Last CSS Powerpoint

Never? Week 3 ½? Who knows?

CSS Units and Values

- Many properties used in CSS, like width and height, use numerical values
  - Setting these require units, either absolute or relative
    - Absolute - (pixels, mm, cm, in, pt (1/72 inches))
    - Relative - em (width of one character in current element), this is the only relative units we'll use this year
- Sometimes values can be unitless
  - Using the number zero for a value doesn't require units
  - Line spacing (as in double spaced that's usually used in English class) is most often unitless
- Values can also be used as percents
  - An object with a width of 50% will always take up half the width of the screen, no matter how wide the screen use. This makes websites able to be viewed on any screen
  - http://codepen.io/cgodfrey/pen/jVbLOw

More Values Stuff

- Colors can be defined as predefined colors, rgb or hexadecimal
  - There 165 color names that can be used, like “red”, “green”, “blue”, etc.
  - Hexadecimal gives you access to 16 million colors
    - Is a # followed by six characters 0-9 and a-f
    - #000000 is black, #ffffff is white, first two is red, second two is green, last two is blue
  - Rgb works the same way but makes it clearer where the color comes from
    - rbg(red, green, blue) where each value is a number 0-255.
    - 0 means none of that color, 255 means all of that color
    - There’s also rgba, where a is how opaque/clear the object is (probably not used by us)
- http://codepen.io/cgodfrey/pen/WoQdKd is an example of each type of color

CSS Functions!

- Like in any other programming language, they can be used to calculate something used later on
  - Can’t be defined, all useable functions are already defined
  - rgb()/rgba() are both functions
  - Other useful functions are transform() and calc()
  - http://codepen.io/cgodfrey/pen/GNpyPW

Cascading!

- When a website gets complex with many selectors and properties, it's common for multiple rulesets to apply to the same element. So how does the browser know which ones apply?
    - Rules apply in order based on three conditions: importance, specificity, and source order
        - Importance - you can append !important to a property and it will override EVERYTHING
            - ***DO NOT USE UNLESS ABSOLUTELY NECESSARY***
        - Specificity - IDs are most specific, then classes/other attributes, then tags
        - Source order - Rules further from the top of the file override rules earlier on
        - The rules only apply to individual properties, not entire rulesets
    - Good example: http://codepen.io/cgodfrey/pen/oYjEbz
        - Normally both would be blue, because that background color is the furthest from the top
        - But class and id are both more specific, so those rules override it
        - #winning should override .better and give both a border (more specific), but the .better ruleset has a !important which overrides EVERYTHING no matter what

Inheritance

- Similar to inheritance for classes, CSS has inheritance for properties
  - Some properties, like font-familiy and color, cascade down from an element to its children, so that it doesn’t have to be rewritten over and over again.
  - On the other hand, padding and margin don’t cascade down because those elements rarely should share those properties
  - Inheritance can be controlled by keywords: inherit, initial, or unset.
    - Inherit - value same as that of the parent ex: `color: inherit;`
    - Initial - use default value in browser stylesheet ex: `color: initial;`
    - Unset - make the property do what it does normally: inherit if it normally inherits, initial if it doesn’t

Advanced Box Model Manipulation

- https://codepen.io/cgodfrey/pen/zovJOa
- When the box boundaries are set with absolute values, what should happen when the content doesn’t fit?
- Auto: The extra content is hidden and a scroll bar is displayed
- Hidden: The extra content is hidden and is never seen again
- Visible: The extra content overflows outside the element (this is default but usually not wanted)

Types of CSS Blocks

- This can get very frustrating, so know these well
  - Block box - each box has it’s own line, and have a set width and height
  - Inline box - impossible to set width/height, flows within a line of text
  - Inline-block - Combination of those above, it can have a set width/height, but can also flow with text inline
- https://codepen.io/cgodfrey/pen/MbaqvG