

Sélenium:


Aide-mémoire des
commandes
couramment utilisées

– Paule Celestine Nguessan, Testeur Automaticien

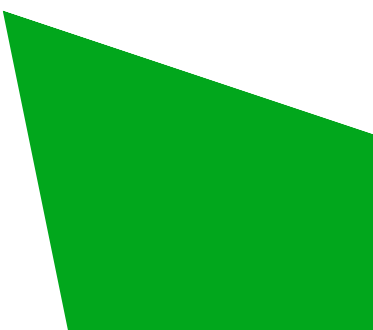


Se

Selenium



Bonjour voici un aide-mémoire des commandes couramment utilisées dans Sélénium, un outil de test automatisé pour les applications web. Ces commandes sont généralement utilisées avec des langages de programmation tels que Python, Java, C#, etc. pour automatiser les tests sur des navigateurs web.



Initialisation et configuration :

1. Importation des modules :

- Python : `from selenium import webdriver`
- Java : `import org.openqa.selenium.WebDriver;`

2. Initialisation du pilote :

- Python : `driver = webdriver.Chrome()` (pour Chrome)
- Java : `WebDriver driver = new ChromeDriver();`

3. Ouverture d'une URL :

- `driver.get("https://www.example.com");`

Recherche d'éléments :

4. Recherche par ID :

- `driver.findElement(By.id("elementId"));`

5. Recherche par nom :

- `driver.findElement(By.name("elementName"));`

6. Recherche par classe :

- `driver.findElement(By.className("className"));`

7. Recherche par balise HTML :

- `driver.findElement(By.tagName("tag"));`
- 

8. Recherche par sélecteur CSS :

- `driver.findElement(By.cssSelector("cssSelector"));``

9. Recherche par XPath :

- `driver.findElement(By.xpath("xpathExpression"));``

Actions sur les éléments :

10. Clique sur un élément :

- `element.click();``

11. Saisie de texte dans un champ :

- `element.sendKeys("text");``

12. Obtention du texte d'un élément :

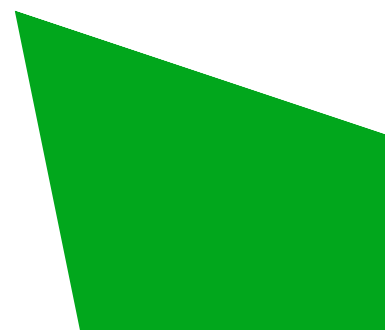
- `element.getText();``

13. Obtention de la valeur d'un attribut :

- `element.getAttribute("attributeName");``

14. Soumission d'un formulaire :

- `element.submit();``



Attentes :

15. Attente explicite :

- `WebDriverWait wait = new WebDriverWait(driver, timeout);`
- `wait.until(ExpectedConditions.someExpectedCondition);`

16. Attente implicite :

- `driver.manage().timeouts().implicitlyWait(time, TimeUnit.SECONDS);`

Navigation :

17. Aller en arrière :

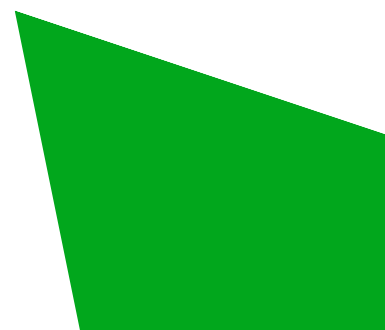
- `driver.navigate().back();`

18. Aller en avant :

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

19. Actualiser la page :

- `driver.navigate().refresh();`



Manipulation de fenêtres et d'onglets :

20. Basculer vers une fenêtre/onglet spécifique :

- `driver.switchTo().window("windowHandle");``

21. Basculer vers un cadre (iframe) :

- `driver.switchTo().frame("frameNameOrId");``

Actions souris :

22. Double-clic :

- `new Actions(driver).doubleClick(element).perform();``

23. Survoler un élément :

- `new`

`Actions(driver).moveToElement(element).perform();``

Gestion des cookies :

24. Ajouter un cookie :

- `driver.manage().addCookie(cookie);``

25. Obtenir les cookies :

- `driver.manage().getCookies();``

26. Supprimer tous les cookies :

- `driver.manage().deleteAllCookies();``

Fermeture et quitter :

27. Fermer la fenêtre actuelle :

- `driver.close();`

28. Quitter le navigateur :

- `driver.quit();`

Ceci n'est qu'un aperçu des commandes de base de Sélénium. Les fonctionnalités complètes sont plus étendues et dépendent du langage de programmation utilisé. Assurez-vous de consulter la documentation officielle de Sélénium (<https://www.selenium.dev/documentation/>) pour plus de détails.