

# Selenium WebDriver

Selenium WebDriver è una libreria che fornisce un'interfaccia di programmazione per l'automazione di test su applicazioni web. Supporta diversi linguaggi di programmazione come Java, Python, C#, ecc. Ecco alcuni dei metodi principali forniti dalla classe WebDriver di Selenium:

## 1. `get(String url)` :

- Carica una nuova pagina web nella finestra corrente del browser.

```
WebDriver driver = new ChromeDriver();  
driver.get("https://www.example.com");
```

## 2. `getCurrentUrl()` :

- Restituisce l'URL della pagina corrente.

```
String currentUrl = driver.getCurrentUrl();
```

### 3. `getTitle()` :

- Restituisce il titolo della pagina corrente.

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

#### 4. `findElement(By by)` :

- Trova il primo elemento che corrisponde al selettore specificato.

```
WebElement element = driver.findElement(By.id("elementId"));
```

## 5. `findElements(By by)` :

- Trova tutti gli elementi che corrispondono al selettore specificato.

```
List<WebElement> elements = driver.findElements(By.className("exampleClass"));
```

## 6. `sendKeys(CharSequence... keysToSend)` :

- Invia una sequenza di tasti a un elemento (ad esempio, un campo di input).

```
WebElement inputField = driver.findElement(By.name("username"));  
inputField.sendKeys("username123");
```

## 7. `click()` :

- Esegue un clic sull'elemento corrente.

```
WebElement button = driver.findElement(By.id("submitButton"));  
button.click();
```



## 8. `clear()` :

- Cancella il contenuto di un campo di input.

```
WebElement inputField = driver.findElement(By.name("password"));  
inputField.clear();
```

## 9. `getAttribute(String name)` :

- Restituisce il valore di un attributo dell'elemento corrente.

```
WebElement link = driver.findElement(By.linkText("Example Link"));  
String hrefValue = link.getAttribute("href");
```

## 10. `getText()` :

- Restituisce il testo visibile sull'elemento corrente.

```
WebElement heading = driver.findElement(By.tagName("h1"));  
String headingText = heading.getText();
```

## 11. `isDisplayed()` :

- Verifica se l'elemento corrente è visualizzato sulla pagina.

```
WebElement element = driver.findElement(By.id("exampleElement"));  
boolean isDisplayed = element.isDisplayed();
```

## 12. isEnabled() :

- Verifica se l'elemento corrente è abilitato o disabilitato.

```
WebElement button = driver.findElement(By.id("submitButton"));  
boolean isEnabled = button.isEnabled();
```

### 13. `isSelected()` :

- Verifica se l'elemento corrente è selezionato (ad esempio, una casella di controllo o un pulsante radio).

```
WebElement checkbox = driver.findElement(By.id("agreeCheckbox"));  
boolean isSelected = checkbox.isSelected();
```

14. `navigate().to(String url)` :

- Naviga verso una nuova pagina web specificata dall'URL.

```
driver.navigate().to("https://www.example.com");
```

## 15. `navigate().back()` :

- Torna alla pagina precedente nella cronologia del browser.

```
driver.navigate().back();
```



## 16. `navigate().forward()` :

- Vai alla pagina successiva nella cronologia del browser.

```
driver.navigate().forward();
```

## 17. `quit()` :

- Chiude tutte le finestre del browser aperte e termina la sessione WebDriver.

```
driver.quit();
```

Questi sono solo alcuni dei metodi principali forniti da Selenium WebDriver. Puoi trovare molti altri metodi e funzionalità avanzate nella documentazione ufficiale di Selenium: [Selenium WebDriver Documentation](#).