

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.

Notice! That all the normal locators (based on a tag attributes) are supported by Selenium, and we do not need to specify the name of the HTML teg, only focus on an attribute.

**driver.findElement(By.id(“id\_name”));**

**driver.findElement(By.name(“name\_of\_the\_tag”));**

**driver.findElement(By.linkText(“Google”)).click();** полное название текста ссылки (**не сама ссылка**)

**driver.findElement(By.partialLinkText(“Goo”)).click();<--частичное название текста**

The command finds the element using link text (здесь надо понимать что речь идёт не о самой ссылке а об названии, т.е.не url a то как она существует на HTML странице) and then click on that element and thus the user would be redirected to the corresponding page. Используя **partialLinkText** помни что чястичное название ссылки может встречатся и в других элементах, что может выдать ошибку.

**driver.findElement(By.className(‘some\_class\_name’)); ←** ClassName is good to use for finding multiple elements because this attribute is the same for big amount of elements (in that case use driver.findElements()). **Important note** if attribute class has value then space and some value again it means that these element use multiple class value, in that case you my choose the value that fits you better.

For example we need to find all images elements on the page. These images has different class value, and different attributes. How to find them? Every image has a similar tag name and that is **img**. So we can use the **tagName** locator.

**List <WebElement> my\_images = driver.findElements(By.tagName(“img”));**