Layout

Class 7

Selectors

- a bit more about selectors
- you already know about

- type selector p
- ID selector #important
- class selector .important
- pseudo-class selector :hover
- pseudo-element selector ::first-line

- you can combine selectors with a comma p:hover, #important

Selector Combinators

- the space combinator selects elements that are descendants of the first element
  `div span` will match all span elements that are inside a div
- the child combinator $>$ selects elements that are direct children of the first element
  `ul > li` will match all li elements that are nested directly inside a ul element
- the general sibling combinator $\sim$ selects elements that are siblings of the first element (i.e., they have the same parent)
  `img ~ span` selects all span elements that come after an img and have the same parent as the img
- the adjacent sibling combinator + is the same, except the second element must come immediately after the first
  `h2 + p` selects a p if it is the very first thing after an h2, in the same block as the h2

Attribute Selectors

- there are also attribute selectors that match elements based on the presence or value of a given attribute

- `a[title]` selects all links that have a title attribute set, regardless of the value of the attribute
- `[text-color="red"]` selects all elements that have the text-color attribute set to red
- there are more complex forms of the attribute selectors

personally, I rarely use these, as I feel that classes do the same thing in a way that is more readable and logical

Grid

- in the bad old old days, web designers used tables for layout: terrible
- in the more recent old days, we used floats, positioning, and inline-block for layout: much better, works well for simple things
  but for complicated stuff still feels like hacking it together

- a modern layout system is flexbox
  we don't have time for it this semester, but it's easy to read about

- we will now learn a bit about the other modern layout system, grid
  most flexible, most comprehensive, most complex