Для того чтобы что-то сделать с веб элементом, надо его сначала найти. Здесь элемент имеется в виду HTML элемент на странице. У каждого элемента есть название и свои атрибуты, но иногда они могут повторятся. Для того чтобы работать с элементом надо его уникально идентифицировать.

**Походу WebDriver умеет работать не со всеми локаторами, хотя так-как можнопользоватся x-path то это не важно.**

Для этого есть локаторы (или Locator stratgies), это способы используя их можно определить/найти вэб элемент на странице.

**driver.findElement(By.locator)**

**Locator strategies**

Locator strategies - Ways to identify one or more specific elements in the DOM.

A locator is a way to identify elements on a page. It is the argument passed to the Finding element methods

**Selenium provides support for 8 traditional location strategies in WebDriver:**

- class name - Locates elements whose class name contains the search value (compound class names are not permitted)

- css selector - Locates elements matching a CSS selector

- id - Locates elements whose ID attribute matches the search value

- name - Locates elements whose NAME attribute matches the search value

- link text- Locates anchor elements whose visible text matches the search value

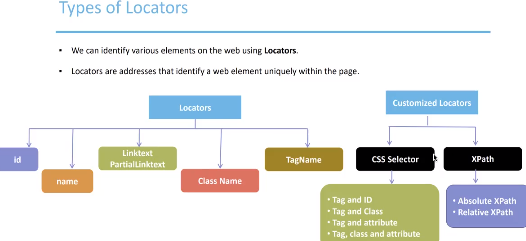
- partial link text -Locates anchor elements whose visible text contains the search value. If multiple elements are matching, only the first one will be selected.

- tag name- Locates elements whose tag name matches the search value

-xpath- Locates elements matching an XPath expression

In general, if HTML **ID**s are available, unique, and consistently predictable, *they are the preferred method* for locating an element on a page. They tend to work very quickly, and forego much processing that comes with complicated DOM traversals. If unique IDs are unavailable, a well-written **CSS selector** *is the preferred* method of locating an element. **XPath** works as well as CSS selectors, but the syntax is complicated and frequently difficult to debug. Though XPath selectors are very flexible, they are typically not performance tested by browser vendors and tend to be quite slow.

The recommendation is to keep your locators as compact and readable as possible. Asking WebDriver to traverse the DOM structure is an expensive operation, and the more you can narrow the scope of your search, the better.



Normal locators (left part of the picture) we can see on a web page, they are all the attributes of web elements. We can capture them from a web tool.

Customize locators - we use when we can not reach the element, for example if it has not attributes known to Selenium.