* If visited is set to blue and it’s the same blue that corresponds to the null link, then it is not distinguishable.
  + We want to know if a link is distinguishable and in order to do this, we can get the computed style in javascript.
  + We want to figure out how to inject code into the page and figure out what the computed style and if that is different from what we are expecting.
  + Write a selenium extension in order to do this.
* Notes:
  + getComputedStyle works by taking the targeted html tag and from there take the properties set up by the css sheets
  + It looks like some websites actually disable the link feature
    - The visited feature is kept, but the color for the link is disabled so there’s no initial color to be set at all.
    - TODO: Figure how to distinguish between the different a tags
      * It looks like there might be differences in what class is being used, but that varies per website. Not exactly a good indicator
      * What could be a good indicator is to remove all the classes associated with any a tag. This might help since instead of focusing on a certain link designated to a class, it would cover all the links.
  + It looks like there’s usually one main CSS file sheet meaning there should only be one main a:visited link and one main a:link selector
    - getComputedStyle should be able to get the main color that is revealed
  + There are some limits to the visited link styles
    - Only attributes that can be used are
      * Color
      * Background-color
      * Border-color
      * Column-rule-color
      * Outline-color
      * Color parts of the fill and stroke attributes
    - This confirms that our only focus will be the color attributes from the CSS files
  + The :link and :visited selectors are extensions to the real tag (which is usually a, which is the HTML tag for a hyperlink). These are known as psuedoelements
    - The default values are:
      * An unvisited link is underlined and blue
      * A visited link is underlined and purple
      * An active link is underlined and red
  + Example getComputedStyle:

let para = document.querySelector('p'); # used to get the html tag

let compStyles = window.getComputedStyle(para); # used to get the computed style from the html tag

para.textContent = 'My computed font-size is ' +

compStyles.getPropertyValue('font-size') +

',\nand my computed line-height is ' +

compStyles.getPropertyValue('line-height') +

'.'; # example showcase on how to extract the elements

* + Other example

var a = document.querySelector('a');

var result = getComputedStyle(a, ':link').content;

* + - This might be helpful with figuring out how to set up to focus on the :link and :visited selectors