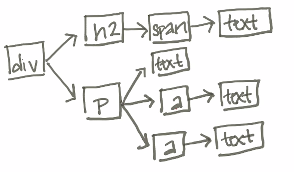
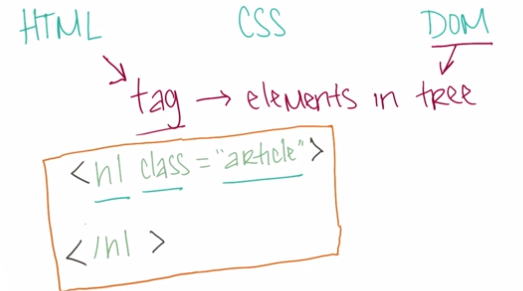
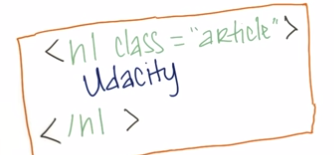
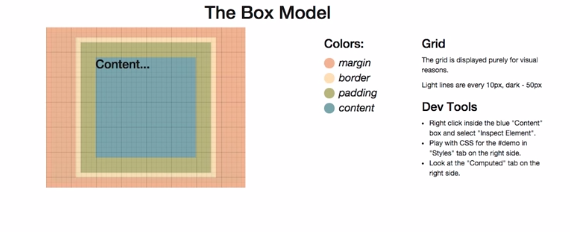
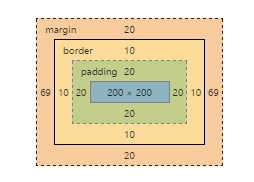
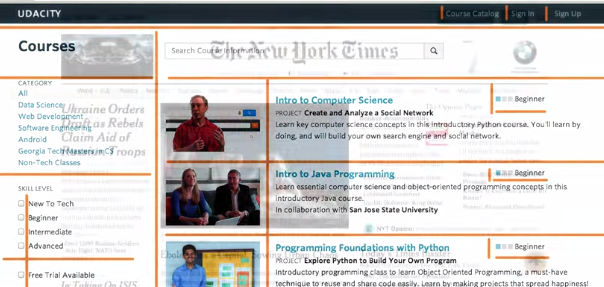
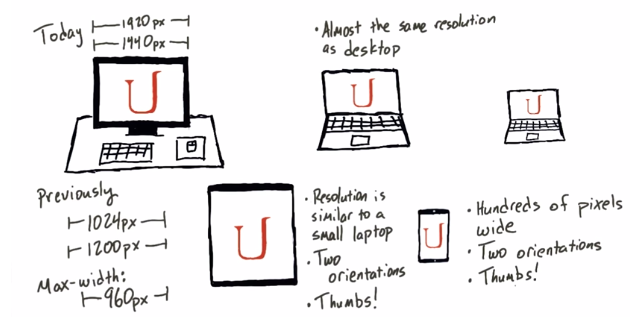
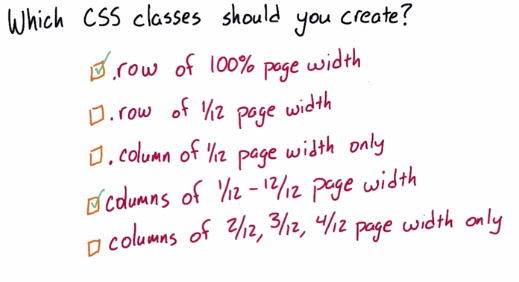
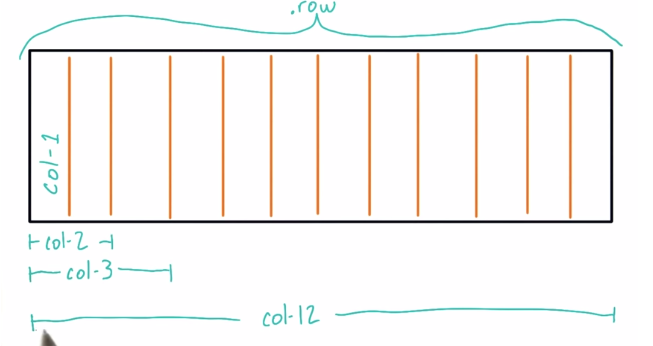
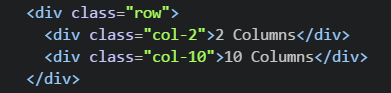
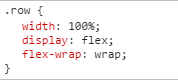
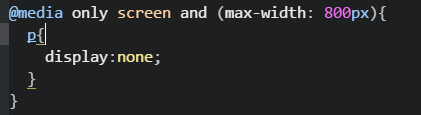
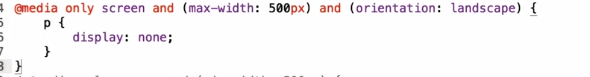
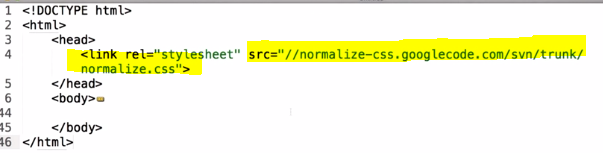
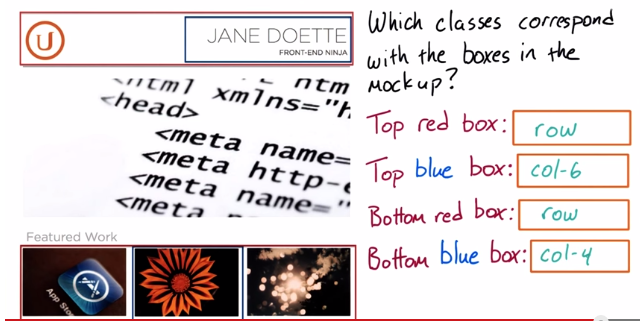
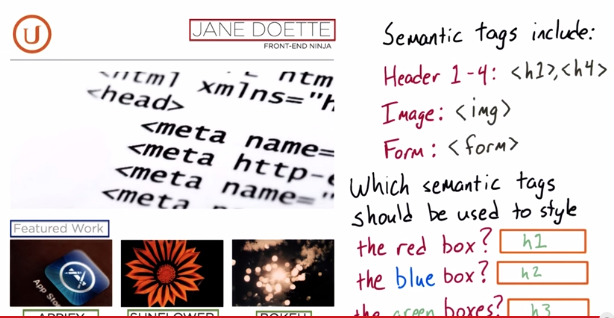
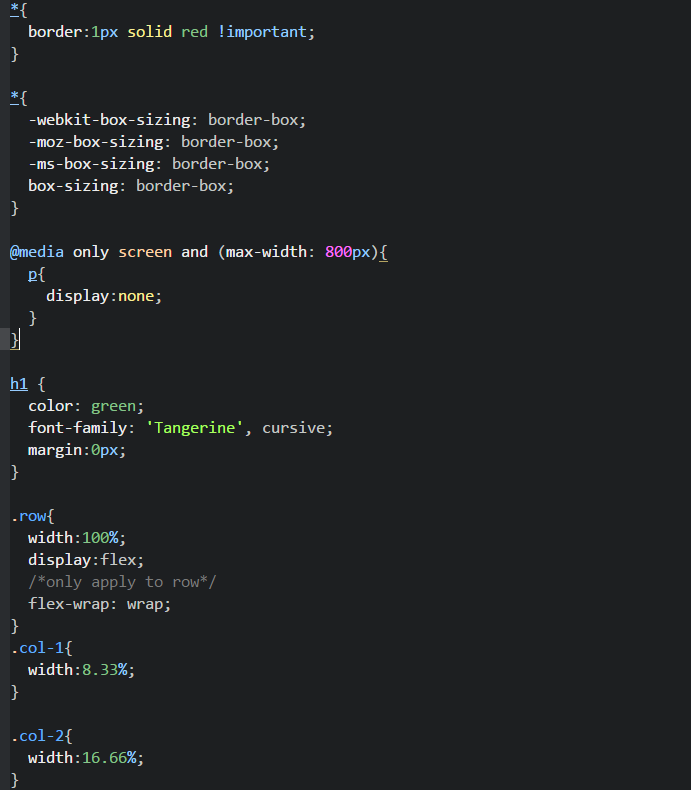
# HTLM, CSS, and Boxes

1. HTML, CSS and Javascript are three fundamental tools of websites
2. Websites - house:
   1. HTML -- structure
   2. CSS –styles
   3. Javascript – interactive components
3. How websites behave:
   1. Web tools
   2. Right click and element🡪 inspect element
4. Google chrome:
   1. Developer tools which shows structure of a page
5. For a webpage viewed using developer tool
   1. Not all elements are visible
   2. All elements are rectangular
   3. You can read the same text as on the page
6. Browse builds a tree-like structure
   1. One example of such tree-like structure:
      1. 
7. How the structure is turned into a page:
   1. Two pages with identical HTML:
      1. 
8. How browsers know what styles to apply and where that style comes from:
   1. HTML – Language of the webs 🡪 syntax + rules
      1. Tag is the basic word 🡪 elements in tree because DOM
      2. Each HTML tag creates an element in the DOM that browsers use to display the page
      3. Everything between and starting tag and ending tage is the content
      4. Each tag has a name which defines the type
         1. div= division = block
         2. 
         3. Class is an attribute. H1 is the name
9. CSS allow you to use specific syntax and rules on the page
   1. For example, you can use CSS to specify that all elements with type h1, and has an attribute class= “article” should use this style:
      1. 
10. Every element in the webpage in the web is a box
    1. Benefits: able to rearrange the webpages
       1. Move the elements in the webpage to fit the need of different devices
11. How to translate a design to HTML:
    1. Give each element a class attribute so you can apply different styles to them
12. CSS:
    1. Resources:
       1. <https://css-tricks.com/how-css-selectors-work/>
       2. https://developer.mozilla.org/en-US/docs/Web/CSS/Reference
    2. Cascaded styling sheet:
    3. You can have several CSS files
    4. With CSS, the most specific rules will be applied to every elements
       1. Find a rule that best describe an element
13. Change div tags to more appropriate types:
    1. First div 🡪 h1
14. The box model:
    1. 
    2. 
15. Make the calculation of box easier:
    1. Browser specific prefixes:
    2. Setting up size of box with percentage of screen size
16. Positioning – one of most complicated
    1. One technique – flexbox
    2. Flexbox: <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>
17. Adding images:
    1. Make image name regular
18. Code, test, refine:
    1. Look for natural boxes
    2. Look for repeated styles& semantics elements
    3. Write your HTML
    4. Apply styles
    5. Fix things
19. W3C CSS validation service;
    1. To verify HTML:
       1. <http://validator.w3.org/#validate_by_input>
    2. To verifcy CSS: <http://jigsaw.w3.org/css-validator/#validate_by_input>

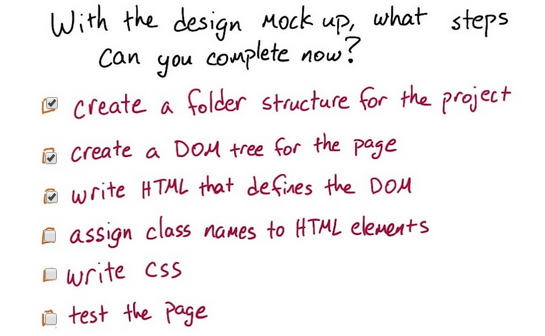
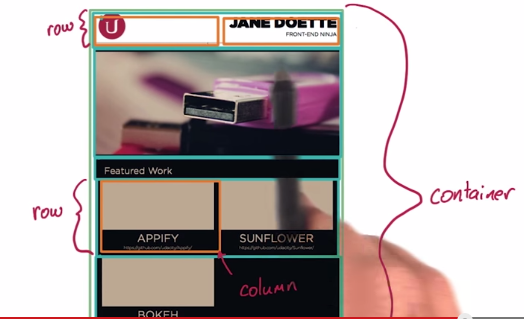
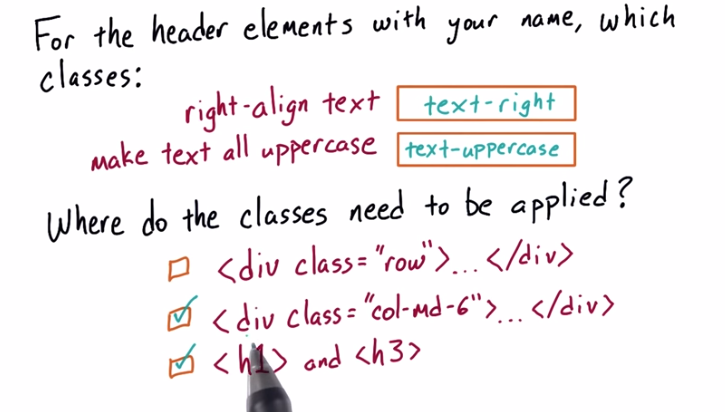
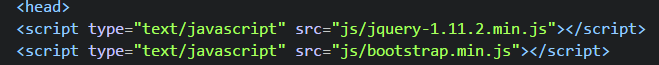
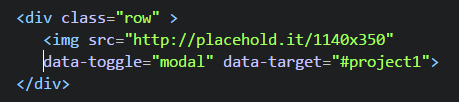
# CSS Framework, Responsive Layouts

1. Responsive design:
   1. Webpages look good for everyone
   2. Grid-based layout:
      1. 
      2. Keep breaking down big grid to smaller grid
2. Where is grid structure come from:
3. Frameworks:
   1. Collections of CSS classes that make page layout easy to implement
4. What is responsive design:
   1. 
5. Implementing responsive design:
   1. Not using fixed pixels
   2. Use percentage
6. Having 12 columns would allow responsiveness for all devices
   1. 2 columns
   2. 3 columns
   3. 4 columns
7. What CSS classes should you create?
   1. 
8. Conventions for naming grid:
   1. 
9. Cascaded style sheet:
   1. 
      1. The .row css will be applied to row level
      2. .col-2 and .col-10 will be applied to col-2 and col-10 respectively
10. Negative space:
    1. Space that does not contain anything
11. Flex:
    1. 
12. Text:
    1. Overflow
13. Media queries:
    1. Change CSS properties depending on device screen size and color
    2. 
    3. 
14. CSS resetting: to make css works the same across different browsers
    1. Use necolas/normalize.css
    2. 
15. Protocol relative URL:
    1. <http://www.paulirish.com/2010/the-protocol-relative-url/>
16. Problem-sovling:
17. Applying the framework:
    1. First identify the structure:
       1. 
    2. Apply semantic tags:
       1. 
    3. Placeholder image:
       1. Pacelhold.it
       2. And more
    4. Continuous improvement:
       1. Fonts
       2. Google.com/fonts
18. Cycle:
    1. Code 🡪 test🡪 refine
    2. Starting out layout
    3. Customize the layout
19. CSS syntax:
    1. Apply to everything:
       1. \*{

}

1. 
   1. Div is used to layout the web
   2. Style is used to have an inline formatting for the element
   3. Real elements are placed inside the div
2. 

# Bootstrap and Other Frameworks

1. 
   1. You CAN do all those things on your own now! You are becoming a front end developer!
   2. But this question was about what you can do when you are going to use an existing framework but you don't yet know which one.
   3. The last three items are not possible until you know which framework you are using. For example, a class like col-12 only has meaning if there is a CSS framework (like the one you made in the last lesson) to interpret it.
2. Frameworks:
   1. Bootstrap: http://getbootstrap.com/
   2. Foundation: http://foundation.zurb.com/
   3. Yaml: http://www.yaml.de/
   4. 960 Grid: http://960.gs/
   5. Suzy: http://susy.oddbird.net/
   6. Frameless: http://framelessgrid.com/
3. Bootstrap is big
   1. For robustness
   2. Allow to pick individual components of bootstrap
   3. 
4. Minify files:
   1. A browser has to download all the CSS and JS file in order to render the web pages
   2. Minify files to be more space efficient
5. Bootstrap grid system:
   1. 
6. Instructions on how to use Developer Tools Shortcuts:
   1. To save the changes you make to the HTML in Developer Tools to your HTML file, right-click or ctrl-click on the <html> element, and select 'Copy as HTML.' Then replace the HTML in your file with the copied text.
7. To make images resize automatically:
   1. Img-responsive
8. Bootstrap documentation:
   1. <http://getbootstrap.com/css/#images>
9. An element can have more than one class associated with it
   1. 
   2. 
10. Custom CSS:
    1. Create a new css file
11. How a real engineer will design a webpage:
    1. Create a structure:
       1. 
12. Add a modal to your project:
    1. Documentation <http://getbootstrap.com/javascript/#modals>
    2. First be sure to download JQuery
       1. Add the scripts file inside the head tag
       2. 
    3. Create the modal:
       1. 
    4. Add the trigger:
       1. 
13. Customized Framework vs Existing framework