Css

* Content/presentation
* Improves content accessibility
* Need to review slides
* Cascading order
  + All the styles will “cascade” into a new “virtual” style sheet by the following rules
    - 1. Inline Style (Inside an HTML element)
    - 2. Internal Style sheets (in the head section)
    - 3 External Style sheets
    - 4. Browser default.
* Colors in CSS are most often specified by
  + A valid color name
  + And RGB value
  + And HEX value
* CSS backgrounds
  + Background-color
  + Background-image
  + Background-repeat
  + Background-attachment
  + Background-position
* CSS3 allows you to add multiple background images for an element, through the background image property
* The different background images are separated by commas and the images are stacked on top of each other.
* The CSS background-origin property specifies where the background image is positioned
* The property takes three different values
  + Border-box (default) the background is painted to the outside edge of the border.
  + The border-style property specifies what kind of border to display
  + Dotted- defines a dotted border
  + Dashed-
* Border-width specifies width
* Border-color specifies color
* Border-radius makes rounded borders

Css lists

* Use classes to make different lists, this way you can use ul.name or ol.name for different styles
* List-style-tyle
* List-style-image
* List –style-position
* In certain shitty browsers named internet explorer, your images in your lists may appear off.
* Css box model
  + All HTML elements can be considered as boxes, in CSS the term boxmodel is used when talking about design and layout.
* Css Outline
  + Style property
  + Dotted, dashed, solid, double, groove, ridge, inset, outset
* Display property
  + The most important Css property for controlling layout
  + The display property specifies if/how an element is displayed
  + Every HTML element has a default value depending on what type of element it is; the default display value for most elements is block or inline
  + Display: none; is commonly used with JavaScript to hide and show elements without deleting and recreating them.
  + You can override the default
  + Changing an inline element to a block element, or vice versa, can be useful for making the page look a specific way, and still follow the web standards.
  + A common example is making inline <li> elements for horizontal menus:

Li

{

Display:inline

}

* Hide an Element
  + Hiding an element can be done by setting the display property to none.
  + Visibility:hidden; also hides an element; however, the element will still take up the same space as before and will still affect the layout.
* The position property
  + Static positioned elements are not affected by the top, bottom, left and right properties; they follow the flow of the page
  + For relative positioned elements, setting the top, right, bottom, and left properties of a relatively-positioned element will cause it to be adjusted away from its normal position
  + An element with position fixed is positioned relative to the viewport, which means it always stays in the same place even if the page is scrolled; the top, right, bottom, and left properties are used to position the element.
  + An element with position absolute is positioned relative to the nearest positioned ancestor (instead of relative to the viewport, like fixed).
* Overlapping elements
  + When elements are positioned, they can overlap other elements
  + The z-index property specifies the stack order of an element
    - Which element should be placed on top, or behind other elements.