selenium

2021年8月2日　6:17

ライブラリをインポート

```
from selenium import webdriver
import chromedriver_binary
from selenium.webdriver.chrome.options import Options
from bs4 import BeautifulSoup
```

メインプログラム

```
#webdriverを定義
driver = webdriver.Chrome()
#更新を最大10秒待つ
driver.implicitly_wait(10)
#URLを指定して起動
driver.get("URL名")

#要素を探す
driver.find_element_by_id("idの名前").send_keys("入力したい値")
driver.find_element_by_class_name("クラスの名前").click()

#seleniumを終了する
driver.close()
```

メソッド	説明
find_element_by_id(id)	id属性で要素を検索する
find_element_by_name(name)	name属性で要素を検索する
find_element_by_class_name(name)	class属性で要素を検索する
find_element_by_tag_name(name)	タグ名で要素を検索する
find_element_by_xpath(xpath)	XPathで要素を検索する
find_element_by_css_selector(css_selector)	CSSセレクタで要素を検索する
find_element_by_link_text(link_text)	リンクテキストで要素を検索する
find_element_by_partial_link_text(link_text)	リンクテキストの部分一致で要素を検索する

メセッド名	説明
send_keys	その要素に対して値を入力する
click	その要素をクリックする
clear	textを削除する
text	textを取得する

★ヘッドレスモードで起動する

```
options = Options()
options.add_argument('--headless')
```

★現在のページのURLを取得

```
url = driver.current_url
```

★現在のページのhtmlソースを取得

```
driver.page_source
```

Beautifulsoup

```
#BeautifulSoupを定義
soup = BeautifulSoup("htmlファイル名","html.parser")

#要素を抽出
soup.find_all("タグ名")
```

パーサー	引数での指定方法	特徴
Python’s html.parser	“html.parser”	追加ライブラリが不要
lxml’s HTML parser	“lxml”	高速に処理可
lxml’s XML parser	“xml”	XMLに対応し、高速に処理可
html5lib	“html5lib”	正しくHTML5を処理可

pandasでtable要素をDataflameで取得する

```
df = pd.read_html("htmlソース",encoding="UTF-8")
```

※同ページ内にtable要素が複数ある場合はDataflameがリスト形式で出力される
　為下記の様に記載して表示する

```
df = df[1]
```