Selenium

Locators

- The “easy” locators
  - Locate by name
  - Locate by id
  - Locate by class name
  - Locate by link text (special to a tags)
  - Locate by partial link text
  - Locate by tag name (a, body, div, h1)
- The “hard” locators
  - Xpath locator
  - CSS locator

findElement v. findElements

- driver.findElement(By.className(“someClass”));
  - What happens if two or more elements have the class “someClass”?
    - It will only select the very first element that appears in our HTML document
- driver.findElements(By.className(“someClass”));
  - This will return a list containing all elements with the class “someClass”
    - List<WebElement> elements = ….

Absolute Path v. Relative Path (XPath)

- Absolute path starts from the root of the document and traverses all the way to the element we want to grab
  - Ex. /html/body/div[1]/p
    - Select the p tag that is a child of the first div tag that is a child of body which is a child of html
- Relative Path
  - We do not start at the root, but instead just dive straight into potentially nested elements and look for certain things those elements match to
    - //div[1]/p
    - We start with '//' instead of starting at the root and traversing deeper like in the absolute path example

CSS Selectors v. XPath Selectors

- Technically, CSS Selectors are faster than Xpath selectors
  - On the order of a magnitude faster (10x)
  - However, this is pretty negligible simply because of how fast both methods work
  - So in reality, it doesn't really matter which one you use
- Xpath selectors are more versatile
  - Selecting by the text of element isn't really possible with CSS Selectors
    - //*[text()='Some text']
  - Traversing from a child element up to the parent element

Selenium Waits

- We have 3 types of waits:
  - Implicit waits: Implicit waits set a timeout for ALL searches of a WebElement after the point in which the implicit wait was initially configured. It will wait up to a maximum of the specified duration before throwing a NoSuchElementException
  - Explicit waits: Allows us to explicitly wait for a certain element up to the time specified in our WebDriverWait object that we construct. If we exceed this maximum time, it throws a TimeoutException rather than NoSuchElementException as in the case of an implicit wait.
  - Fluent waits
- Really the ones that we probably want to mostly focus on would be the first 2 (Implicit v. Explicit)