

Download HTML Page Content

At step 2, we will download HTML page content with selenium package. And try to control Chrome web browser in two mode normal mode and headness mode.

Installation

First of all we need to install selenium package in order to control browser with python. Let open command prompt on Windows (terminal in Mac) and typing in.

```
pip install selenium
```

To control Chrome, We also need to download [Chrome driver](#) . Let choose stable version and place some where in your local.

All versions available in Downloads

- Latest stable release: [ChromeDriver 75.0.3770.140](#)
- Latest beta release: [ChromeDriver 76.0.3809.68](#)



Control browser in normal mode

We will use `webdriver` to control Chrome browser, so the first step is import `webdriver` from `selenium` package.

```
from selenium import webdriver
```

Then we want to create `Chrome` object, note that we need to specify absolute path where we place downloaded `chrome driver` above.

```
driver_path = r'C:\chromedriver_win32\chromedriver.exe'  
driver = webdriver.Chrome(executable_path=driver_path)
```

Now with `driver` object, we could download HTML content with `get` function. Support we want to download HTML from <https://www.nba.com/>. You could see that a Chrome browser instance is created then NBA home page is loaded.

```
driver.get('https://www.nba.com/')
```

Now to access HTML content, we just need to use property `page_source`

```
print(driver.page_source)
```

At here we complete download HTML content and print it out, at final step we should `close` driver so you will see Chrome driver close.

```
driver.close()
```

That it, let summary every thing again

```
# import webdriver
from selenium import webdriver

# create Chrome object
driver_path = r'C:\chromedriver_win32\chromedriver.exe'
driver = webdriver.Chrome(executable_path=driver_path)

# access url
driver.get('https://www.nba.com/')

# access HTML content
print(driver.page_source)

# close browser
driver.close()
```

Control browser in headness mode

Headness mode mean we still create a Chrome browser instance to download HTML but Chrome browser **do not show up**.

Why we do not want browser to show up ?

The answer is in `headness` mode every thing will faster and our scraping script consume less cpu, ram resource. So after we build up script completely, we should running it in `headness` mode. We do normal mode when we want to debug and develop script.

To run brower in `headness` mode, the first step we need to create `Chromeoptions` object and set value to `headless` property.

```
# create option for headness mode
options = webdriver.ChromeOptions()
options.headless = True
```

Now when we create `Chrome` object, we need put in `option` parameter.

```
# create Chrome object with option
driver_path = r'C:\chromedriver_win32\chromedriver.exe'
driver = webdriver.Chrome(executable_path=driver_path,
                          options=options)
```

That it, let summary every thing again

```
# import webdriver
from selenium import webdriver

# create option for headness mode
options = webdriver.ChromeOptions()
options.headless = True
```

```
# create Chrome object
driver_path = r'C:\chromedriver_win32\chromedriver.exe'
driver = webdriver.Chrome(executable_path=driver_path,
                           options=options)

# access url
driver.get('https://www.nba.com/')

# access HTML content
print(driver.page_source)

# close browser
driver.close()
```