# Selenium

Selenium è un framework di test sviluppato in Java che permette di interagire con diversi browser Web. In effetti permette di realizzare dei test dinamici che facilitano dei test funzionali e quindi non richiedono l’esecuzione di software. Per realizzare i nostri test abbiamo utilizzato l’estensione che si chiama Selenium IDE che può essere installata su qualsiasi browser.

## Librerie o package usate pero lo scrapper

Re: libreria per le espressioni regolari

Time: libreria che contiene dei metodi per il tempo.

Datetime: libreria che contiene dei metodi per le date.

Selenium: Package di selenium che contiene dverse librerie.

Webdriver: Libreria che permette di determinare il browser da utilizzare per realizzare i test.

Options: Libreria che contiene diverse opzioni per webdriver.

Bs4: Package che la libreria beautiful soup che permette di parsare un codice HTML.

Beautiful soup: Libreria che permette di parsare del codice HTML.

## Estrapolazione dati (Scrapper)

Per estrapolare i dati dal sito abbiamo realizzato uno script. Per realizzarlo quest’ultimo ci siamo basati su quello del professore Petrini modificando alcune cose. Il browser che utilizziamo per eseguire i test è chrome, quindi nello script abbiamo dovuto specificare quale driver utilizzarlo. Per farlo abbiamo solo dovuto scrivere questa riga di codice:

self.browser = webdriver.Chrome()

Ogni volta che testiamo questo scrapper apre il browser Google Chrome andando sul sito dell'orario della scuola ed inseguito esegue i vari test. Questo possiamo solo lasciarlo in fase di sviluppo, ma certamente per il sistema finale la "GUI" dobbiamo nasconderla, per farlo dobbiamo aggiungere le options come parametro in più nel webdriver.Chrome( ).

opts = Options()

opts.set\_headless()

assert opts.headless

self.browser = webdriver.Chrome(options=opts)

Per aprire raggiungere un sito web dobbiamo fare una richiesta GET con webdriver di selenium.

self.browser.get(self.url)

Questo comando permette di verificare che un determinato elemento sia caricato. All’interno del metodo loaded\_check bisogna mettere l’id del elemento che disederiamo caricare. In questo caso l’id corrisponde al link interno “corsi”.

self.loaded\_check("GInterface.Instances[0].Instances[0]\_Combo1")

Per potere ricavare un elemento HTML con selenium bisogna utilizzare il metodo: find\_element\_by\_id() 🡺 permette di trovare un elemento con un determinato id.

Per cliccare diversi elementi di un sito con selenium bisogna utilizzare il metodo click().

Ad esempio, qui permette di cliccare il link interno “corsi”.

self.browser.find\_element\_by\_id("GInterface.Instances[0].Instances[0]\_Combo2").click()

Per potere scrivere il nome della classe all’interno della barra della ricerca abbiamo utilizzato il seguente codice:

Permette di selezionare la barra di ricerca che permette di cercare l’orario di una classe.

cerca\_orario = self.browser.find\_element\_by\_id("GInterface.Instances[1].Instances[1].bouton\_Edit")

Il metodo clear() permette di cancellare il contenuto che c’è all’interno della barra di ricerca (magari in precedenti ricerca nella barra di ricerca c’è scritto qualcosa che dobbiamo per forza cancellare).

cerca\_orario.clear()

Il metodo send\_keys() permette di scrivere all’interno della barra di ricerca.

cerca\_orario.send\_keys(ricerca)

Permette di cliccare la “lente” che avvia la ricerca.

self.browser.find\_element\_by\_id("GInterface.Instances[1].Instances[1].bouton\_Bouton").click()