Avec selenium, pour remplir un élément il faut faire : * 1 point

- ○ element.fill()
- ○ element.send_fill()
- ○ element.keys()
- ◉ element.send_keys()
- ○ element.make_fill()

Avec selenium, pour cliquer sur un élément il faut faire : * 1 point

- ○ element.make_click()
- ○ element.get_click()
- ◉ element.click()
- ○ element.set_click()
- ○ element.send_click()

Avec sélénium, on peut sélectionner un élément par (cocher toutes les bonnes réponses) : * 1 point

- ☑ css
- ☑ id
- ☐ div
- ☑ xpath
- ☐ span

Le temps d'attente pour le chargement de la page est important quand on interagit avec le navigateur automatique. La librairie python qui permet d'imposer un temps d'attente est : * 1 point

- ( ) date
- ( ) datetime
- (•) time
- ( ) timestamp
- ( ) sleep

** **

En se basant sur le code HTML ci-dessus, quel(s) bout(s) de code retourneraie(nt) la valeur "16/01/2020" avec Selenium ? * 1 point

- [ ] driver.find_element_by_css_selector("div>span").get_attribute("data")
- [x] driver.find_element_by_css_selector("div>span[data]").get_attribute("data")
- [ ] driver.find_element_by_css_selector("div>span[type$='avec']").get_attribute("data")
- [x] driver.find_element_by_css_selector("div>span[type^='avec']").get_attribute("data")
- [ ] driver.find_element_by_css_selector("div>span[type*='data']").get_attribute("data")

Pour extraire le code source d'une page on peut utiliser (cocher toutes les bonnes réponses) : * 1 point

- [ ] driver.source_page
- [ ] driver.find_element_by_tag_name("html").get_attribute("html")
- [x] driver.find_element_by_tag_name("html").get_attribute("outerHTML")
- [x] driver.page_source
- [ ] driver.find_element_by_css_selector("html").get_attribute("html")