Websites

Download zencoding

* + HyperText Markup Language
    - Not a language 🡪 used to write things down
    - What you write down to make a structure
  + CSS – Cascading Style Sheets
    - Also not a language
    - Skin or physical features
    - Style of the internet
    - Defines how html elements looked
      * Color, height, weight
  + Javascript - A programming language, not related to Java
    - Interaction or animation of the internet
    - Makes HTM interact with one another

HTML

* + Boxes within boxes within boxes
  + Written in tags
    - Opening and closing
    - <tagname>………<tagname>
      * stuff within tag is governed by tags
    - Nested tags
      * Abide by stuff in both tagnames
      * Use indentation
      * <tag1>
        + <tag2>

your content

* + - * + <tag2>
      * <tag1>
  + Tags define boxes
  + Tag we use for a generic box is named <div>
    - <div>Content</div>
  + <head> 🡪 nothing ever shows up in web page
    - content shows up in “tab title”
    - Only put in metadata
    - put things in the body if you want to look at it
  + Every page should have:
    - <!DOCTYPE html>
    - <html>
    - <head>
    - <body>
  + Types of Boxes
    - Inputs
      * <input type=”text”>
        + Value of attribute type is text
        + Non-movable input for text
      * You don’t have to close an input tag!
        + No content in it, user puts content in
    - Boxes (viewable)
    - Images
      * Self closing
      * a href
        + Creates a link that will change the page
      * How to open in a new tab
        + Target=”\_blank”
    - Links
* The internet
  + The internet is the roads in a neighborhood
  + Each house is a computer
    - Client = you and your browser
      * Google Chrome
    - Server = machine sending you the data and files you request
      * Google, AOL
    - Host
  + IP address
    - Internet Provider and numbers
    - ###.###.#.#
  + Request-response cycle
    - Client makes a “request”
      * Request == textual message whose syntax and semantics are defined by HTTP
    - Server issues a response
      * Contains status code
    - Cycle repeats itself
    - Another type of protocol: sftp
      * Secure file transfer protocol
* CSS – the skin of the web
* Oopsies
  + Use lower case for everything
    - Tag names, attributes, etc
  + Folder name: hwX
    - X is the number
  + Only use one stylesheet
* Look up “ul” for HTML tags]
  + Answer: Unordered list
* CSS
* Ways to select elements
  + By tag name
    - Just use p, img, or div
    - Rarely used
    - Good to define a global feel
  + By class name
    - **An element can have many classes, just separate by spaces!!**
      * <p class=”blue underline”>
  + By ID:
    - ID is an attribute given to only one element
    - Only one element may have a certain ID
      * Not reusable
    - Pre-fix with a #
    - Fast, but not practical
  + Whole CSS block is a rule
    - Name (e.g. p) is a selector
    - Property (like font size)
    - Value (like 18px)
* Common Styles
  + Color, background-color
  + Text-decoration
    - None, underline
  + Width, height, font-size
    - Percentage, or pixel value
  + Border
    - Size, sytle, color
* The Box Model
  + Content
    - Adjust with width and height
  + Margin
    - Adjust room between boarder and other elements
  + Border
  + Padding
    - Gives blank space between border and content
* Display
  + Display: block

element takes up the full content

* + Display: inline
    - Want to change something within a div box
    - Element takes up only the width of its content
    - Use <span=.somefunclass>text</span> to do so
      * Span by default has the display inline feature
  + Display: inline-block
    - Rendered like an inline element
    - But also allows us to set block-display properties
  + Display: none
    - Element is not rendered in the browser
    - Removed from flow
    - Leaves it in the HTML without affecting anything
    - Used to hide elements that you want to show upon user action
  + Default display properties
    - Inline
      * Span, a, em, most text modifiers
    - Inline block
      * img
    - Block
      * Div, p, ul, form
* Parent/Sibling elements
  + <div id=”grandpa”>
    - <div id=”mom”>
      * <div id=”you”></div>
      * <div id=”brother></div>
    - </div>
  + </div>
* The Position Property
  + Way for an element to position itself with regard to the browser window, the parent element, or relative to the sibling elements
  + Position: static
    - Renders element relative to neighbor
    - Default setting
  + Position: fixed
    - Sets element to be rendered at a fixed location in the browser window, regardless of page-scrolling
    - Top, left, right, bottom tell where in browser window
      * top: 10px;
        + Move 10 down from the top
        + Etc
  + Position: relative
    - Render relative to sibling
  + Positions: absolute
    - Tell element to be rendered at a specific location within first parent element that is not position static
* Homework 1
  + Finish command line tutorial
  + Create a “hw1” git repository on GitHub
  + Upload your assignment to git repository
  + On Lore, submit JUST the link to your “hw1” repository