

## Métodos get()

- [get\(\)](#)
- [getClass\(\)](#)
- [getCurrentUrl\(\)](#)
- [getWindowHandles\(\)](#)

### get()

- El comando inicia un nuevo navegador y abre la URL especificada en la instancia del navegador
- El comando toma un único parámetro de tipo de cadena que suele ser una URL de la aplicación bajo prueba
- Para los usuarios de Selenium IDE, el comando puede parecerse mucho a un comando abierto.

```
driver.get("https://google.com");
```

### getClass()

- El comando se usa para recuperar el objeto Class que representa la clase de tiempo de ejecución de este objeto

```
driver.getClass();
```

### getCurrentUrl()

- El comando se usa para recuperar la URL de la página web a la que el usuario está accediendo actualmente.
- El comando no requiere ningún parámetro y devuelve un valor de cadena

```
driver.getCurrentUrl();
```

### getPageSource()

- El comando se usa para recuperar la fuente de la página web a la que el usuario está accediendo actualmente
- El comando no requiere ningún parámetro y devuelve un valor de cadena
- El comando se puede usar con varias operaciones de cadena como `contains()` para determinar la presencia del valor de cadena especificado

```
boolean result = driver.getPageSource().contains("String to find");
```

## getTitle()

- El comando se utiliza para recuperar el título de la página web en la que el usuario está trabajando actualmente. Se devuelve una cadena nula si la página web no tiene título
- El comando no requiere ningún parámetro y devuelve un valor de cadena recortado

```
String title = driver.getTitle();
```

## getText()

- El comando se usa para recuperar el texto interno del elemento web especificado
- El comando no requiere ningún parámetro y devuelve un valor de cadena
- También es uno de los comandos más utilizados para verificar mensajes, etiquetas, errores, etc. que se muestran en las paginas web

```
String Text = driver.findElement(By.id("Text")).getText();
```

## getAttribute()

- El comando se utiliza para recuperar el valor del atributo especificado.
- El comando requiere un único parámetro de cadena que hace referencia a un atributo cuyo valor aspiramos a conocer y devuelve un valor de cadena como resultado.

```
driver.findElement(By.id("findID")).getAttribute("value");
```

## getWindowHandle()

- El comando se utiliza para abordar la situación cuando tenemos más de una ventana para tratar.
- El comando nos ayuda a cambiar a la ventana recién abierta y realiza acciones en la nueva ventana.
- El usuario también puede volver a la ventana anterior si lo desea.

```
private String winHandleBefore;  
winHandleBefore = driver.getWindowHandle();  
driver.switchTo().window(winHandleBefore);
```

## getWindowHandles()

- El comando es similar al de "getWindowHandle()" con la sutil diferencia de que ayuda a manejar múltiples ventanas, es decir, cuando tenemos que manejar más de 2 ventanas.