

Welcome!
Starting 7:15 due
to rain

Front

Heidi	Jessie
Brenda	Linus
Jessica	Steven
Nicole	Waitlisted/ Auditors

Announcements

- **No lecture next Monday :o** – (We'll still be meeting on Wednesday)
- **Post questions on Piazza** for a quicker response, please
(Really hard to debug over email)
Email us if you have **logistical** questions
- **Common Bugs and Errors**
 - Do **NOT** rename your zip file when submitting to **'hw-2.zip'**
 - Make sure your HTML file is properly named **'index.html'**
 - Ensure your image names are properly referenced (it's CaSe-SeNsItIvE!)

Week 2

Hands on: CSS, Selectors, and Pseudo-Selectors

strawpoll.me/17426682

First, a Quick Recap on CSS

HTML and CSS: At a Glance

```
<!doctype html>
<html>
  <head>...</head>
  <body>
    <p>Welcome to WDD!</p>
    
  </body>
</html>
```

HTML

```
body {
  width: 200px;
  text-align: center;
}

p {
  font-size: 32px;
  color: green;
  font-family: Karla;
}
```

CSS

Selectors

We use **selectors** to describe (or “select”) the HTML elements for which we want to create rules. Just like HTML tags, they can describe parent-child relationships, ids, classes, and attributes.

In this case, we:

- Select the **body** tag
- Select all **p** tags
- Apply rules to each!

```
body {  
    width: 200px;  
    text-align: center;  
}  
  
p {  
    font-size: 32px;  
    color: green;  
    font-family: Karla;  
}
```

CSS

Chaining Selectors

Sometimes, we want our selector to be more specific than just one tag name, class, id, attribute, or other piece of information. When that's the case, we can chain selectors together! A few examples:

- `div.hello` selects `div`s with at least `class="hello"`
- `div.hello.world` selects `div`s with at least `class="hello world"`
- `#hi.there` selects elements with `id="hi"` and at least `class="there"`
- `p#text` selects the `p` tag with `id="text"`

We'll also show you some more selectors you can use soon!

Rules

CSS works declaratively--that is, it tells the browser, “I declare that **x** is true for **y**,” where **x** is our set of rules, and **y** is our selector.

How do we define rules?

- The **property** tells us **what** we are changing about the element (e.g. padding, color, font-size).
- The **value** tells us **how** we want to change that property (e.g. 10%, red, 20px);

```
body {  
  width: 200px;  
  text-align: center;  
}  
  
p {  
  font-size: 32px;  
  color: green;  
  font-family: Karla;  
}
```

CSS



State your business!

(Please indulge the bad pun.)

Where does state come from?

State in web pages and web applications can come from many places:

- How the user interacts with the website (clicking, hovering, etc.)
- How the webpage is displayed (which browser is being used, the size of the window, what time it is, the scroll position, etc.)
- Data coming from the backend (don't worry about this for now!)

We'll be most focused on the first source of state in this class--ask us about the rest during lab if you're curious!

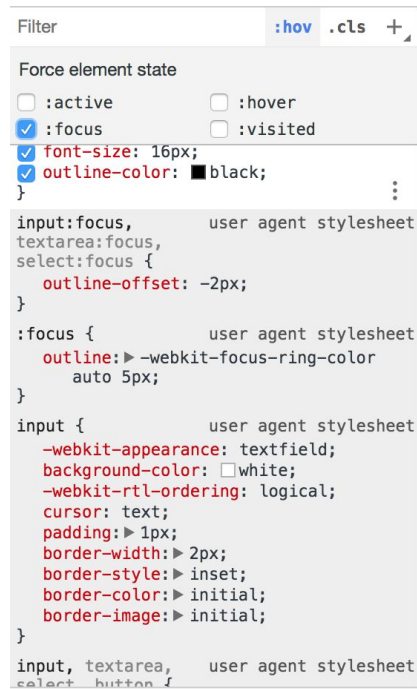
State in CSS: Pseudo-Selectors!

Pseudo-selectors allow us to select only the elements on a webpage that have a specific state. A few examples:

- To select for something that the user is hovering over, use **:hover**.
- To select a link that the user has already visited, use **:visited**.

```
/* Hover over element */  
.item:hover { /* ... */ }  
  
/* Element in "focus" */  
.item:focus { /* ... */ }  
  
/* Unvisited link */  
.item:link { /* ... */ }  
  
/* Link already visited */  
.item:visited { /* ... */ }
```

CSS



Even more pseudo-selectors!

Pseudo-selectors can do much more than understand how a user interacts with a webpage--they also can help us understand parent-child relationships (alongside other selectors that we've not yet taught you).

- `:nth-child(n)` selects the `n`th element in a container
- `:only-child` selects elements that are the only child of their parent
- `:empty` selects elements that have no children
- And more! Read the [full w3schools spec here](#) (you don't need to know all of this!).

CSS Selector Game (with Sushi!)

flukeout.github.io/



Questions?

Get there
Your day belongs to you



Lab 2: WDD-yft

Our Rides

Tesla Model X



Travel in the future,
to the future

Van



What a throwback
to the good old
days

Classic



Some classic style,
for your classy
style

Rollerblades



Quick, affordable,
and fun!



Attendance