

# HTML/CSS Workshop

Led by Kelsey Karin Hawley

sometimes a coding veteran  
sometimes a coding newb  
(really depends on the day)

# Me

Engineer in - Python, HTML, CSS, Javascript,  
Django, Flask...

Teacher for - Girls Who Code

Mentor for - 3 cohorts of Hackbright

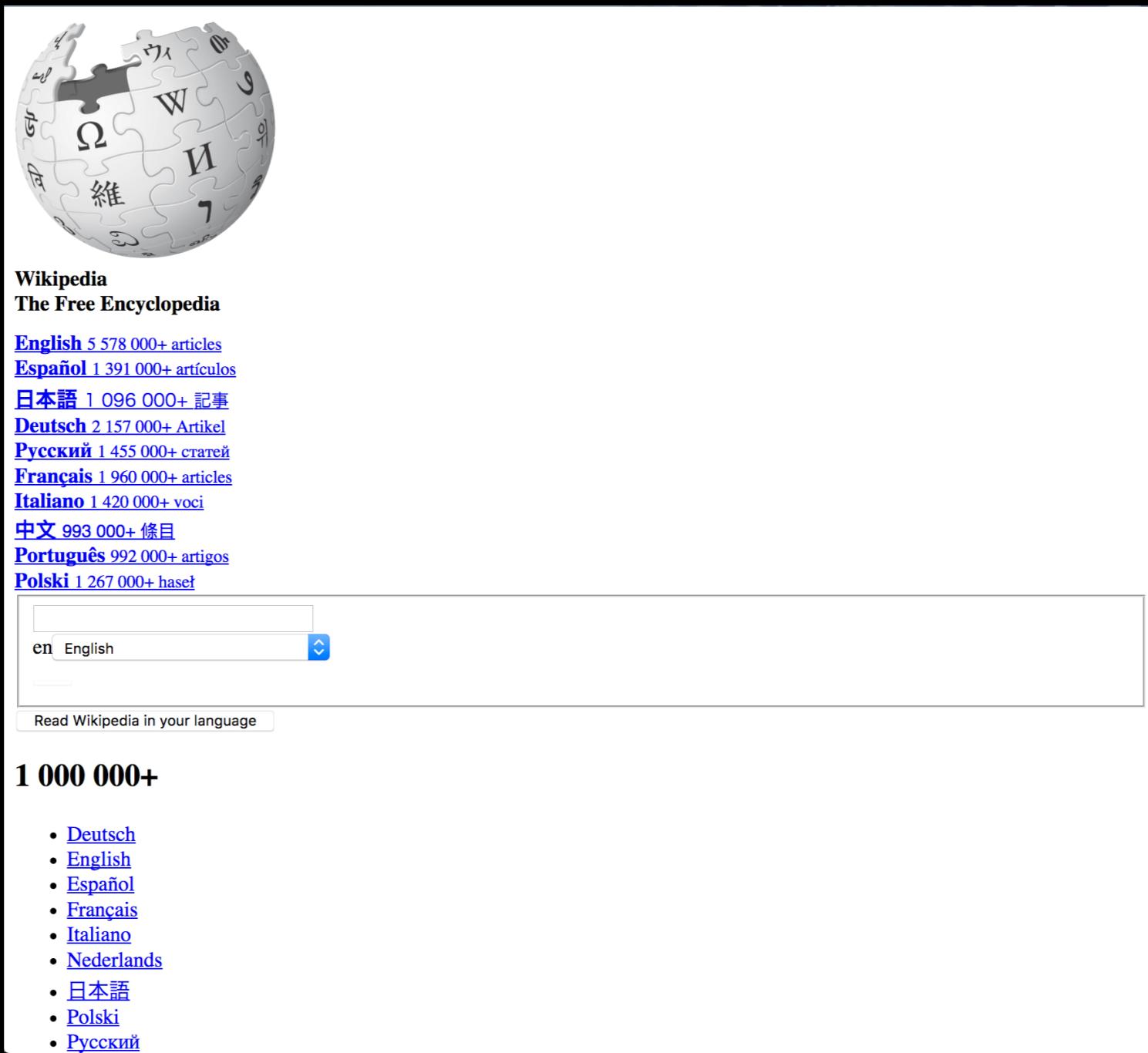
Speaker at - NorthBayPython, DjangoGirls

# HTML

## HyperText Markup Language

# HTML - The skeleton of a webpage

# HTML - The skeleton of a webpage



# CSS

## Cascading Style Sheets

# CSS - The skin and shape of a webpage

# CSS - The skin and shape of a webpage



# HTML + CSS - Page Layout

The image shows two versions of the Wikipedia homepage side-by-side, illustrating how HTML and CSS can be used to create different visual layouts.

**Left Version (Original):**

- Header:** Wikipedia - The Free Encyclopedia
- Image:** A globe composed of puzzle pieces, with a large piece missing from the top-left corner.
- Text:** English 5 578 000+ articles, Español 1 391 000+ artículos, 日本語 1 096 000+ 記事, Deutsch 2 157 000+ Artikel, Русский 1 455 000+ статей, Français 1 960 000+ articles, Italiano 1 420 000+ voci, 中文 993 000+ 條目, Português 992 000+ artigos, Polski 1 267 000+ haseł.
- Search Bar:** A search bar with a dropdown menu showing "en English". Below it is a link "Read Wikipedia in your language".
- Text Block:** "1 000 000+" followed by a bulleted list of languages: Deutsch, English, Español, Français, Italiano, Nederlands, 日本語, Polski, and Русский.
- Footnote:** Wikipedia is hosted by the Wikimedia Foundation, a non-profit organization that also hosts a range of other projects.
- Footer:** Wikipedia apps are now available: Download for iOS on the App Store.

**Right Version (Stylized):**

- Header:** WIKIPEDIA - The Free Encyclopedia
- Image:** A globe composed of puzzle pieces, with a large piece missing from the top-left corner.
- Text:** English 5 578 000+ articles, Español 1 391 000+ artículos, 日本語 1 096 000+ 記事, Deutsch 2 157 000+ Artikel, Русский 1 455 000+ статей, Français 1 960 000+ articles, Italiano 1 420 000+ voci, 中文 993 000+ 條目, Português 992 000+ artigos, Polski 1 267 000+ haseł.
- Search Bar:** A search bar with a dropdown menu showing "EN". Below it is a link "Read Wikipedia in your language".
- Footnote:** Wikipedia is hosted by the Wikimedia Foundation, a non-profit organization that also hosts a range of other projects.
- Footer:** Commons (Freely usable photos & more), Wikivoyage (Free travel guide), Wiktionary (Free dictionary), Wikibooks (Free textbooks), Wikinews (Free news source), and Wikidata (Free knowledge base).

Let's start with  
HTML

# HTML elements

Elements are used to structure the content of a page

They consist of tags and content

```
<tag>content</tag>
```

# HTML tags

<tag>content</tag>

Notice that a complete tag is in **2** pieces:  
an *opening* and *closing* tag

Tags **almost** always come in pairs

# Spacing?

If you know python, you know spaces and newlines matter

HTML and CSS **don't care** about whitespace

But its *good practice* to use indentation for readability

# Code it - <html>

Every file should have these:

- Start every HTML document with  
`<html>`
- And end with the matching  
`</html>`

# Important Note about placing tags

Tags can nest

Be sure to close them in the right order!

```
<first_tag>
  <second_tag>Some text! <second_tag>
</first_tag>
```

# Code it - <body>

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

<body> will contain all of our *viewable* webpage content

# Code it - <p>

```
<html>
  <body>
    <p>My first paragraph</p>
  </body>
</html>
```

<p> is the **paragraph** tag

Notice its nested *inside* the <body> tag

# Code it - <p>

```
<html>
  <body>
    <p>My first paragraph</p>
  </body>
</html>
```

Also notice the *text* inside the <p> tag

We can see it on the screen!

# Code it - more <p>

```
<html>
  <body>
    <p>My first paragraph</p>
    <p>Another paragraph!</p>
  </body>
</html>
```

This time, we've put it *after* our first <p> tag

Notice it shows up *after* it in the webpage

# Code it - <h1>

```
<html>
  <body>
    <h1>This is a header</h1>
    <p>My first paragraph</p>
    <p>Another paragraph!</p>
  </body>
</html>
```

Notice that the text inside is visible again

# Code it - <h1>

```
<html>
  <body>
    <h1>This is a header</h1>
    <p>My first paragraph</p>
    <p>Another paragraph!</p>
  </body>
</html>
```

Notice this header tag, had the number *1*...

# Code it - <h2>

```
<html>
  <body>
    <h1>This is a header</h1>
    <h2>Less big header</h2>
    <p>My first paragraph</p>
    <p>Another paragraph! </p>
  </body>
</html>
```

There are more

Each gradually decreases in size

# All the <h\_>'s

```
<html>
  <body>
    <h1>Biggest header</h1>
    <h2>Less big header</h2>
    <h3>Big medium header</h3>
    <h4>Small medium header</h4>
    <h5>Smaller header</h5>
    <h6>Smallest header</h6>
  </body>
</html>
```

# Code it - <a>

What about links?

# Code it - <a>

What about links?

```
<html>
  <body>
    <h1>This is a header</h1>
    <p>My first paragraph</p>
    <a href="https://google.com">Google</a>
    <p>Another paragraph! </p>
  </body>
</html>
```

# **href ?**

# HTML Attributes

**Attributes** are modifiers on an html tag

- Some attributes are only on some tags  
(like href on <a> )
- Others will be on many tags  
(like class we'll see later)

# HTML Attributes

Usually look something like:

```
<tag attribute="something"></tag>
```

# <a> href?

It stands for “hypertext reference”

We can see in our file, its a link to google

Is that all it can do?

# <a> href

- An external html page

```
<a href="https://google.com">Google</a>
```

# <a> href

- An external html page

```
<a href="https://google.com">Google</a>
```

- An external file link

```
<a href="http://superposition.tech/assets/images/logo.png">Logo</a>
```

# <a> href

→ An internal link

```
<a href="new_page.html">New page</a>
```

→ An internal file link

```
<a href="cartoon.jpg">some random image</a>
```

# Code it - <ul>, <li>

```
<p>Groceries</p>
<ul>
  <li>Oranges</li>
  <li>Apples</li>
</ul>
```

Each `<li>` (list item) is part of the main `<ul>` (*unordered* list)

# Code it - <ol>, <li>

```
<p>Tasks</p>
<ol>
  <li>Wake up</li>
  <li>Brush teeth</li>
  <li>Get dressed</li>
</ol>
```

Each <li> (list item) is part of the main <ol> (*ordered list*)

# Code it - nesting lists

You can also nest lists

Just be sure close them properly!

# Code it - nesting lists

```
<p>Groceries</p>
<ul>
  <li>Oranges</li>
  <li>Apples
    <ul>
      <li>Fuji</li>
      <li>Pink Lady</li>
    </ul>
  </li>
</ul>
```

# Code it - nesting lists

```
<p>Tasks</p>
<ol>
  <li>Wake up</li>
  <li>Brush teeth</li>
  <li>Get dressed
    <ol>
      <li>Socks on</li>
      <li>Shoes on</li>
    </ol>
  </li>
</ol>
```

# Code it - <b>, <i>

Bold and italic texts

```
<p>Some <b>bold</b> text.  
And <i>italic</i> too.</p>
```

# Images

Are a bit special:

- No closing tag
- No inner content
- Require a `src` attribute
- `src` is a link, like with `href`

# Code it - <img>

```

```

# Code it - <a>, <img>

You can add a link to your image by nesting in an <a>

```
<a href="https://google.com">  
    
</a>
```

# Code it - comments

You might want to leave notes in your code

These are hidden on the webpage,  
and just seen in the code

```
<!-- comments galore here -->
```

# More body tags you might see

- `<div>` used to format and divide content
- `<table>`, `<tr>`, `<td>` for creating data tables
- Sooooo many more

# The head

Remember <body> was full of the *visible* bits?

Well, <head> has lots of *invisible* bits

# Code it - <head>

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

```

# The head

Head can contain things like:

- page title (seen in the browser tab)
- language (spanish, german, english, ...)
- javascript things
- css things
- search/social media sharing things

# Code it - <title>

```
<html>
  <head>
    <title>Tutorial Page</title>
  </head>
  <body>
    ...
  </body>
</html>
```

# CSS

# CSS Rules

CSS is based on a series of rules

Each rule tells the browser:

- which element
- what part of the element is affected
- how it is affected

# CSS Rules

```
p {  
    color: green;  
}
```

# CSS Rules

```
p {  
    color: green;  
}
```

Notice the syntax

- '{' and '}' surround the *properties*
- each *property name* is followed by a ':'
- each *property value* is followed by a ';'

# CSS Rules

```
p {  
    color: green;  
}
```

P is the *selector*

Says which element the rule is touching

Here, all <p> tags would be affected

# CSS Rules

```
p {  
    color: green;  
}
```

**color** is the *property name*  
**green** is the *property value*

# CSS Rules

```
p {  
    color: green;  
}
```

The how and what part of the element

The *color* will be affected by appearing *green*

# CSS Rules

```
p {  
    color: green;  
}
```

As a whole, the rule reads:

All <p> tags will be **green** in **color**

Okay, but where does  
that go?

# CSS options

Three ways to add CSS to your HTML:

- inline
- in a style tag
- linking an external file

# CSS options

Three ways to add CSS to your HTML:

- **inline**
- **in a style tag**
- **~~linking an external file~~**

# CSS style tag

Let's add a style tag to our head

```
<html>
  <head>
    <style>
      ...
    </style>
  </head>
```

# CSS style tag

Now let's add that CSS rule from earlier

```
<html>
  <head>
    <style>
      p {
        color: green;
      }
    </style>
  </head>
  ...

```

What if I only want  
one green <p>?

# More selectors - classes

class is an attribute for any html element

You can group elements in different ways

# More selectors - classes

```
<p class="green">Another paragraph! </p>
```

# More selectors - classes

```
p.green {  
    color: green;  
}
```

.green says the <p> will have a green class

# More selectors - classes

```
<p class="green bordered">Another paragraph!
  <i class="bordered">Italic</i>
</p>
<p class="special">Special paragraph</p>
```

You can add multiple classes to an element

Just separate them with a space

# More selectors - classes

```
p.green, .special {  
    color: green;  
}  
.bordered {  
    border: 1px solid black;  
}
```

You add multiple class selectors to a rule

Even leaving off an element

# More ways to select

- **ids** (#so-special)
- **pseudo-classes** (:link, :active, :hover, etc)
- **attributes** ([href\*="google"])
- **siblings** (a + p)
- **children** (p > i)
- **descendants** (p i)

# Further Resources

MDN HTML

MDN CSS

W3Schools HTML

W3Schools CSS

CSS Box Model

Bootstrap

# Gratuitous Plug

Kelsey Karin Hawley

- Github <https://github.com/khawley>
- These slides  
<https://github.com/khawley/my-talks>
- Honor <http://joinhonor.com>