CSS & HTML Level 1: Week 1

Devon Persing, Instructor
Tiffany Kosa, TA

Browser designed by Fernando Vasconcelos from The Noun Project

What we'll cover

- Major **HTML/HTML5** elements
- Major **CSS** properties and capabilities
- Web site **structure** and **navigation**
- Prepping **images** for the web
- Overviews of related **technologies, issues,** and **techniques**

How we'll cover it

- “Lecture”
- **Discussion** and **questions**
- Writing and editing **code**
- Building **a small website** on a topic of your choosing
- Optional **reading**

Class details

- Thursdays, June 20-July 25 (no class July 4!)
- 6:30-9:30pm (with a break!)
- No grades, no tests, no requirements
- Participate, ask questions, practice

Class resources

Slides, code samples from class, tools, and online resources will be found at dpersing.github.io/svc/

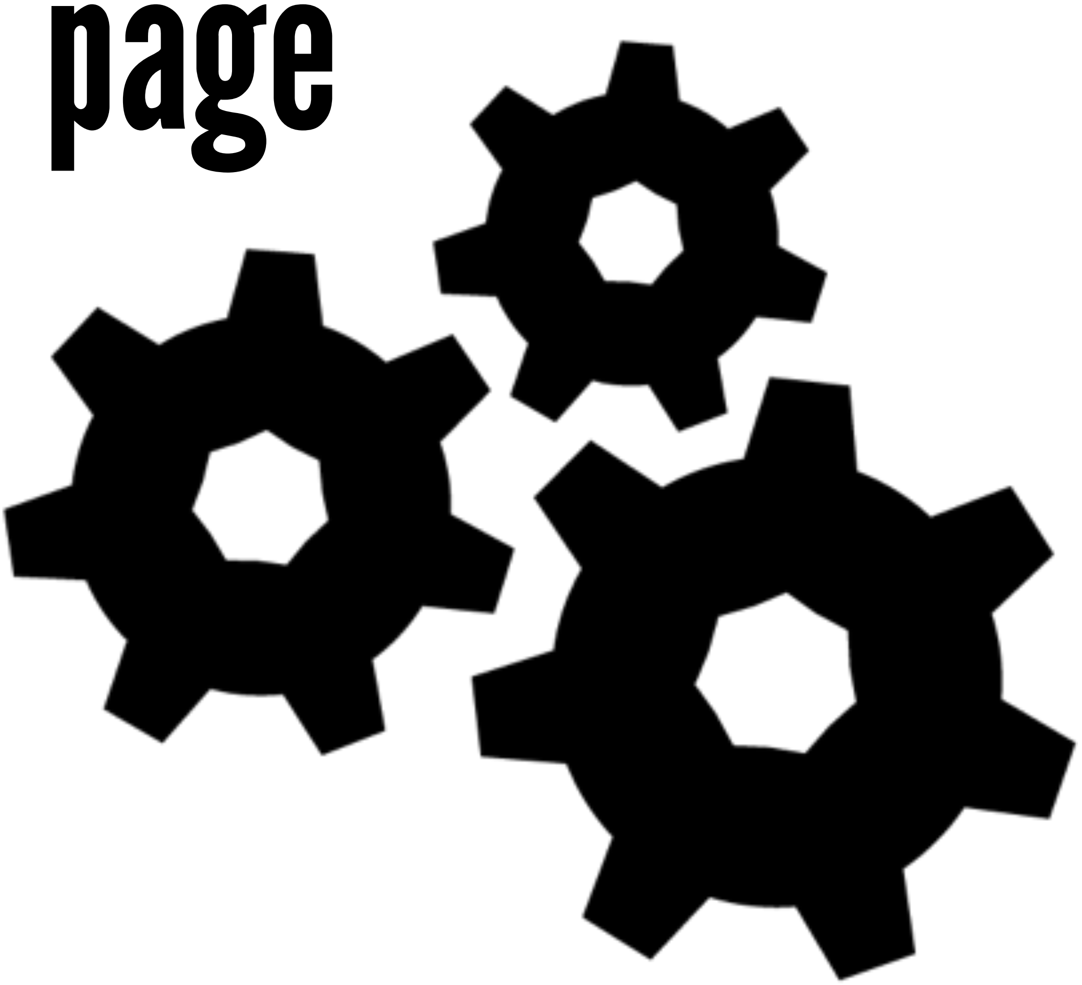
Introductions

- Who are you?
- What do you do/make/etc.?
- What do you want to get out of this class?

Tonight

- **Components and planning** of web pages
- Code editing **tools**
- Basic **HTML elements**
- File **structures and conventions**
- Adding and viewing HTML files in **Dropbox**

Components of a web page



Gears designed by Dima Yagnyuk from The Noun Project

Content. Design. Code.

- **Content** is the reason we make web pages
- **Design** is how we create user experiences and emotional responses
- **Code** is how we deliver content and experience

Content first

Text

- Articles
- Blog posts
- Links
- Captions
- Etc.

Media

- Images
- Video
- Audio
- Interactive experiences

Design second

Experience design and information architecture

- Layout
- Navigation
- User flows
- Labeling

Graphic design

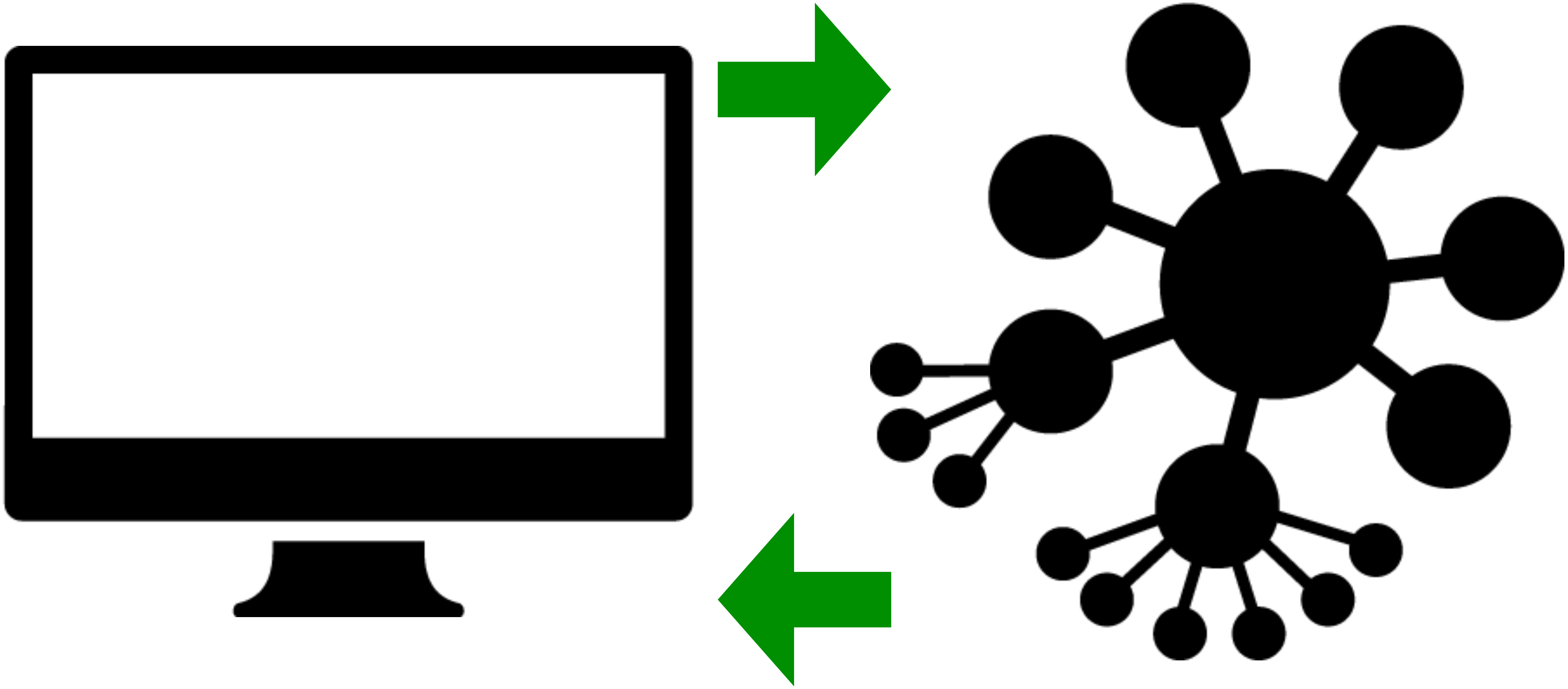
- Colors
- Fonts
- Backgrounds
- Borders
- Icons

Code third

- **HTML*** structures content
- **CSS*** creates style and layout
- **Javascript** adds extra interactivity

***We'll be focusing on HTML and CSS.**

From the web with love



(the internet)

Computer from The Noun Project
Network designed by Jerry Wang from The Noun Project

A brief history of HTML

- HTML was proposed in 1991 and revised until 1999 by the WC3 ↖ world wide web consortium
- XHTML was specced in 2000
  - Good news: CSS and separation of content and style
  - Bad news: Became almost too theoretical and impractical

A brief history of HTML (con’t.)

- **W3C** continued to work on **XHTML**
- **WHATWG** started to work on what we now know as **HTML5**
- These two groups now **work together** (sort of) with different processes
  - **WC3** = Let’s Plan Very Carefully
  - **WHATWG** = Let’s Try It and See

Web Hypertext Application
Technology Working Group

Is HTML5 ready to use?

YES.

Tools

Tools designed by Jeremy J Bristol from The Noun Project

Major browsers

- **Internet Explorer** on Windows
- **Safari** on OSX and iOS
- **Chrome** across platforms
- **Firefox** across platforms

Types of editors

What You See Is What You Get

WYSIWYG editors

- Dreamweaver
- CMS and blogging editors

Text editors

- Sublime Text*
- TextMate
- BBEdit
- Notepad

***We'll be using Sublime Text in class.**

Basic HTML elements

< >

Let’s get started!

```
<!DOCTYPE html>

<html lang=”en”>

      <head>
         <meta charset=”UTF-8”>
         <!-- Author: Me | Date: Today | HTML and CSS Level 1: Week One -->
         <title>My First Page</title>
      </head>

      <body>
         <h1>My First Page</h1>

         <p>Yes, we’re going to write some HTML today.</p>
      </body>

</html>
```

`<!DOCTYPE html>`

- Always comes first
- Tells the **browser** that this is an HTML document
- Older but very common XHTML DOCTYPE:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

Anatomy of an HTML element

```
<p>I'm a paragraph!</p>
```

opening tag

closing tag

HTML elements **almost always** have **opening** and **closing** tags.

Element attributes

```
<a href=”index.html”>I’m a link!</a>
```

attrName=”value”

Some HTML elements will have additional attributes that give them more **meaning**.

The “rules”

- Tags are **lower case***

  `<p>` not `<P>`

- Tags must be **closed****

  `<p>foo</p>` not `<p>foo`

- Attributes must have **values in quotes*****

  `<img src=”img.png”>` **not** `<img src=img.png>` **or** `<img src>`

*** HTML5 technically allows uppercase tags, but it’s considered bad form.**

**** We’ll break this rule for some elements.**

***** HTML5 breaks this rule sometimes, but we probably won’t get that far.**

<html> element

```
<html lang=”en”>
 <!-- everything else -->
</html>
```

The **html** element starts right after the DOCTYPE and closes at the very end of your document.

<head> element

```
<head>
  <meta charset=”UTF-8”>
  <title>Page Title</title>
</head>
```

- Contains **metadata** about the document, **scripts** and **styles**
- Contains the **page title** that displays in the browser
- **Is required!**

<meta> elements

```
<head>
  <meta charset=”UTF-8”>
  <meta author=”Devon Persing”>
  <meta description=”This page is an example to
  show how meta elements work.”>
  <title>Page Title</title>
</head>
```

The **meta** elements describe the document with information like **character set, author, and description.**

<title> element

```
<head>
  <title>Page Title</title>
</head>
```

- Tells the **browser** what to display in the title bar or page tab
- Important for **usability**, **accessibility**, and **SEO**
- **Is required!**

HTML comments

```
<head>
  <meta charset=”UTF-8”>
  <meta author=”Devon Persing”>
  <meta description=”This page is an example to
  show how meta elements work.”>
  <title>Page Title</title>
  <!-- comments go in here -->
</head>
```

HTML comments can be placed anywhere in the HTML document, and are great for leaving yourself and others notes.

Indentation

```
<head>
  <meta charset=”UTF-8”>
  <meta author=”Devon Persing”>
  <meta description=”This page is an example to
  show how meta elements work.”>
  <title>Page Title</title>
  <!-- comments go in here -->
</head>
```

Indenting child elements helps keep your code clean and easy to read.

<body> element

```
<body>
 <!-- all your visible content -->
</body>
```

- Comes right after the **head** element
- Wraps all of the **visible content**
- **Is required!**

<h1> element

```
<body>
  <h1>Page Title</h1>
</body>
```

- Most important element in the **body**
- Typically will match the **title** element of the same page
- Used by **search engines** and **assistive technologies** to identify the page

More header elements

- Range from `<h1>` (most important) to `<h6>` (least important)
- Lesser headers should always come after more important ones
- Provide structure and **semantic meaning** to HTML pages

<p> element

```
<p>Here’s a paragraph. You may remember it from the
slide about closing your tags.</p>
```

- **Paragraph** elements are exactly what they sound like!
- Used to structure **text** in the document
- When in doubt, put a <p> around it

List elements

```
<ul>
   <li>Here’s a list item.</li>
   <li>Here’s another list item.</li>
</ul>
```

- Most lists are of two main kinds:
  - **Unordered**: `<ul>`
  - **Ordered**: `<ol>`
- Lists always contain **list items**: `<li>`
- Lists can be nested within each other

<img> element

```
<img src=”filename.png” alt=”a picture of kitten”>
```

- **Image** elements **do not have closing tags**
- Images have **two required attributes**:
  - **`src`** is where the file lives and its filename
  - **`alt`** is a description of the image

<br> element

```
<p>Here’s a paragraph.
<br>
It has an arbitrary, visual break between sentences.</p>
```

- Break elements create visual line breaks in the browser
- Do not have closing tags
- Use with caution!*

***Next week we’ll talk about separating style from content.**

<a> element

```
<a href=”http://google.com”>Google</a>
```

- **Anchor** or **link** elements create paths that connect HTML pages together
- Link elements are **required to have an `href` element** to tell them where to go
- Content inside the element will be **clickable**

Types of file paths

Absolute paths

- Full URL of the page or file

http://google.com

http://dpersing.github.io/svc/img/svc-logo.png

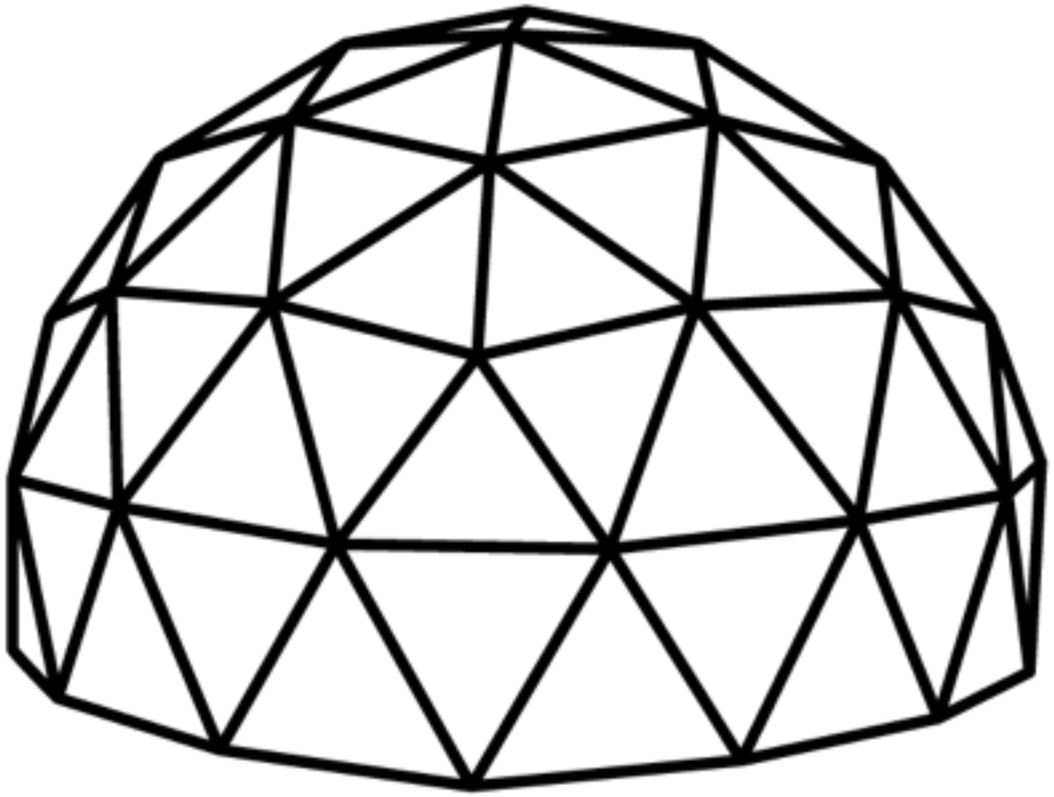
Relative paths

- URL in relation to the file you're in

/svc/img/svc-logo.png

../svc/img/svc-logo.png

File structure and conventions



Rules of file naming

- **No spaces** in file names
- **Capitalization** matters
- Use only **letters**, **numbers**, **hyphens** (-), and **underscores** (_)
- Filenames must **start with a letter**

Recommendations for file naming

- Be consistent
- Use meaningful filenames
- Standardizing your filenames will help prevent errors
- Current standard:

```
svc-logo-small.png
photo-gallery.html
shoes-i-have-loved.html
```

File structure

- **Make subdirectories** for CSS, JS, and media files
- Start with HTML files in the **main directory**
- Standardizing your structure:
  - Makes updates faster
  - Makes links easier
  - Prevents needing to move files around

Organization isn't just for you!

Other people might need to find or edit your files and media, and **other people** will navigate your site.

- Standardized **file structure**
- Standardized **filenaming**
- **Comments** in code (`<!-- these! -->`)
- **Indenting your code** so it's readable

Displaying web pages with Dropbox

Dropbox? Really?

- **Dropbox** is a free file hosting/sharing service
- Dropbox **Public folders** can be used to serve HTML, CSS, and media files
- Dropbox has some basic **version control**
  - Takes snapshots of files periodically
  - Keeps track of when files are moved or deleted

Setting up a Dropbox account

- Go to http://dropbox.com and create an account if you don't have one
- Log in to your account

(If you want to have your Dropbox files locally on your own machine, click on the "Install" button in the Dropbox page footer to download the app.)

Adding a Public folder for class

- Open the **Public** folder in your Dropbox homepage
- Create a folder called **"yourname-svc"**
- **Upload your files** from class to your new folder

View a Public file in the browser

- **Right click** on the filename of the file you want to view
- Select **“Copy public link”**
- **Copy and paste the URL** to your browser window
- See your webpage!

Make a mistake with Dropbox?

- Click on the **“Events”** link to see uploads, deletions, and other changes to your files
- Click on the link for a deleted file to restore it to a previous version
- In your file folder, you can see (and roll back to) previous versions of your existing files

For next week

- Review **HTML5 tags** at http://html5doctor.com/element-index/
- Make a **3-5 page website** and upload it to your Dropbox Public folder
- ***HTML5 for Web Designers***: ch. 1-2
- ***HTML and CSS***: ch. 1-4 (ignore pg. 45 and use pg. 51 instead)

Next week

- Questions and review from week 1
- Prepping images for the web in Photoshop and Illustrator
- Introduction to CSS
- Styling HTML with CSS
- Validating HTML and CSS

Questions?

- Visit http://dpersing.github.io/svc
  - Class slides
  - Code examples from class
  - Additional general and class-specific resources
- Email me at dep@dpersing.com