

Nom : Le Franc

Prénom : Matthieu

N°étudiant : 71800858

Email : matthieu.le-franc@etu.u-paris.fr

TP - micro projet Selenium

Partie 1 : recherche prénom + nom

tests_assert.side

J'ai pu aisément effectuer ce premier exercice sans blocages, peu importe la vitesse du test. Un éventuel blocage arrive peut être si le chargement de la page est trop long ? Dû à un problème quelconque entrainant un temps de réponse jugé trop important par sélénium ?

Partie 2 : recherche avec assertion

tests_assert.side

Partant du moteur de recherche "duckduckgo" j'ai ajouté une assertion pour vérifier la présence d'un élément. On vérifie que le bouton de lancement de la recherche dans la page d'accueil est présent après l'ouverture de la page avec :

```
assert element present
```

Le résultat de cette commande est toujours **OK** car le logo de la page d'accueil est toujours présent.

Partie 3 : vérification de la présence d'un élément

tests_assert.side

Ici on veut s'assurer si en effectuant une recherche avec 2 mots, un troisième mot est présent. Pour cela on va récupérer la **value** de la **target** considéré, après l'avoir rempli, grâce à la commande :

```
store value
```

Cette valeur est stockée dans la valeur **len** que l'on va passer à la commande :

```
execute script
```

Ce mini script va retourner le nombre de mots de la recherche dans la **value len** que l'on passe enfin en **target** de la commande :

```
assert
```

La **value** que l'on donne à cette assertion est 3, car on veut vérifier la présence de 3 mots dans la recherche. Le retour de cette commande est **Failed: Actual value '2' did not match '3'** car len vaut 2, ce qui signifie que la recherche ne contient que 2 mots et pas 3.

J'ai utilisé un petit script pour compter le nombre de mots, avant de réaliser l'assertion, car je ne suis pas parvenu à utiliser correctement une expression régulière dans selenium IDE.

Partie 4 : comportement unique

test_UIO

Ce test se trouve dans un autre projet selenium car il semble que la convention soit : un url = un projet selenium.

Nous nous rendons à cette url : <https://www.apple.com/go/awards/index.html> avec la commande

```
open
```

J'ai trouvé cette page en cherchant les pages du site www.apple.com n'étant pas censées être indexées. On peut visualiser cette liste grâce à la ressource de format texte placée généralement à la racine des sites web, ici : <https://www.apple.com/robots.txt>

La route /go/awards/index.html n'est pas indexée, elle n'est donc pas censée être accessible par un utilisateur depuis la racine du site.

Après avoir redimensionné la page, le **préambule** du test est passé.

On va maintenant cliquer sur le lien *Get a Mac / Viruses* qui nous redirige vers la page : https://www.apple.com/go/awards/getamac_viruses/index.html Cette page, pas indexée non plus, est censée être accessible uniquement depuis la page précédente. On a donc notre **transition**.

Pour la **sortie** du test, on veut s'assurer que l'on est bien au point d'arrivée escompté. On va pour cela vérifier la présence d'un élément et de son contenu, ici le titre de la page, avec la commande :

