Basic Locators

Locator Type	**Syntax**	**Example**	**Selenium syntax Example**
Xpath	//tagname[attribute=value]	"//input[@name='email']"	driver.find_element_by_xpath("//input[@name='email']")
Css selector	tagname[attribute=value]	"input[name='email']"	driver.find_element_by_css_selector("input[name='email']")
ID	No syntax	"id"	driver.find_element_by_id("exampleFormControlSelect1")
Name	No syntax	"name"	driver.find_element_by_name("name")
Class name	No syntax	"class-name"	driver.find_element_by_class_name("btn-success")
Link text	No syntax	"link-text"	driver.find_element_by_link_text("Genealogies")
Partial link text	No syntax	"partialtext"	driver.find_element_by_partial_link_text("partialtext")
Tag name	No syntax	"tag-name"	driver.find_element_by_tag_name("span")

Customized XPATH Locator Without Tag Name

Replace tag name with asterisk (*) sign
Syntax: //*[attribute=value]
Example: //*[@name = ‘email’]

Customized CSS Selector Syntax Without Tag Name

Remove tag name
Syntax: [attribute=value]
Example: [class*=‘alert-succes’]

Generating XPATH based on text

Syntax: //tagname[contains(text(), ‘actual-text’)]
Example: //span[contains(text(),'Users Info')]

Creating XPATH by traversing tags

Syntax: ParentTag/ChildTag
Example: //div[@class='product-action']/button

Creating a CSS Selector by traversing to nth child

Syntax: Tagname:nth-child(x)
Example: div:nth-child(1)

Select Parent Locator from Child using XPATH

Syntax XPATH/parent::tagname
Example: //*[title="test"]/parent::div

Generating CSS Selector from Tag and Class Name

Replace spaces with period(.) to use more than one class name
Syntax: Tagname.ClassName
Example: input.search-keyword