h2>
    <a href="https://cbc.ca">CBC</a>
    <a href="https://npr.org">NPR</a>
    <h3> Here is an ordered list:</h3>
    <ol>
      <li>Jimmy John's</li>
      <li>Luigi's</li>
      <li>Planet Sub</li>
    </ol>
    <p>If this seems puzzling right now, it's OK:
    you will <span>figure it out</span>. At least
    you're not <em>flooded out</em>, like the
    people in Houston!</p>
    <p>Here is what the people in Houston need:</p>
    <ul>
    <li>Clean Water</li>
    <li>Shelter</li>
    <li>Food</li></ul>
    <img src="main_900.jpg" width="440"/>
  </body>
</html>
```



Some Famous HTML Tags

Here are some news media sites:

CBC NPR

Here is an ordered list:

1. Jimmy John's
2. Luigi's
3. Planet Sub

If this seems puzzling right now, it's OK: you will figure it out. At least you're not *flooded out*, like the people in Houston!

Here is what the people in Houston need:

- Clean Water
- Shelter
- Food

Things to know: HTML

- Is a **Markup** language (*not* a programming language)
- HTML documents contain **tags**.
- HTML **elements** are the components generated from parsing the tags.
- Elements can have **attributes** - and other content like text or child elements.
- Elements are arranged as a **tree** (with a root node of HTML)
- Elements typically have a **start tag** and an **end tag** with content in between, except a few like <br>

More About HTML Tags

- **Block-level** elements start on a **new line** and take all the available width.
  - Examples: **<div>, <h1>-<h6>, <p>, <form>**
- **Inline** elements do not start on a new line and take only the width required.
  - Examples: **<span>, <a>, <img>**
- **Container** elements: **<div> and <span>** don't really do anything by themselves: they are designed as containers for css*

* Cascading style sheets - used to add style to content. Coming up later this week.

OPTION 1

HTML in Action

1. Find a programming partner. Work together on one notebook.
2. Open a web browser.
3. Google *Zhengrui Qin*
4. Right-click and view page source.
5. What common html elements and tags can you identify?

Link: http://www.cs.wm.edu/~zhengrui/

OPTION 2

HTML in Action

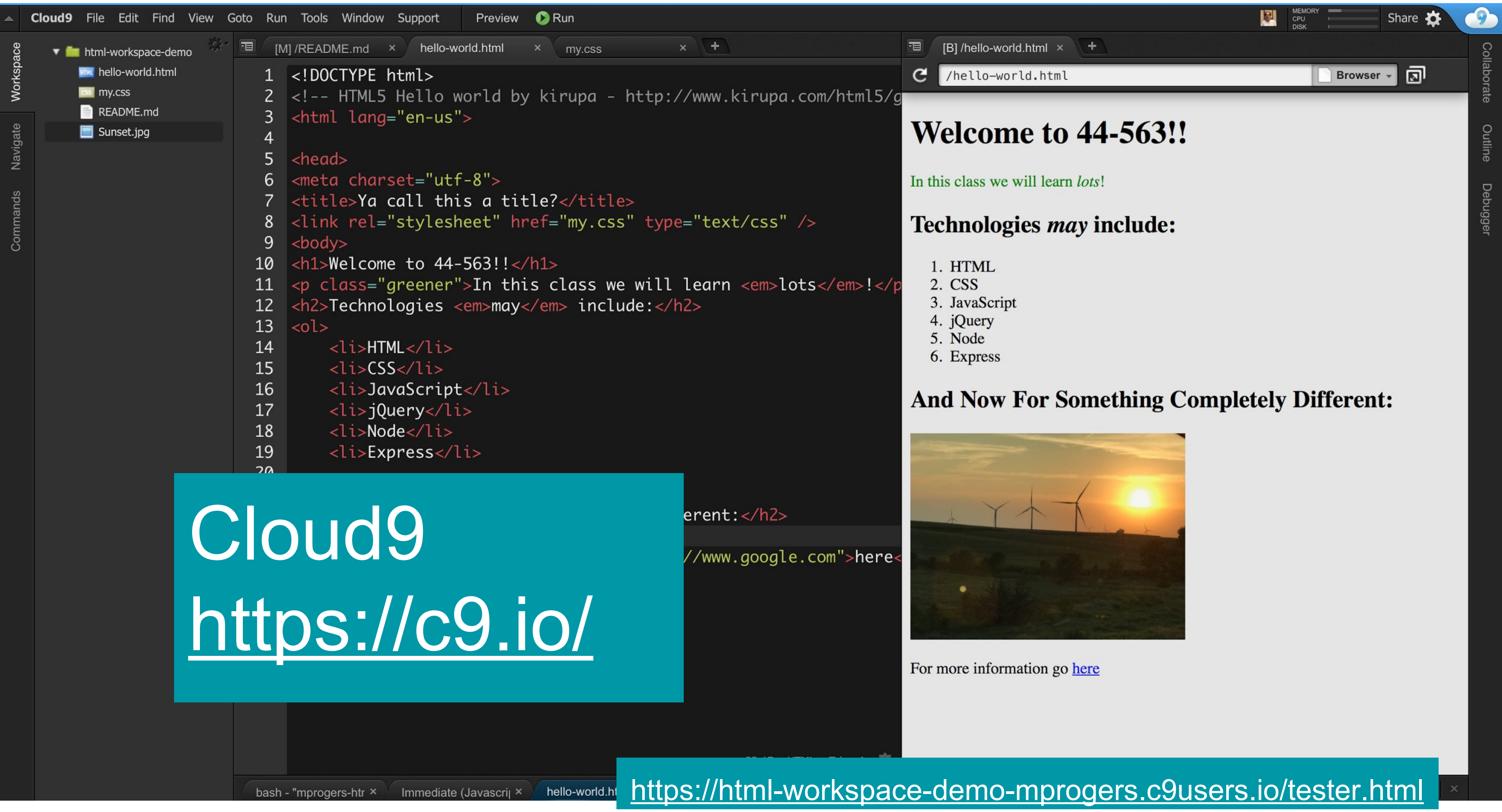
1. Open this link in your browser.
2. Right-click and view page source (CTRL-U)
3. What common html elements and tags can you identify?
4. Select All (CTRL-A)
5. Copy (CTRL-C)
6. Paste into a new text file.

Link:

https://www.w3schools.com/html/tryit.asp?filename=tryhtml_layout_float

OPTION 3

HTML in Action (and a shameless plug for c9.io)



```
<!DOCTYPE html>
<!-- HTML5 Hello world by kirupa - http://www.kirupa.com/html5/g
<html lang="en-us">

<head>
<meta charset="utf-8">
<title>Ya call this a title?</title>
<link rel="stylesheet" href="my.css" type="text/css" />
<body>
<h1>Welcome to 44-563!!</h1>
<p class="greener">In this class we will learn <em>lots</em>!</p
<h2>Technologies <em>may</em> include:</h2>
<ol>
    <li>HTML</li>
    <li>CSS</li>
    <li>JavaScript</li>
    <li>jQuery</li>
    <li>Node</li>
    <li>Express</li>
```

```
erent:</h2>
```

```
//www.google.com">here<
```

Welcome to 44-563!!

In this class we will learn *lots*!

Technologies *may* include:

1. HTML
2. CSS
3. JavaScript
4. jQuery
5. Node
6. Express

And Now For Something Completely Different:

For more information go here

Cloud9
https://c9.io/

https://html-workspace-demo-mprogers.c9users.io/tester.html

HTML is powerful

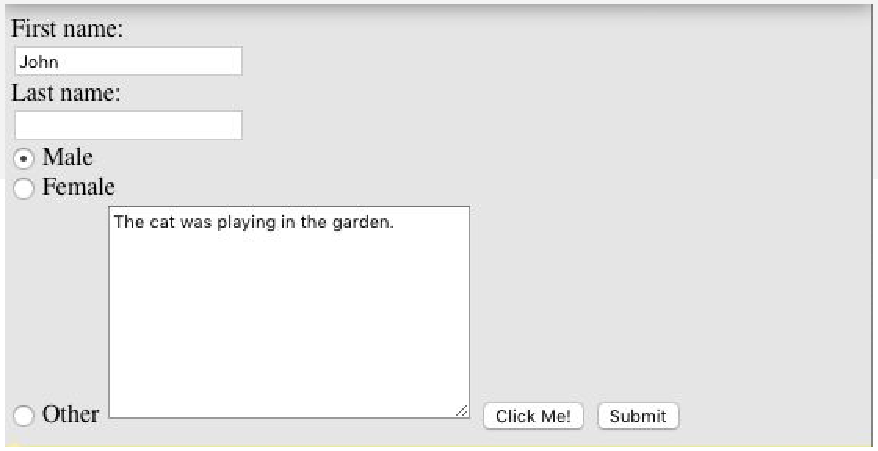
- You can get input from users (button clicks, swipes, sound streams) and send it back for processing.
- The built-in <form> tag makes it easy.
- We'll just introduce it here - we won't use forms until later so don't worry too much about these yet.

HTML **forms** for user input

Input types include:

- text
- radio
- url
- email
- phone
- color
- date
- datetime
- datetime-local
- month
- search
- tel
- time
- week

```
<form action="">
     First name:<br>
     <input type="text" name="firstname" value="John" required>
     <br>
     Last name:<br>
     <input type="text" name="lastname" required>
     <br>
     <input type="radio" name="gender" value="male" checked> Male<br>
     <input type="radio" name="gender" value="female"> Female<br>
     <input type="radio" name="gender" value="other"> Other
     <textarea name="message" rows="10" cols="30">
     The cat was playing in the garden.
     </textarea>
     <button type="button" onclick="alert('Hello World!')">Click Me!</button>
     <input type="submit" value="Submit">
</form>
```



First name:

John

Last name:

● Male

○ Female

The cat was playing in the garden.

○ Other

Helpful Tip: Some of these are HTML5 types, and if not supported in an earlier browser, will be interpreted as type=”text”.

~~Unhelpful Tip: Spiders hate it when you say attercop to them.~~

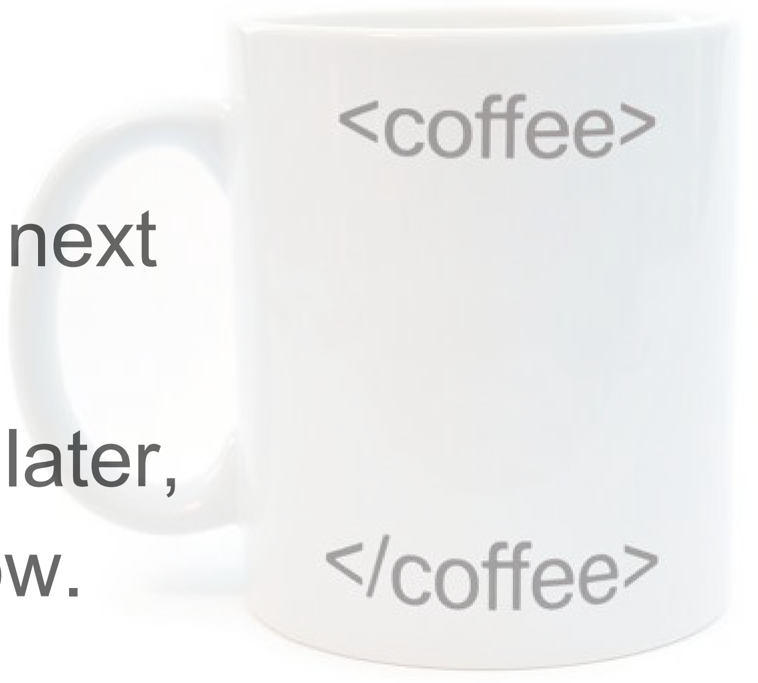
Dropdown Lists

```html
<form method="post">

        <select name="Ice cream Flavours">

                <option value="double chocolate">Double Chocolate</option>

                <option value="vanilla">Vanilla</option>

                <option value="strawberry">Strawberry</option>

                <option value="caramel">Caramel</option>

        </select>

</form>
```

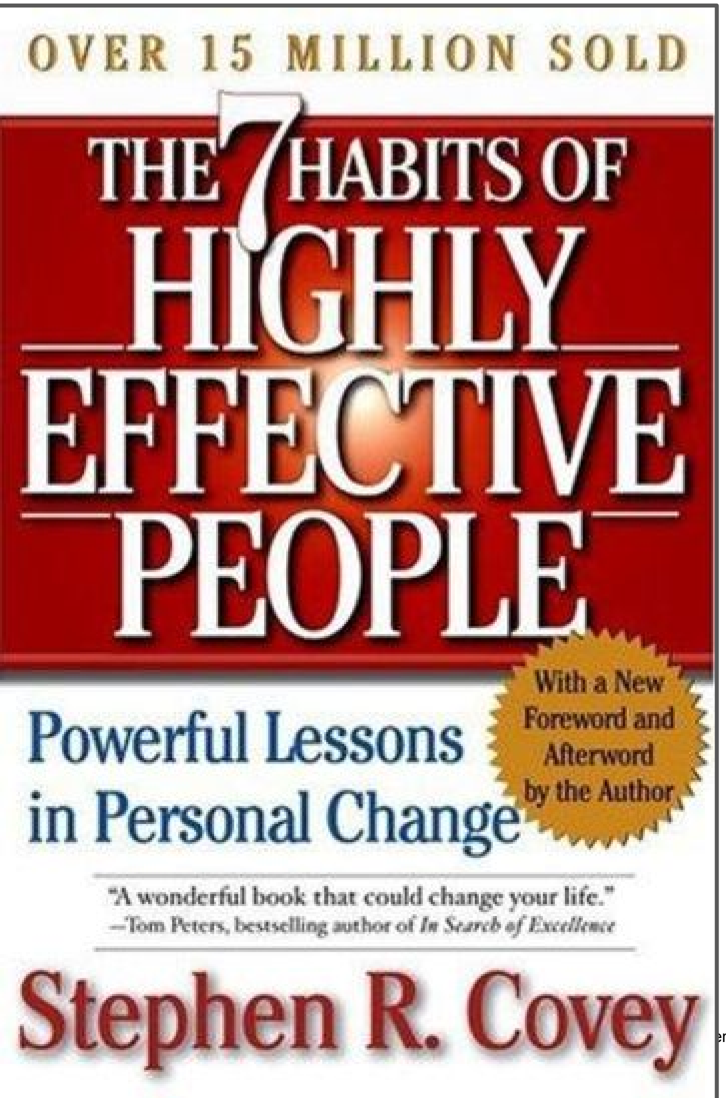
http://www.html5-tutorials.org/forms/dropdown-lists/

What do we need to know?

- There are **many** HTML tags.
- See the **Exam 01 Review Guide**, which lists the HTML tags (and attributes) that you should memorize for this course.
- We'll explore many over the next few weeks.
- Forms will be covered more later, but practice all other tags now.



Highly recommended


OVER 15 MILLION SOLD
THE 7 HABITS OF HIGHLY EFFECTIVE PEOPLE
Powerful Lessons in Personal Change
With a New Foreword and Afterword by the Author
"A wonderful book that could change your life." —Tom Peters, bestselling author of In Search of Excellence
Stephen R. Covey

Habit 2

Begin with the End in Mind

n.wikipedia.org/wiki/The_7_Habits_of_Highly_Effective_People#/media/File:The_7_Habits_of_Highly_Effective_People.jpg

M01

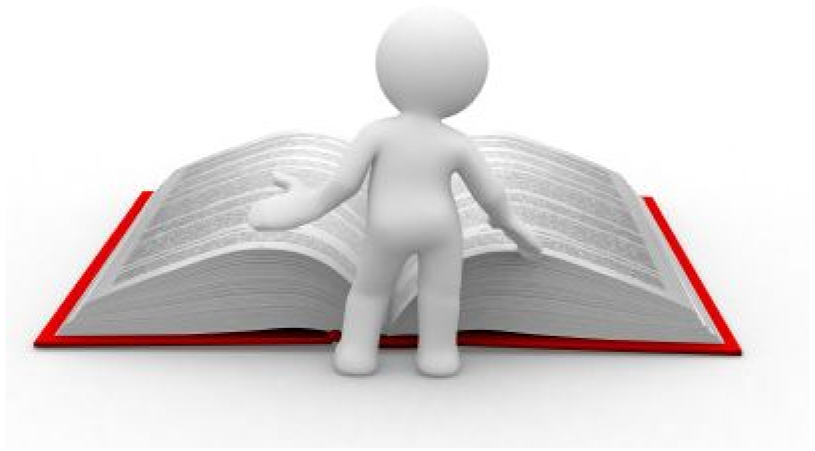
Find a programming partner.
Introduce yourself.

Work together on one notebook or discuss. Email files as desired.

- Build your personal web page from scratch.
- Add at least all HTML 1 tags from the **Exam 1 review guide**. (Habit 2). Be sure to nest them without overlapping.
- What can you do if you're not sure how to use a tag?
- We'll build on this Wednesday and you'll use it all semester.
- Your final version will get published with our project.

https://media.giphy.com/media/11JTxkrmq4bGE0/giphy.gif

Before CSS



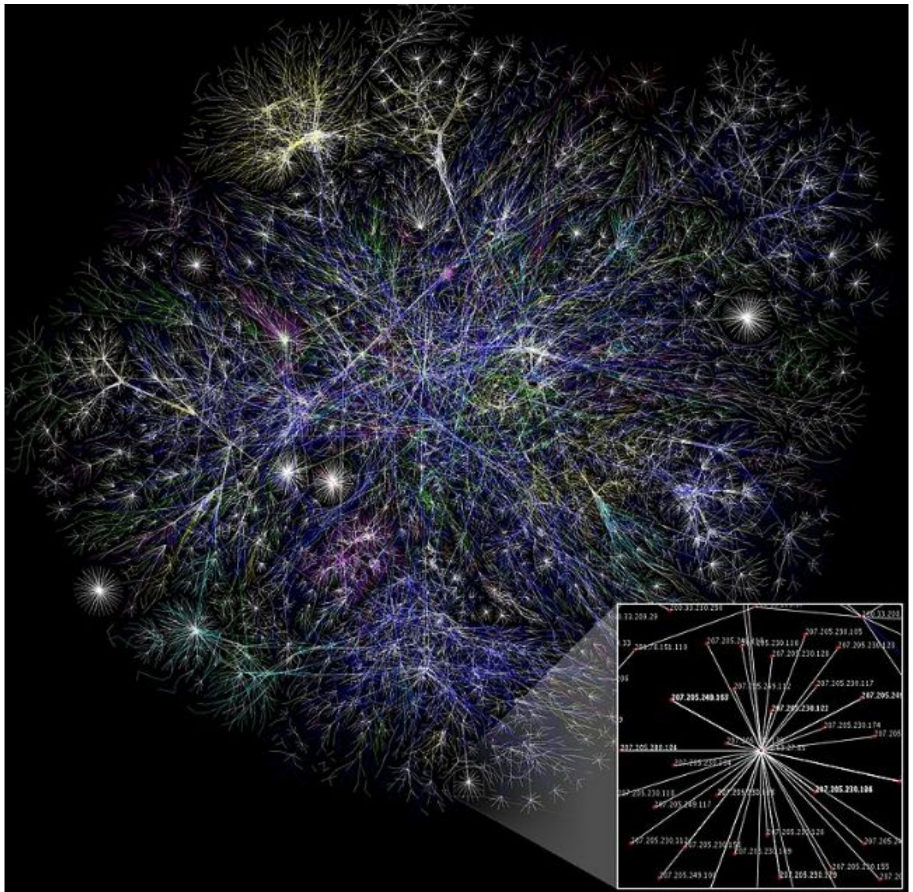
A few more terms & definitions

Web Speak

What is the **Internet**?

What is the **Internet**?

The **Internet** is the global system of **interconnected** computer **networks** that use the Internet protocol suite (TCP/IP) to link billions of devices worldwide.



What is the **WWW**?

What is the **WWW**?

The **World Wide Web** (**WWW**) is an **information space** where **documents and other web resources** are identified by URLs, interlinked by hypertext links, and can be accessed via the Internet.[1]

What is a web page?

A web page (or webpage) is a **document** suitable for the web and web browsers.

The web page is what **displays**, but the term also refers to a **computer file**, usually written in HTML or comparable markup language.

What is a web browser?

A **web browser** (or just **browser**) is a software **application** for retrieving, presenting, and traversing **information resources** on the web.

What are examples of browsers?

What is a Web App?

What is a Web App?

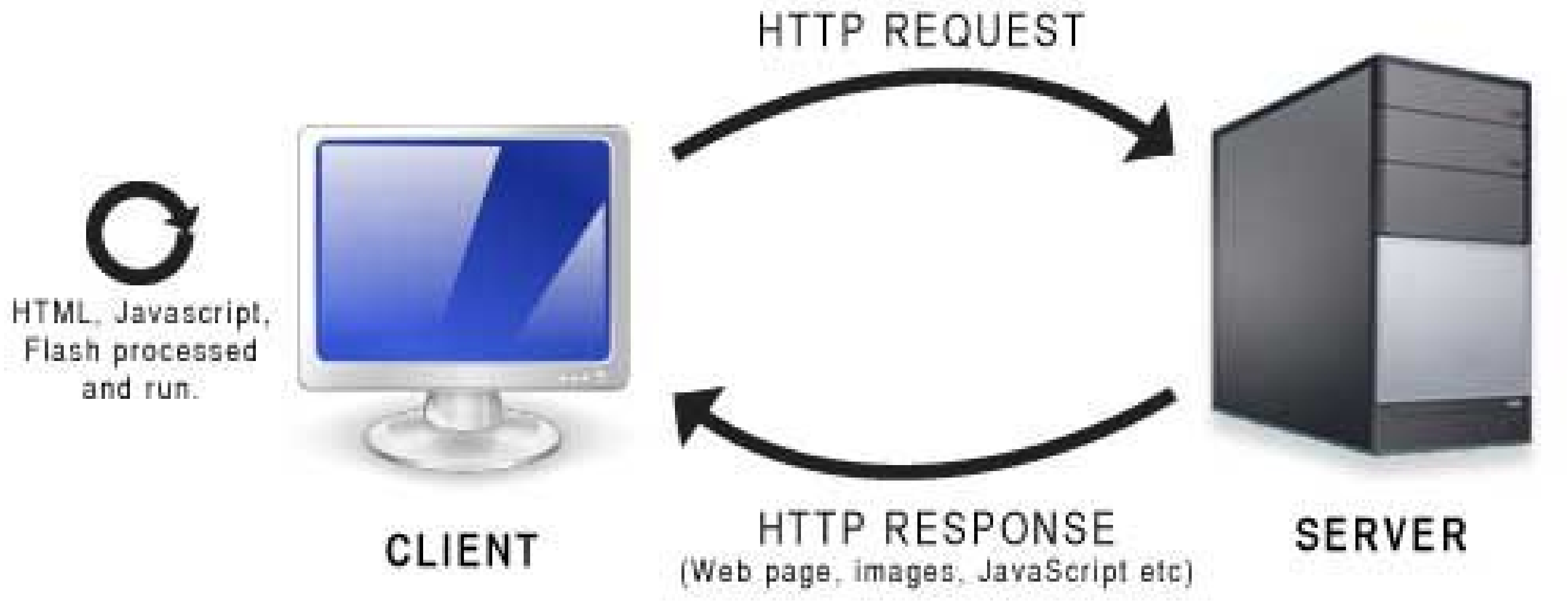
A web application (web app) is a ~~**client–server**~~ **software application** in which the client (or user interface) runs in a web browser.
What's the advantage? The user needn't install an app on their device. Developers can be assured that their users are running up-to-date software.

Source: https://en.wikipedia.org/wiki/Web_application

Web Apps: They're *Everywhere*

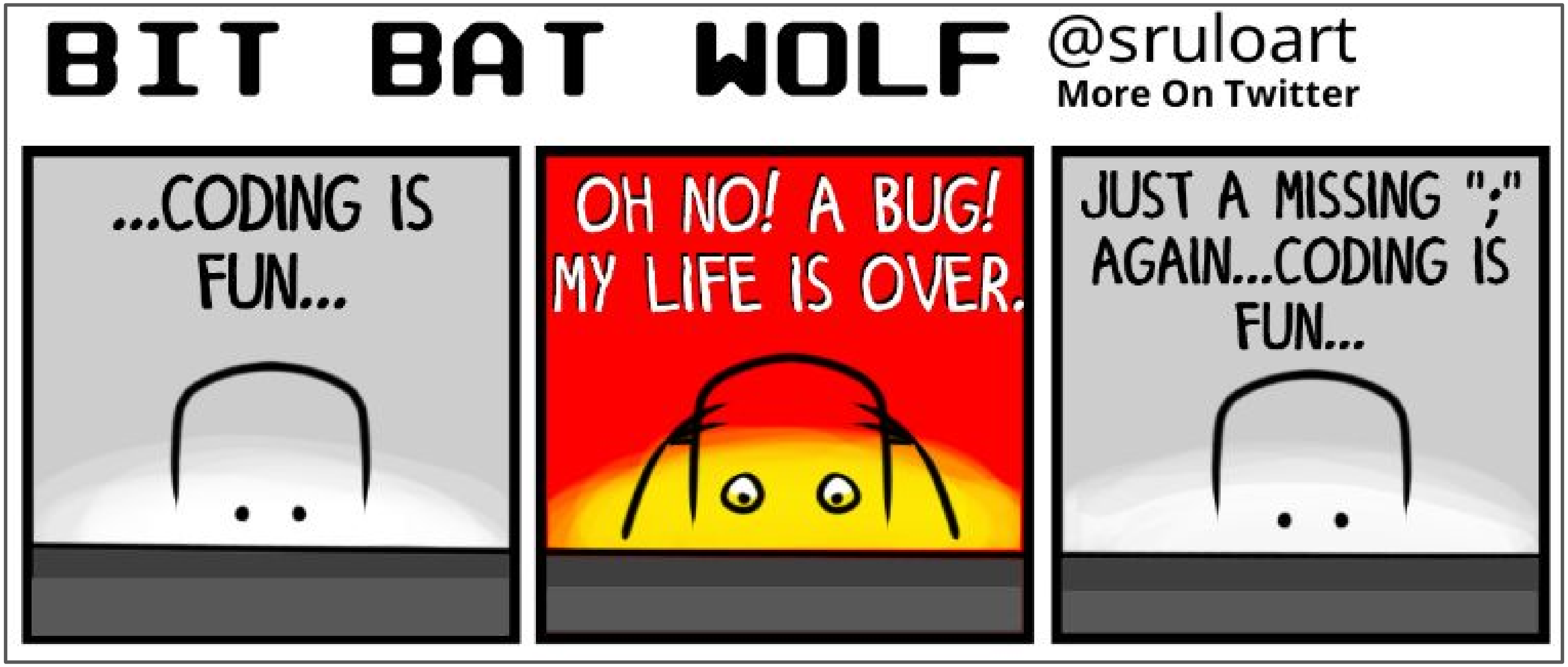
- when2meet.com
- Cloud 9 (c9.io)
- repl.it
- Microsoft Office Live
- Google Docs, Slides, Etc.
- Thimble (from Mozilla)
- codingbat.com
- … and a gazillion more!

Communication in a Web App


HTTP REQUEST
HTML, Javascript, Flash processed and run.
CLIENT
HTTP RESPONSE
(Web page, images, JavaScript etc)
SERVER

Where is the **front-end**?
Where is the **back-end**?
What is the key **client-side software**?

Back to the fun stuff - coding :)


BIT BAT WOLF
@sruloart
More On Twitter
...CODING IS FUN...
OH NO! A BUG! MY LIFE IS OVER.
JUST A MISSING ";" AGAIN...CODING IS FUN...

Note: JavaScript historically used semicolons in practice - but they aren't required. In this class, we now recommend (coding like the cool kids) and dropping them.

HTML + CSS

HTML is for **content** and **structure**

Academic Integrity Note: All non-original content should include a link to the original source. How important is citing all external content? **Every sentence, every image, every class, every time.**

Bonus: If HTML5 doesn't give you all the elements you need, you can make your own with Angular.

http://i.istockimg.com/file_thumbview_approve/12722030/3/stock-illustration-12722030-cartoony-framed-house.jpg

CSS is for **style**

https://www.loginradius.com/engineering/wp-content/uploads/sites/2/2015/07/CSS3-Responsive-Icons1.png

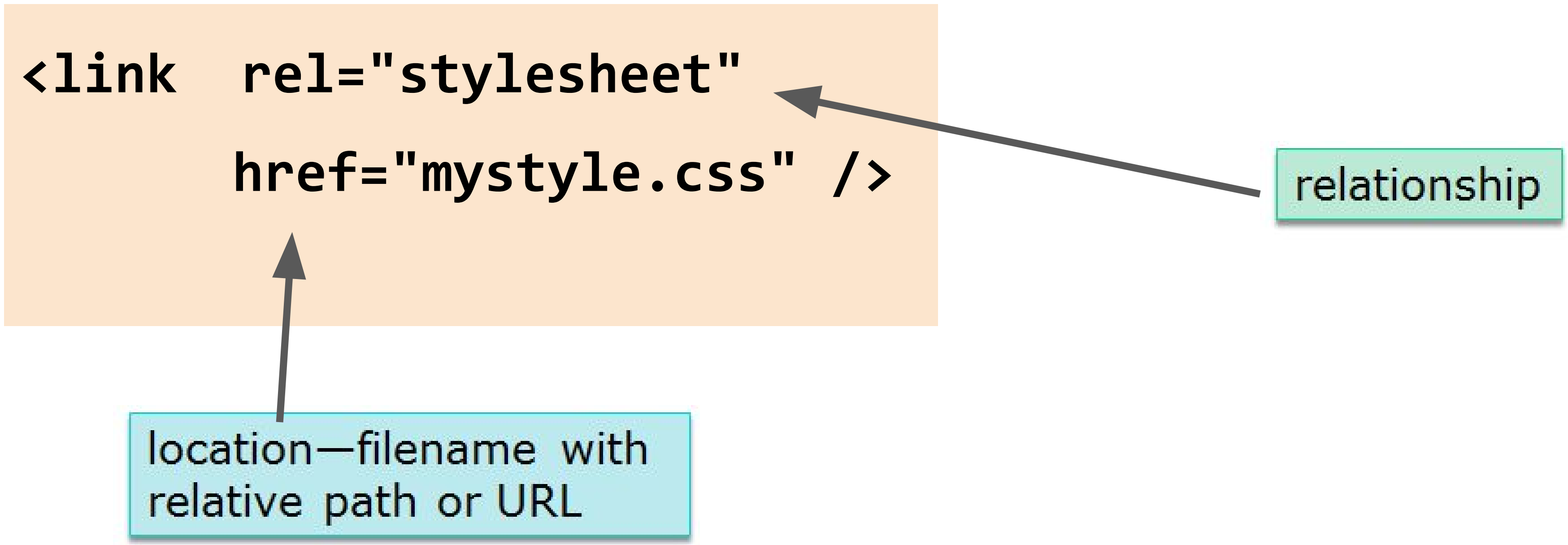
CSS

Cascading Style Sheets specify the style of elements in HTML documents.

One style sheet can be used for **multiple pages** to enforce consistent style.

Link Stylesheet to Page

To associate a stylesheet **mystyle.css** with an HTML file, add a line to the <head> section of the HTML file.



```
<link  rel="stylesheet"
       href="mystyle.css" />
```

Why link to another file?

HTML specifies the content of a web page.

Information regarding style (how to display) could be put in the same file, but in this course we teach the standard convention that style information **should always be stored in a separate file.** **Always** separate content from style keep files focused on their one job. It makes code much easier to maintain.

"Separation of concerns" is an important principle in software development.

CSS
(style)

Example CSS

```
* {
    padding: 0;
    margin: 0;
}

body {
        background: #fff;
        font: .74em "Trebuchet MS" Verdana, Arial, sans-serif;
        line-height: 1.5em;
}

a {
        color: #3B6EBF;
        text-decoration: none;
}
a:hover {
        text-decoration: underline;
}
```

Can you decipher this CSS?
*What does * typically mean?*
What does the second entry apply to?
What about the 3rd and 4th?
Px is pixels, but what is 1.5 em?

http://www.oswd.org/design/preview/id/3681/

CSS Syntax is Easy

```
CSS_Selector {
    property1 : value1;
    property2 : value2;
}
```

```
a {
	color: #3B6EBF;
	text-decoration: none;
}
a:hover {
	text-decoration: underline;
}
```

DIY Selection (Do It Yourself)

- The selectors on the previous slide applied the same style to **all elements of that tag type** -- e.g., all <a> tags tags would have that style applied to them.
- What if you want to be more **selective** with your selectors?
- What if you want to apply a style to a specific set of elements?
- What if you want to apply a style to just one of the <a> tags?
- In that case, use either a **class (.)** or **id (#)** selector.

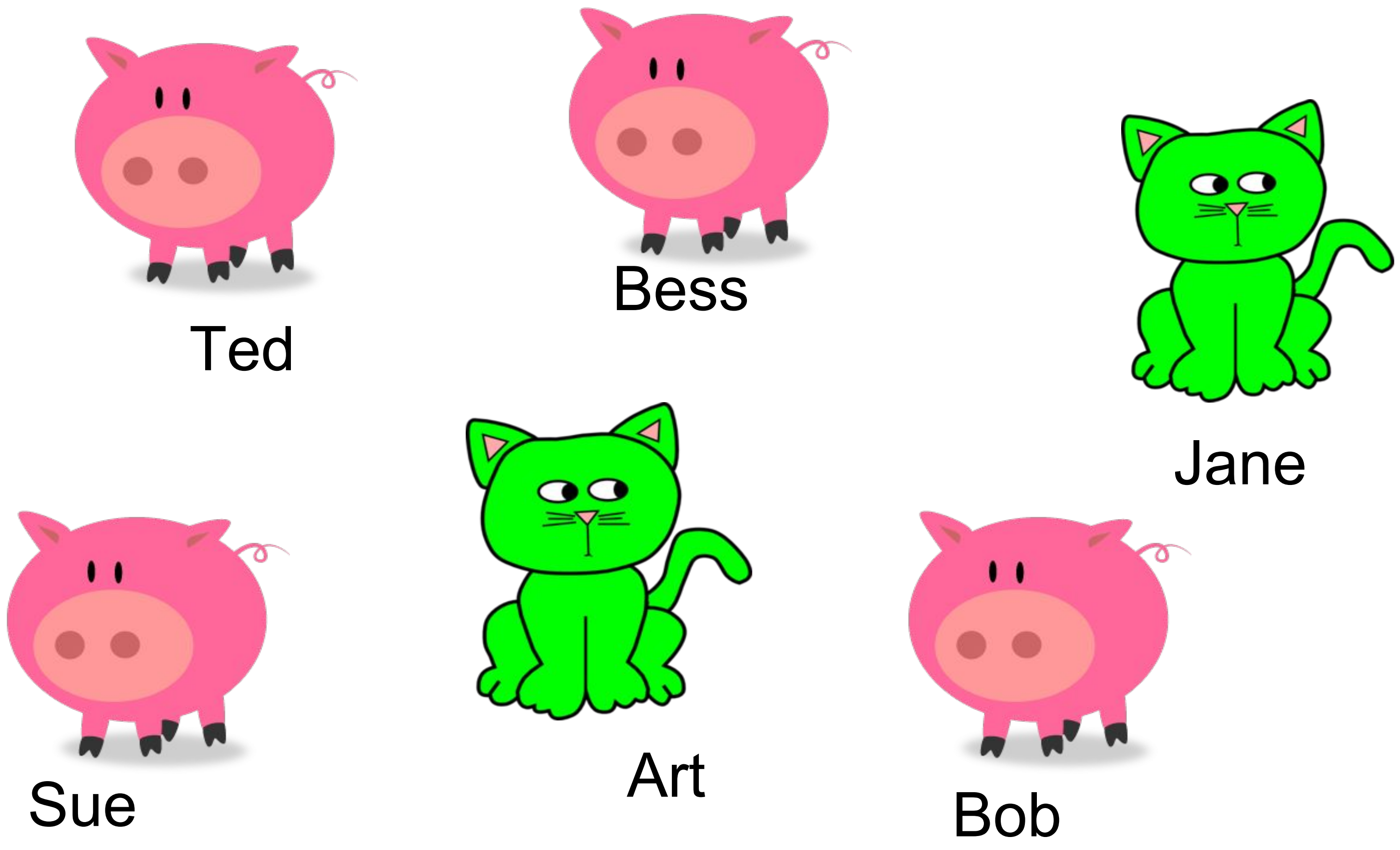
CSS Selectors: # and .

- style **one** with **#id**. ***Unique Identifier*** (have you seen that in databases?)
- style **many** elements with same **.class**. A class can have ***many instances*** (have you seen that in Java yet?)

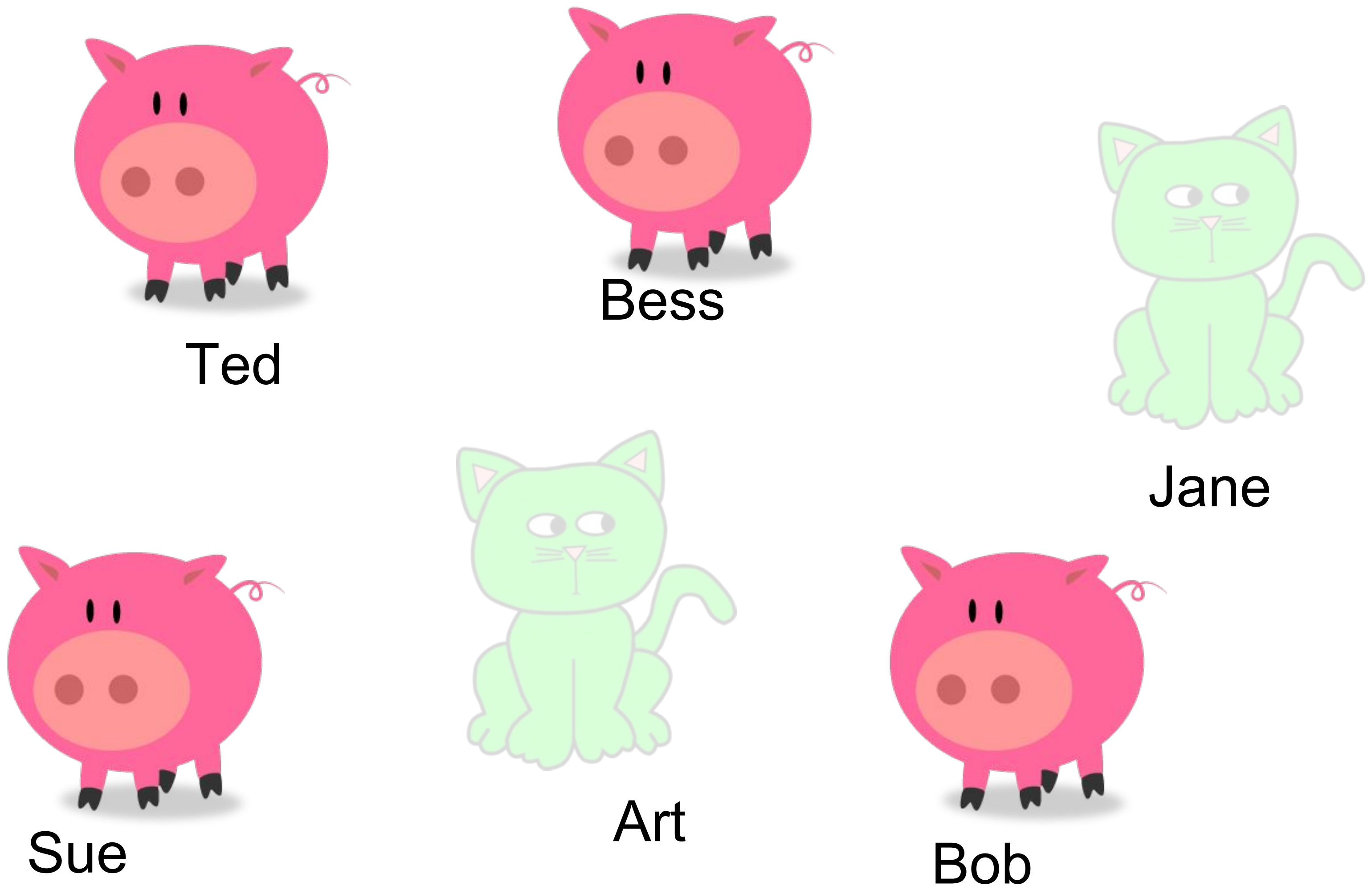
We can assign elements to **many classes** but should give each one only **one id**.

Let's look at an example....

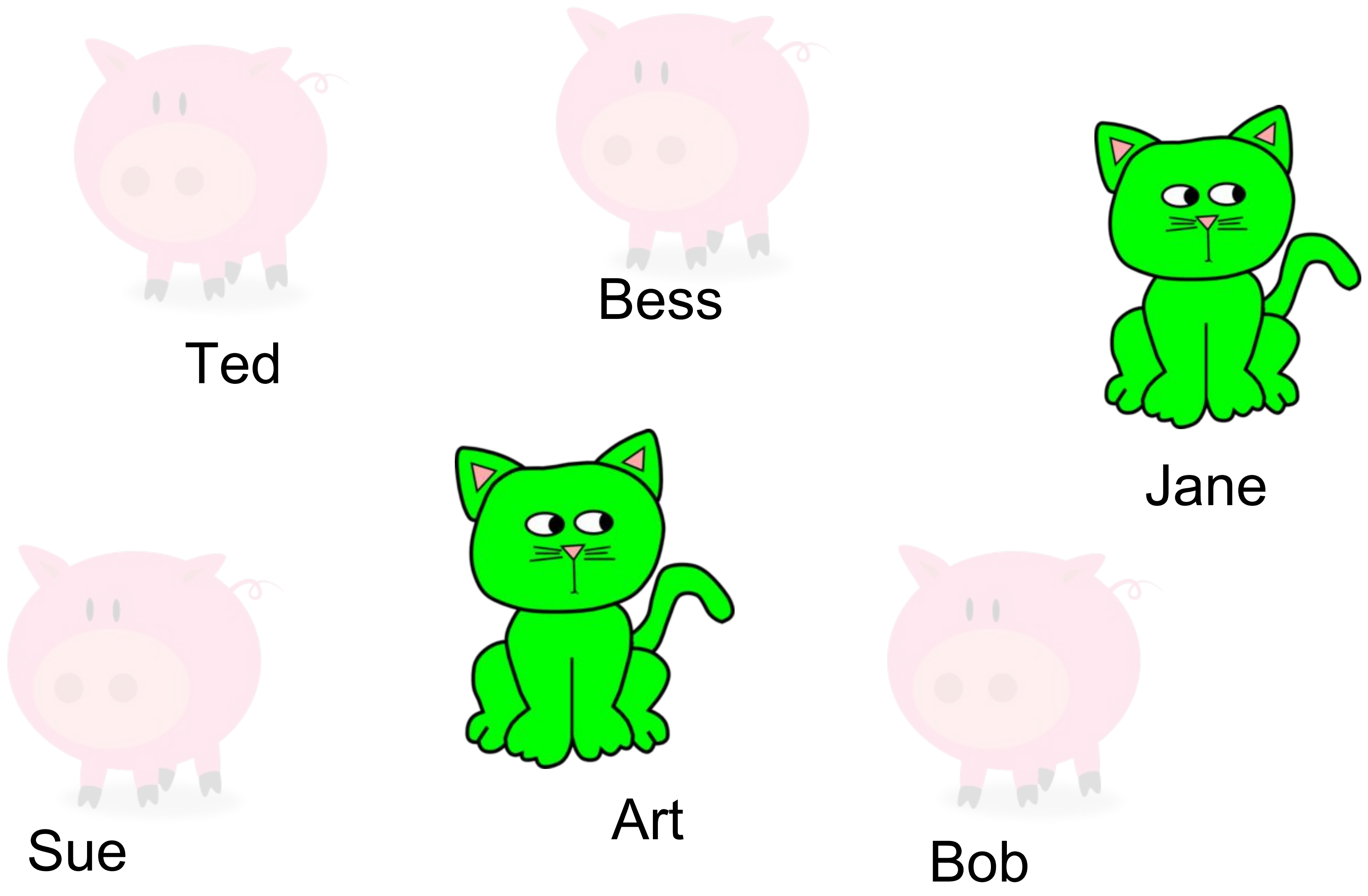
To identify a **particular pig**, what would you use? To identify **all pigs**, what would you use? To identify a particular cat, what would you use? To identify all cats, what would you use? To identify all animals what would you use?



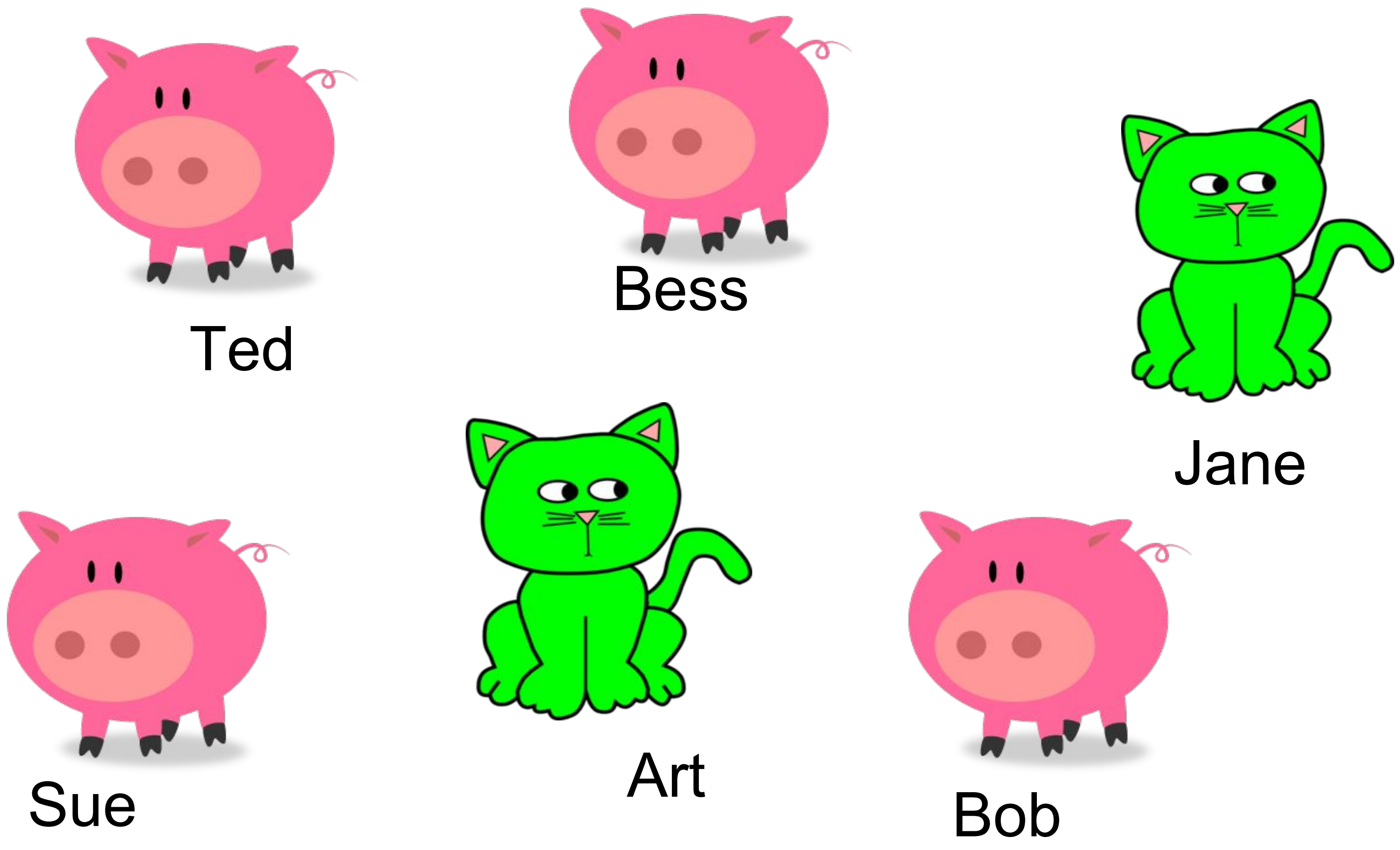
Selector: class; .pig



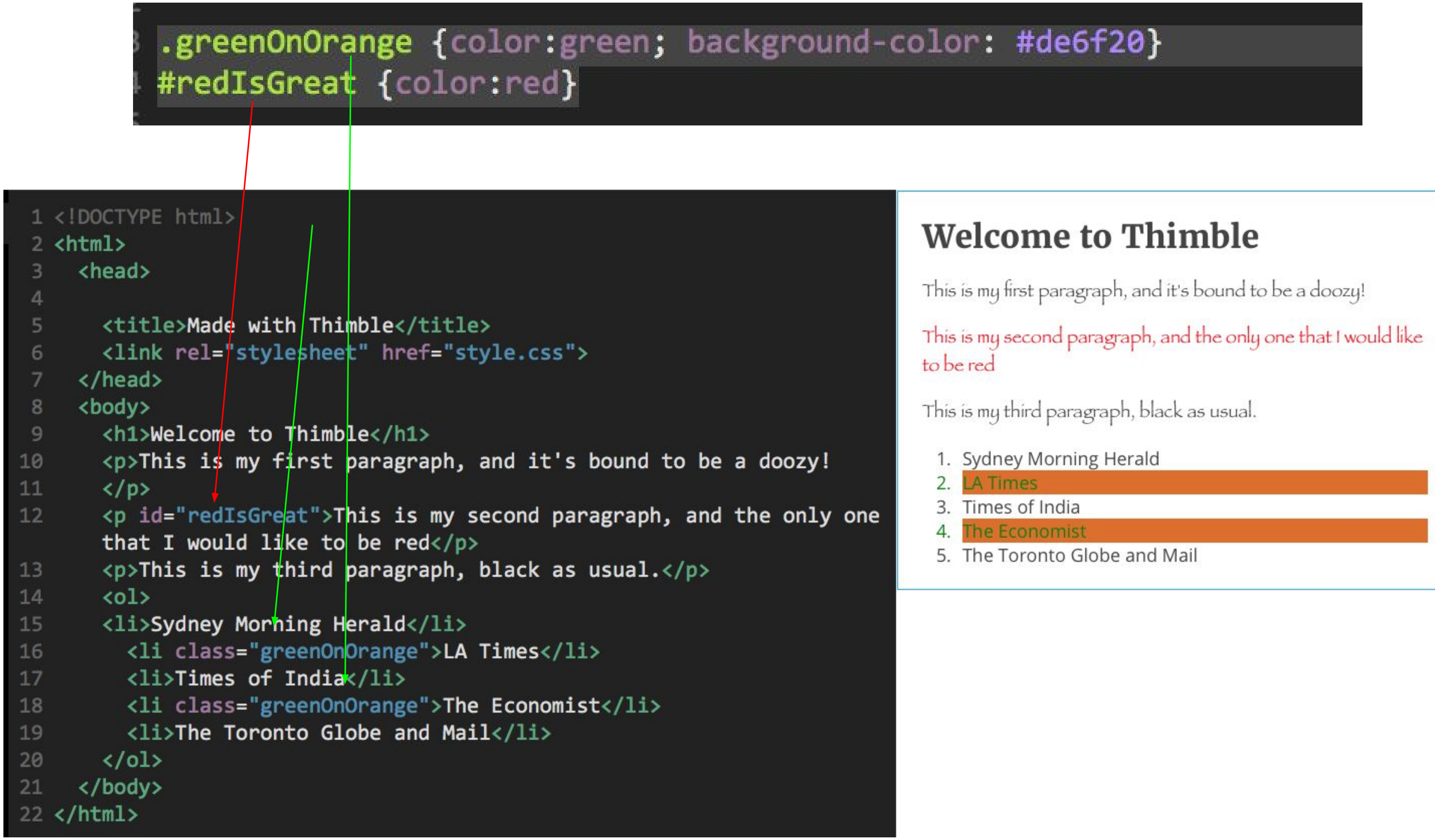
Selector: class; .cat



Selector: class; .animal



A Text Based Example



```css
.greenOnOrange {color:green; background-color: #de6f20}
#redIsGreat {color:red}
```

```html
<!DOCTYPE html>
<html>
  <head>

    <title>Made with Thimble</title>
    <link rel="stylesheet" href="style.css">
  </head>
  <body>
    <h1>Welcome to Thimble</h1>
    <p>This is my first paragraph, and it's bound to be a doozy!
    </p>
    <p id="redIsGreat">This is my second paragraph, and the only one that I would like to be red</p>
    <p>This is my third paragraph, black as usual.</p>
    <ol>
    <li>Sydney Morning Herald</li>
      <li class="greenOnOrange">LA Times</li>
      <li>Times of India</li>
      <li class="greenOnOrange">The Economist</li>
      <li>The Toronto Globe and Mail</li>
    </ol>
  </body>
</html>
```

Welcome to Thimble

This is my first paragraph, and it's bound to be a doozy!

This is my second paragraph, and the only one that I would like to be red

This is my third paragraph, black as usual.

1. Sydney Morning Herald
2. LA Times
3. Times of India
4. The Economist
5. The Toronto Globe and Mail

Example of setting **attributes** in HTML elements

```
<ul>

<li id="Sue"class="pig">Sue</li>

<li id="Ted"class="pig">Ted</li>

</ul>
```

CSS Selectors

- You can assign any HTML element an id attribute (but it is not required).
- You can assign any HTML element to zero, one, or more classes.
- You can reuse classes (e.g. many with class="pig").
- You should not repeat an id (e.g. "Sue")

A class selector, defined.
Notice the presence of the **.**

```
.bigTitle {font-size:xx-large;color:#aa0000;}

<h1 class="bigTitle">Welcome to 44-563</h1>

<p>This class is going to be great</p>

<h2 class="bigTitle">A Few Opening Remarks</h2>
```

A class selector, in use.
Notice the *absence* of the .

Elements with 2+ Classes

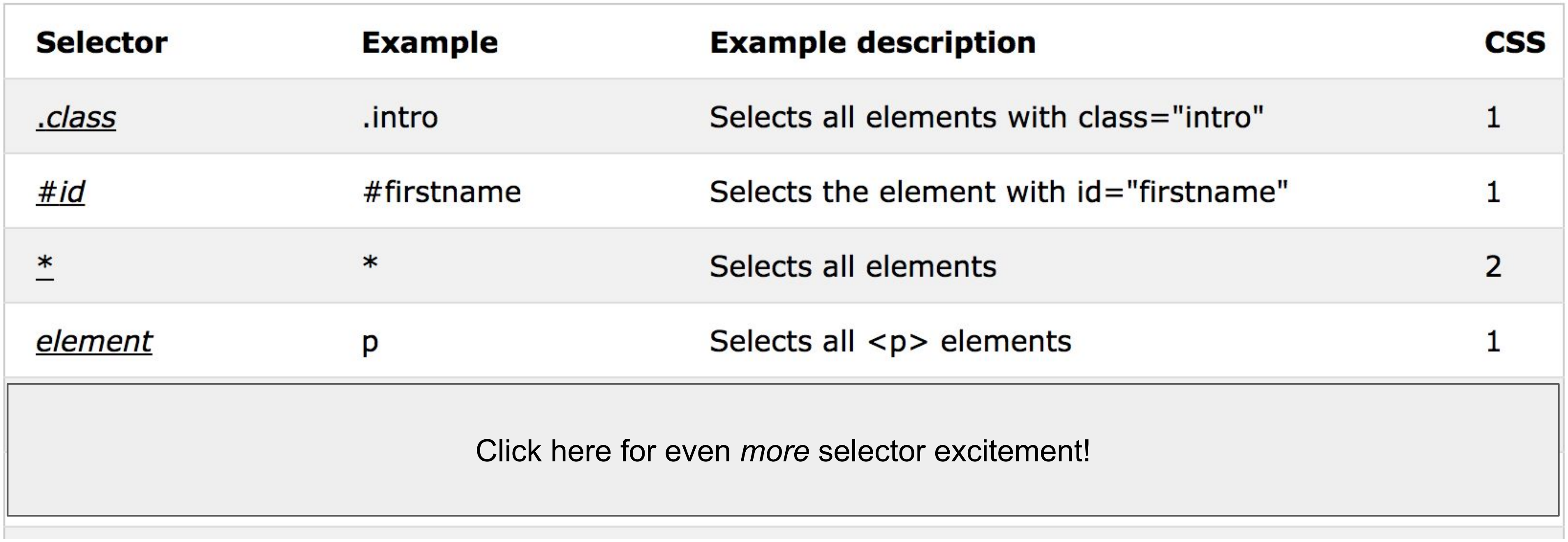
Each HTML element can be associated with many style classes -- a bit like multiple inheritance, allowing an object to acquire characteristics from different classes. Separate multiple classes with a space. We'll use this a lot next week. :)

```
.code {font-family:sans-serif;}
.blue_tinting {color:#5577FF;}


<!-- Meanwhile, somewhere in the <body> -->
<span class="code blue_tinting">
void main(){
     print("I wonder if this will work?");
}
</span>
This should appear in a blue(ish), sans-serif font
```

Mnemonics

- If you have trouble keeping class and id css selectors straight, this *may* help
  - an **id** should uniquely identify something/someone so it can be only used once.
  - If you take an ‘i’ and a ‘d’ and smash them together in a letter-collider (which is like a particle collider, only cooler) then the end result is **#** (just use your imagination).
- ~~For those of you having trouble remembering what a mnemonic is … you probably need a mnemonic for that!~~

CSS Selectors: What does this style affect?



Selector	Example	Example description	CSS
.class	.intro	Selects all elements with class="intro"	1
#id	#firstname	Selects the element with id="firstname"	1
*	*	Selects all elements	2
element	p	Selects all <p> elements	1

Click here for even *more* selector excitement!

See more at: http://www.w3schools.com/cssref/css_selectors.asp

What can we style?

Container HTML elements

- <span> (within a line)
- <div> (line break)

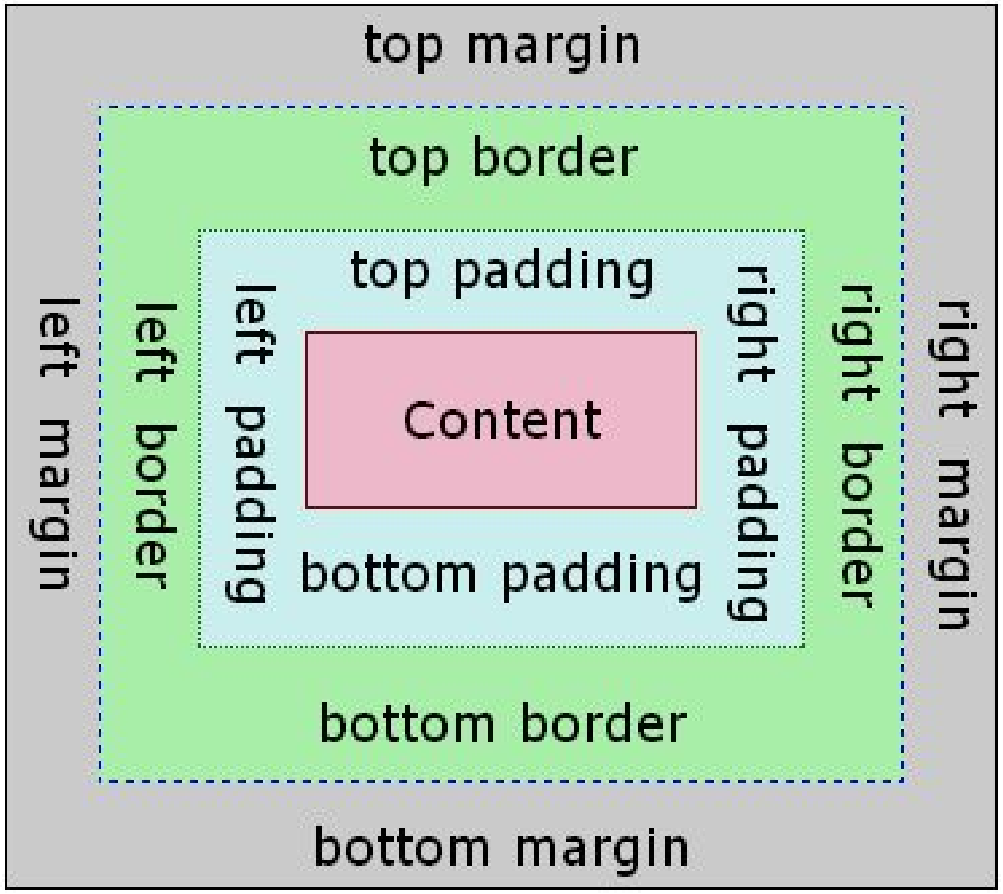
<div **class="instructions"**>
<span **class="important"**> Display</span> your images.
</div>

And all others..

Here we see <span> (as opposed to <spam>) at its finest: providing a mechanism for applying a style to an arbitrary chunk of text.

In HTML, everything's a **box**

Padding is inside the border.



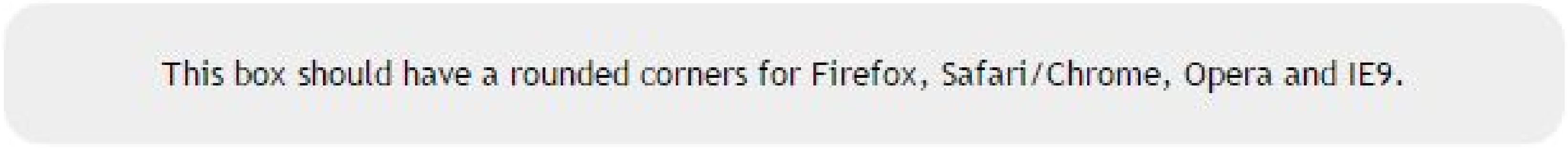
Margin is outside the border.

```
padding: top right bottom left;
```

http://www.codeproject.com/Articles/567385/CSSplusBoxplusModelplusandplusPositioning

Explore CSS Examples

```
#example1 {
    border-radius: 15px;
    border-style: ridge;
}
```



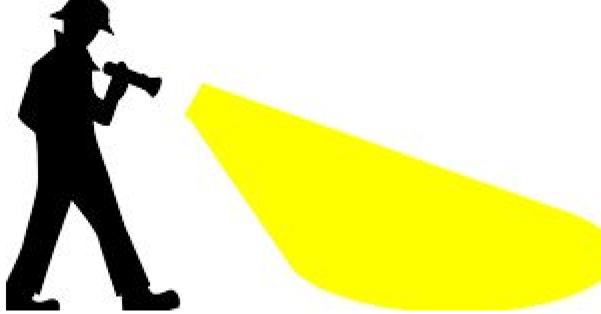
This box should have a rounded corners for Firefox, Safari/Chrome, Opera and IE9.

MDN (Mozilla Developer Network)

https://html5boilerplate.com/ -- awesome templates for kickstarting your own site

http://www.css3.info/preview/rounded-border/ -- details for rounded corners

Explore: CSS Zen Garden



http://www.csszengarden.com/214/

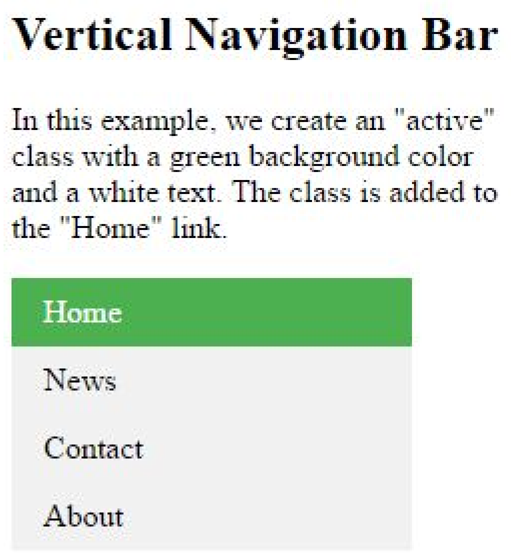
http://www.csszengarden.com/221/

http://www.csszengarden.com/219/

All these pages have the same content, just different CSS (use your favorite number and see what page you get).

https://pixabay.com/static/uploads/photo/2014/04/03/00/32/man-308611_960_720.png

What to know?



Vertical Navigation Bar

In this example, we create an "active" class with a green background color and a white text. The class is added to the "Home" link.

There are a lot of CSS options including really nice effects like **gradients**, **transforms**, and **animations**.

See: MDN CSS3 Tutorials

See: http://www.w3schools.com/css/css_examples.asp

Best List: https://developer.mozilla.org/en-US/docs/Web/CSS/Reference

Colors: http://www.color-hex.com/color-names.html

The **Exam 01 review guide** will include a list of all CSS selectors, properties, and values you should know for this course.

W01

Find a programming partner.
Introduce yourself.

Work together on one notebook
or discuss.

Start a new project folder. **Copy in** your M01 files - and style your web page introduction.

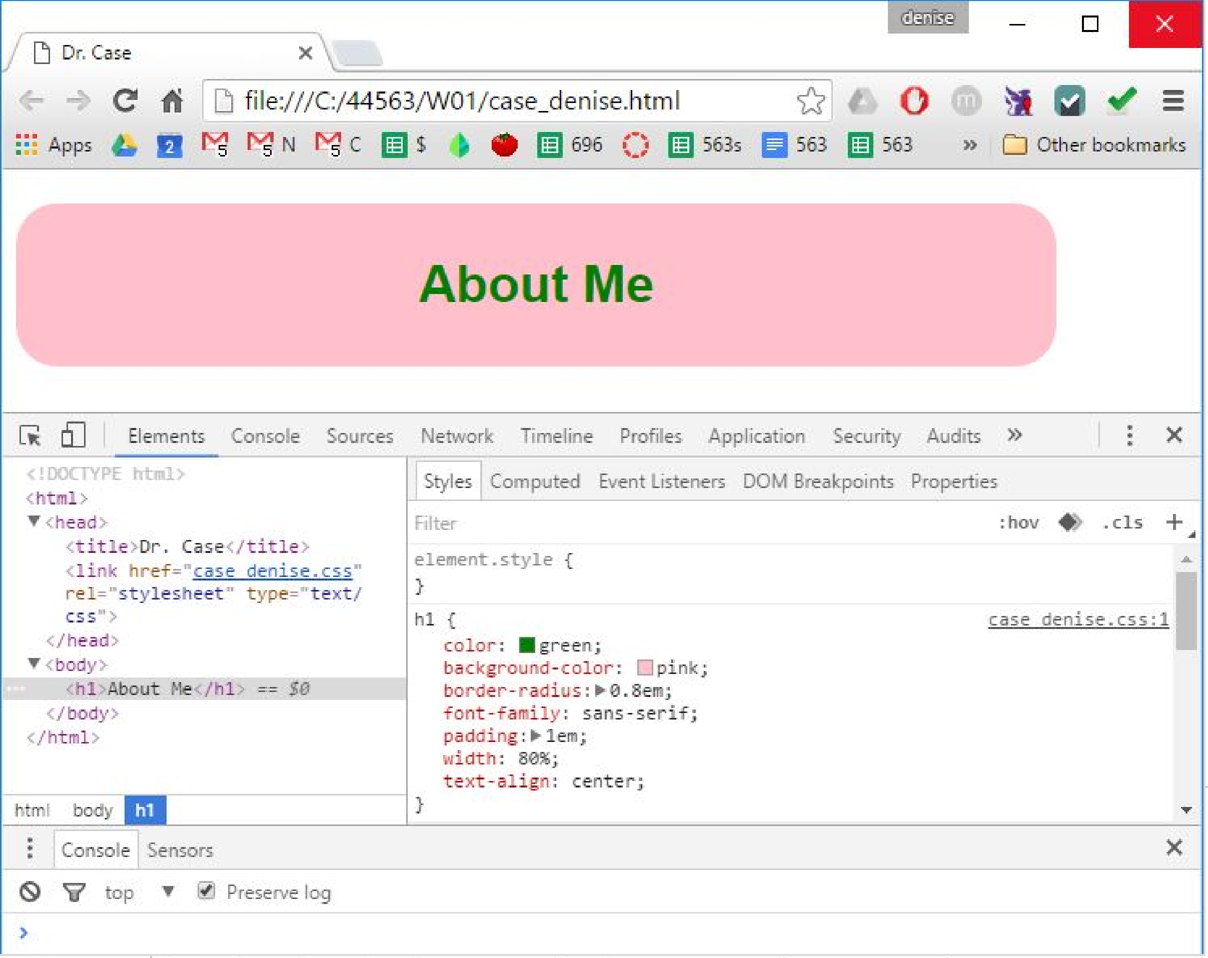
Style some standard HTML tags first (<p>, <h1>..)

Add an **id** attribute to an element and style it (#)

Add a **class** attribute to an element and style it (.)

https://media.giphy.com/media/11JTxkrmq4bGE0/giphy.gif

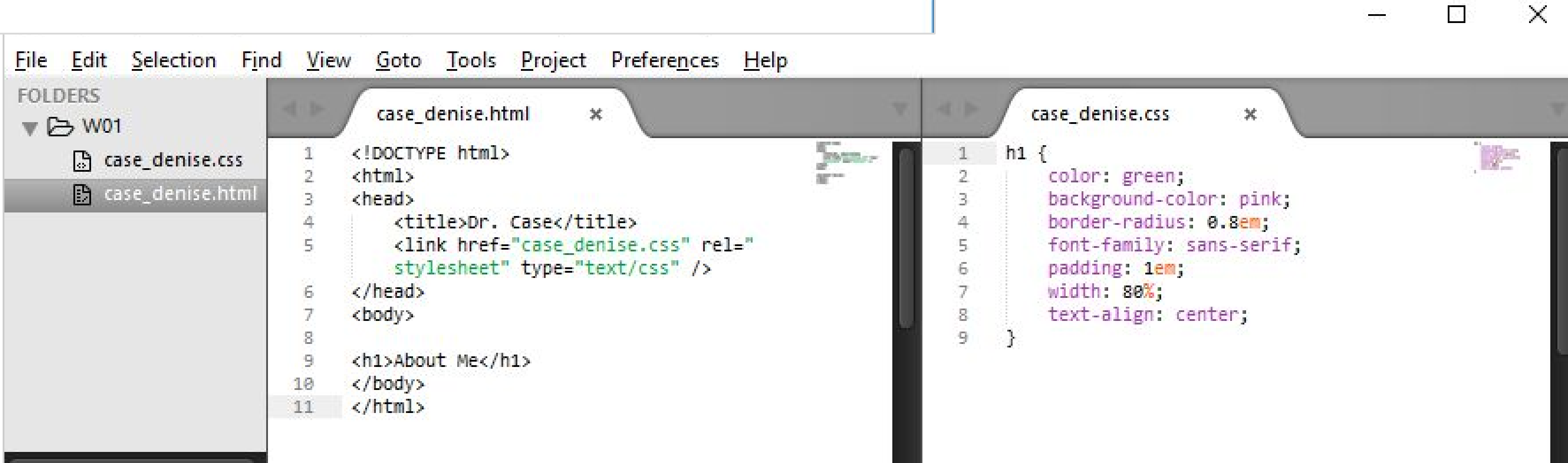
Example W01



```
file:///C:/44563/W01/case_denise.html
```

```
<!DOCTYPE html>
<html>
▼<head>
    <title>Dr. Case</title>
    <link href="case_denise.css"
    rel="stylesheet" type="text/
    css">
  </head>
▼<body>
    <h1>About Me</h1> == $0
  </body>
</html>
```

```
element.style {
}
h1 {                                case_denise.css:1
   color: green;
   background-color: pink;
   border-radius: ▶0.8em;
   font-family: sans-serif;
   padding: ▶1em;
   width: 80%;
   text-align: center;
}
```



case_denise.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Dr. Case</title>
    <link href="case_denise.css" rel="
    stylesheet" type="text/css" />
</head>
<body>

<h1>About Me</h1>
</body>
</html>
```

case_denise.css

```
h1 {
    color: green;
    background-color: pink;
    border-radius: 0.8em;
    font-family: sans-serif;
    padding: 1em;
    width: 80%;
    text-align: center;
}
```

Resources:

JS Bin (live coding) http://jsbin.com/

Code Pen (HTML-CSS-JS) examples http://codepen.io/hello/

Initializer (responsive) http://www.initializr.com/try#

HTML Goodies http://www.htmlgoodies.com/

Thimble https://thimble.mozilla.org

Cloud9 https://c9.io

Optional HTML exploration

Explore and be creative:

MDN

https://developer.mozilla.org/en-US/docs/Web/Guide

W3 Schools

http://www.w3schools.com/html/html_intro.asp

Optional CSS Exploration

http://jgthms.com/web-design-in-4-minutes/

Find free content* at: https://unsplash.com/
https://www.freesound.org

*Please share links to good open source content in your discussion posts!

Friday

Quiz

1. Take it on your own (hardest)
2. You may discuss with your neighbor.
3. We will cover the answers in class.

Review

1. HTML tags, syntax, usage, which are hardest to remember?
2. CSS purpose, how to associate styles with HTML elements, what can be styled, how to set styles, which styles are commonly applied?
3. Terminology - review basic terms

Workshop topics (pick in Wk 2)

1. Accessibility (image alt text, readers, more)
2. Internationalization
3. Security - Cross-site Scripting vulnerabilities
4. Security - SQL injection vulnerabilities
5. Security - Authentication & Authorization
6. Security - other exploits and vulnerabilities
7. JavaScript Promises
8. Dependency Injection
9. Code coverage, static analysis
10. DevOps
11. Jasmine
12. IFTTT.com

Goal: Awareness + your class knows **3 key points + enjoys a demo!**

Workshop - your goal is to teach just **3 key points** during the 10 minutes.

For example:

1. When to use it
2. Why do we care
3. Where to go to learn more

These topics are suggested, but not required - work with your instructor if you want to do something else

Optional: Build your portfolio - You can host an article for free on GitHub Pages. Be sure to credit all resources!

Work in small groups

1. Look at Assignment 1.
2. Break into small groups.
3. Share your name and web development background with your group members.
4. Show your previous assignments and start working on Assignment 1.

No class on Monday!

Labor day is a day off in the US. Photo citation: https://c.tadst.com/gfx/750w/careoftime4.png?1