

# WEB SCRAPING CON PYTHON

LEONARDO TAMIANO

Created: 2023-12-31 Sun 18:38

# TABLE OF CONTENTS

- What is Web Scraping?
- BeautifulSoup
- Real (Life) Example

# WHAT IS WEB SCRAPING?

Per "**web scraping**" si intende l'utilizzo di una serie di tecnologie al fine di estrarre dati dal web (tipicamente da pagine **HTML**) in modo da poterli poi processare come si vuole.

Un tipico esempio di web scraping consiste nell'estrarre dei dati di interesse da una pagina web per poi metterli in un'altra pagina web, andando però a cambiare lo stile utilizzato per mostrare i dati.

# DOCUMENT OBJECT MODEL

I file scritti in **HTML** possono essere rappresentati tramite una struttura dati chiamata **Document Object Model (DOM)**. Il DOM è una struttura albero che contiene sia la struttura del documento e sia il contenuto del documento.

```
<!DOCTYPE html>
<head>
  <title> Titolo Pagina </title>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
</head>

<body>
  <div id="content">

    <h1> Headline level 1 </h1>
    <p> This is a paragraph! </p>

    <div id="footer">
      <p class="author"><b>Author</b>: Leonardo Tamiano</p>
    </div>
  </div>
```

Esempio file HTML

```
└─ DOCTYPE: html
  └─ HTML
    └─ HEAD
      └─ #text:
        └─ TITLE
          └─ #text: Titolo Pagina
        └─ #text:
      └─ #text:
    └─ BODY
      └─ #text:
      └─ DIV id="content"
        └─ #text:
        └─ H1
          └─ #text: Headline level 1
        └─ #text:
        └─ P
          └─ #text: This is a paragraph!
        └─ #text:
        └─ DIV id="footer"
          └─ #text:
          └─ P class="author"
            └─ B
              └─ #text: Author
              └─ #text: : Leonardo Tamiano
            └─ #text:
          └─ #text:
        └─ #text:
```

Esempio DOM generato con [live-dom-viewer](#)





Esempio DOM generato con **Graphviz**

Tipicamente le librerie utilizzate per fare web scraping funzionano in due passi:

1. Si costruiscono il DOM rappresentante il documento da analizzare.
2. Offrono una serie di APIs per muoversi all'interno del DOM e raccogliere solamente i dati di nostro interesse.

**BEAUTIFUL SOUP**

**Beautiful Soup** è una libreria Python che ci permette di fare web scraping.

*Beautiful Soup is a Python library for pulling data out of HTML and XML files. It works with your favorite parser to provide idiomatic ways of navigating, searching, and modifying the parse tree. It commonly saves programmers hours or days of work.*

Source: [Beautiful Soup](#)

# INSTALLATION

Per quanto riguarda l'installazione possiamo utilizzare **pip**, il packat manager per python.

```
pip install beautifulsoup4
```

# GENERATING THE DOM

Supponiamo di aver salvato il codice html fatto vedere prima nel file **web**<sub>pageexample.html</sub>. Per generare il DOM possiamo procedere come segue

```
#!/usr/bin/env python3

from bs4 import BeautifulSoup

# -- read file
f = open("./web_page_example.html", "r")
text = f.read()

# -- generate DOM structure
soup = BeautifulSoup(text, 'html.parser')
```



# NAVIGATING THE DOM

Una volta che abbiamo generato la struttura DOM la possiamo navigare in vari modi:

- Trovare tutti i tags di un certo tipo

```
# -- find all tags of the form <p> ... </p>  
paragraphs = soup.find_all("p")
```

- Trovare tutti i tags con un certo attributo

```
# -- find all tags of the form <div id="footer"> ... </div>  
footer_div = soup.find("div", {"id": "footer"})
```

- A partire da un nodo del DOM possiamo ripetere la ricerca per trovare tutti i tags contenuti in quel particolare sotto-albero del DOM.

```
if footer_div:  
    # -- find firsts <p>...</p> inside <div id="footer"> ... </div>  
    author_p = footer_div.find("p")
```

- Possiamo anche esplorare il DOM utilizzando la notazione con il punto (.) come segue

```
author_p = footer_div.p
```

Così facendo però non siamo sicuri se l'elemento a cui stiamo tentando di accedere esiste davvero.

# EXTRACTING DATA FROM THE DOM

Una volta che abbiamo i tag di interesse possiamo accedere ai dati veri e propri come segue

```
# -- get all data
print(author_p.decode_contents()) # prints: <b>Author</b>: Leonardo Tamiano
# -- get only text data          # prints: Author: Leonardo Tamiano
print(author_p.text)
```

Il codice esempio fatto vedere è disponibile al seguente  
link:

**REAL (LIFE) EXAMPLE**

# Consideriamo la seguente pagina, che mostra gli orari delle lezioni.

Orario lezioni secondo semestre A.A. 2020-2021					
Dal 08-03-2021 al 18-06-2021					
Primo anno					
Lezioni in aula online se non diversamente specificato					
Ora	Lunedì	Martedì	Mercoledì	Giovedì	Venerdì
9:00					
10:00					
11:00	Geometria ed algebra ( )	Fisica ( )	Geometria ed algebra ( )	Fisica ( )	
12:00	Geometria ed algebra ( )	Fisica ( )	Geometria ed algebra ( )	Fisica ( )	
13:00					
14:00	Logica e reti logiche (Aula 13)		Programmazione dei calcolatori con laboratorio ( )	Logica e reti logiche (Aula 13)	Programmazione dei calcolatori con laboratorio ( )
15:00	Logica e reti logiche (Aula 13)		Programmazione dei calcolatori con laboratorio ( )	Logica e reti logiche (Aula 13)	Programmazione dei calcolatori con laboratorio ( )
16:00			Programmazione dei calcolatori con laboratorio ( )		Programmazione dei calcolatori con laboratorio ( )
17:00					
18:00					
Secondo anno					
Lezioni in aula online se non diversamente specificato					
Ora	Lunedì	Martedì	Mercoledì	Giovedì	Venerdì
9:00	Algoritmi e strutture dati ( )	Sistemi operativi e reti ( )	Fondamenti di informatica ( )	Algoritmi e strutture dati ( )	
10:00	Algoritmi e strutture dati ( )	Sistemi operativi e reti ( )	Fondamenti di informatica ( )	Algoritmi e strutture dati ( )	
11:00	Linguaggi e metodologie di programmazione ( )	Fondamenti di informatica ( )	Linguaggi e metodologie di programmazione ( )	Sistemi operativi e reti ( )	
12:00	Linguaggi e metodologie di programmazione ( )	Fondamenti di informatica ( )	Linguaggi e metodologie di programmazione ( )	Sistemi operativi e reti ( )	
13:00					
14:00	Basi di dati e di conoscenza (Aula 3PP2)	Ricerca operativa ( )	Basi di dati e di conoscenza (Aula 13)		
15:00	Basi di dati e di conoscenza (Aula 3PP2)	Ricerca operativa ( )	Basi di dati e di conoscenza (Aula 13)		
16:00		Ricerca operativa ( )			
17:00		Ricerca operativa ( )			
18:00					
Terzo anno					
Lezioni in aula online se non diversamente specificato					
Ora	Lunedì	Martedì	Mercoledì	Giovedì	Venerdì
9:00					
10:00	Ingegneria del software (9:30)				Ingegneria del software (9:30)
11:00	Ingegneria del software (10:30)	Programmazione Java per dispositivi mobili (11:30)	Modelli e linguaggi di simulazione (11:30)		Ingegneria del software (10:30)
12:00	Modelli e linguaggi di simulazione (11:30)	Programmazione Java per dispositivi mobili (12:30)	Modelli e linguaggi di simulazione (12:30)	Programmazione Web ( )	
13:00	Modelli e linguaggi di simulazione (12:30)			Programmazione Web ( )	
14:00		Programmazione Web ( )		Programmazione Java per dispositivi mobili ( )	
15:00		Programmazione Web ( )		Programmazione Java per dispositivi mobili ( )	
16:00					
17:00					
18:00					

Ripreso da: <http://www.informatica.uniroma2.it/>



Il nostro obiettivo è scaricare il file `.html` contenente le informazioni degli orari e salvare i dati in un file `.csv`, in modo poi da poterli processare a nostro piacimento.

Lo script python che fa questo è disponibile al seguente  
link:

