

# beautifulsoup

scraping web

# Scraping WEB

---

- Abbiamo visto come prelevare informazioni da siti web che attraverso delle API mettono a disposizione più o meno gratuitamente informazioni e dati
- Ma spesso queste informazioni NON sono a disposizione dell'utente per una facile acquisizione informatica
- Molti siti sono human—readable ma non computer—readable
- In questi casi l'attività di prelevamento va sotto il termine web scraping (letteralmente raschiatura del WEB)

# Scraping WEB

---

- Per poter utilizzare questa attività dovremo prendere un po' di confidenza con il linguaggio HTML
- Questo ci permetterà di capire come una pagina web è strutturata e come localizzare ed estrarre le informazioni che ci servono
- Python mette a disposizione una libreria chiamata bs4 che contiene un potente strumento chiamato BeautifulSoup

- Che cos'è l'HTML?
  - HTML è il linguaggio di markup standard per la creazione di pagine Web.
  - HTML è l'acronimo di Hyper Text Markup Language
  - HTML descrive la struttura di una pagina Web
  - HTML è costituito da una serie di elementi
  - Gli elementi HTML indicano al browser come visualizzare il contenuto
  - Gli elementi HTML sono rappresentati da tag
  - I tag HTML identificano parti di contenuto come "titolo", "paragrafo", "tabella" e così via
  - I browser non visualizzano i tag HTML, ma li usano per visualizzare il contenuto della pagina

# Scraping WEB

- Esempio

```
<!DOCTYPE html>
<html>
<head>
<title>Titolo</title>
</head>
<body>
  <h1>Un header</h1>
  <p>Un paragrafo</p>
</body>
</html>
```

The diagram illustrates the structure of an HTML document. It shows the opening and closing tags for the document, head, body, title, h1, and p elements. Arrows indicate the flow from the opening tag to the closing tag, showing the nesting of the elements. For example, the <html> tag is followed by <head>, which contains <title>. The <body> tag contains <h1> and <p>. The closing tags are shown in reverse order: </p>, </h1>, </body>, </head>, and </html>.

# Scraping WEB

---

- I tag HTML sono nomi di elementi racchiusi tra parentesi angolari:

`<tagname>content goes here...</tagname>`

- I tag HTML normalmente si presentano in coppie come `<p>` e `</p>`
- Il primo tag in una coppia è il tag di inizio, il secondo tag è il tag di fine
- Il tag di fine è scritto come il tag di inizio, ma con una barra avanti inserita prima del nome del tag

- Una pagina HTML può essere rappresentata come una serie di scatole (i TAGS) che possono contenere:
  - Testo e/o Immagini
  - Altre scatole

# Scraping WEB

```
<html>
```

```
<head>
```

```
<title>Page title</title>
```

```
</head>
```

```
<body>
```

```
<h1>This is a heading</h1>
```

```
<p>This is a paragraph.</p>
```

```
<p>This is another paragraph.</p>
```

```
</body>
```

```
</html>
```



```
Elements Console Sources >> 10 5 1 ⚙️ ⋮  
<!DOCTYPE html>  
<html lang="it">  
  <head>...</head>  
... > <body>...</body> == $0  
  > <iframe id="google_esf" name="google_esf" src="https://googleads.g.doubleclick.net/pagead/html/r20220223/r20190131/zrt_lookup.html" style="display: none;">...  
  </iframe> ad  
</html>
```

# Scraping WEB



Year	Version
1989	Tim Berners-Lee invented www
1991	Tim Berners-Lee invented HTML
1993	Dave Raggett drafted HTML+
1995	HTML Working Group defined HTML 2.0
1997	W3C Recommendation: HTML 3.2
1999	W3C Recommendation: HTML 4.01
2000	W3C Recommendation: XHTML 1.0
2008	WHATWG HTML5 First Public Draft
2012	<a href="#"><u>WHATWG HTML5 Living Standard</u></a>
2014	<a href="#"><u>W3C Recommendation: HTML5</u></a>
2016	W3C Candidate Recommendation: HTML 5.1
2017	<a href="#"><u>W3C Recommendation: HTML5.1 2nd Edition</u></a>
2017	<a href="#"><u>W3C Recommendation: HTML5.2</u></a>

- TAG principali
  - Header <h1>, <h2>, ..., <h6>
  - Sezione <div>
  - Paragrafo <p>
  - Contenitore <span>
  - Link <a>
  - Immagini <img>
  - Liste <ol><ul>
    - Elemento della lista <li>
  - Tabelle <table>
    - Riga in tabella <tr>
    - Cella in riga <td>

- TAG di formattazione
  - `<b>` - Testo in grassetto
  - `<strong>` - Testo importante
  - `<i>` - Testo in corsivo
  - `<em>` - Testo enfaticizzato
  - `<mark>` - Testo contrassegnato
  - `<small>` - Testo piccolo
  - `<del>` - Testo eliminato
  - `<ins>` - Testo inserito
  - `<sub>` - Testo in pedice
  - `<sup>` - Testo in apice

- Attributi

- Ogni tag può avere uno o più attributi che ne determinano il comportamento o l'aspetto
- Alcuni attributi sono specifici altri sono generici

- Esempio

- `<a href="url" target="_blank">contenuto</a>`
- ``
  - Il tag `<img>` non prevede alcun tag di chiusura, in HTML, pertanto, deve essere chiuso in questo modo

- Tutti gli elementi HTML possono avere attributi
- Gli attributi forniscono informazioni aggiuntive su un elemento
- Gli attributi sono sempre specificati nel tag iniziale
- Gli attributi di solito si presentano in coppie nome / valore come: nome = "valore"

- L'attributo style è può essere usato con quasi tutti i tag html
- Ha lo scopo di impostare attributi di stile seguendo la sintassi CSS
  - Il CSS (sigla di Cascading Style Sheets), in informatica, è un linguaggio usato per definire la formattazione di documenti HTML, XHTML e XML, ad esempio i siti web e relative pagine.
  - Le regole per comporre il CSS sono contenute in un insieme di direttive (Recommendations) emanate a partire dal 1996 dal W3C.

- CSS è l'acronimo di Cascading Style Sheets.
- CSS descrive come gli elementi HTML devono essere visualizzati su schermo, carta o su altri media.
- CSS consente di risparmiare molto lavoro. Può controllare il layout di più pagine Web contemporaneamente.
- I CSS possono essere aggiunti agli elementi HTML in 3 modi:
  - In linea: utilizzando l'attributo di stile negli elementi HTML
  - Interno: utilizzando un elemento `<style>` nella sezione `<head>`
  - Esterno: utilizzando un file CSS esterno



- Quando si usano i CSS esterni per attribuire uno stile si possono usare i TAG
  - id
  - class
- La differenza è minima e non entreremo nei dettagli
- Ma proprio su questi attributi si basa buona parte delle tecniche di scraping web

# Scraping WEB

---

- Ogni sito web che contenga informazioni interessanti da prelevare viene sviluppato attraverso le raccomandazioni del consorzio W3C
- Questo si trasforma in una serie di attività sempre uguali che il programmatore web mette in atto per formattare le proprie pagine in maniera opportuna
- Questa peculiarità ci permetterà di facilmente localizzare le scatole che contengono i nostri dati

# Scraping WEB

---

- tripadvisor
- facebook
- twitter
- Alcuni forum e home page interessanti

# Scraping WEB

---

- BeautifulSoup è una libreria Python per ottenere dati da HTML, XML e altri linguaggi di markup.
- Supponiamo che ci siano alcune pagine web che visualizzano dati rilevanti, come la data o l'indirizzo, ma che non forniscono alcun modo per scaricare direttamente i dati.
- BeautifulSoup aiuta a estrarre determinati contenuti da una pagina web, rimuovere il markup HTML e salvare le informazioni.
- È uno strumento per il web scraping che aiuta a ripulire e analizzare i documenti web.

# Scraping WEB

Comunità

Portale Comunità

Bar

Il Wikipediano

Fai una donazione

Contatti

Strumenti

Puntano qui

Modifiche correlate

Carica su Commons

Pagine speciali

Link permanente

Informazioni pagina

Elemento Wikidata

Cita questa voce

Stampa/esporta

Crea un libro

Scarica come PDF

Versione stampabile

In altre lingue

Català

Deutsch

Ελληνικά

English

Français

हिन्दी

Hrvatski

Slovenščina

中文

Altre 90

Modifica collegamenti

1 Stati indipendenti (196)

2 Stati autoproclamati (8)










3 Curiosità

4 Note

5 Bibliografia

6 Voci correlate

Stati indipendenti (196) [ modifica | modifica wikitesto ]

Stato	Capitale	Nome locale	Anno	Altitudine (m s.l.m.)	Superficie (km²)	Abit.
 Afghanistan	Kabul	کابل	1779	1 791	1 023	3 67
 Albania	Tirana	Tiranë	1920	110	1 110	895
 Algeria	Algeri	مدينة الجزائر	1962	424	363	3 41
 Andorra	La Vella		1278	1 013	30	22 8
 Angola	Luanda		1975	6	113	5 17
 Antigua e Barbuda	Saint John's		1981	8	10	22 2
 Arabia Saudita	Riyād	الرياض	1818	612	1798	9 80
 Argentina	Buenos Aires	Ciudad Autónoma de Buenos Aires	1816	25	203	3 06
 Armenia	Erevan	Երևան	23 agosto 1990	998	223	1 09

Elements

Console

Sources

labelledby="mw-toc-heading">...</div>

<h2>...</h2>

<table class="wikitable sortable jquery-tablesorter" style="font-size:100%"> == \$0

<thead>...</thead>

<tbody>

<tr>

<td>

<span style="white-space:nowrap">...</span>

<a href="/wiki/Afghanistan" title="Afghanistan">Afghanistan</a>

</td>

<td>

<a href="/wiki/Kabul" title="Kabul">Kabul</a>

</td>

<td>کابل</td>

<td>1779</td>

<td>1 791</td>

<td>1 023</td>

#mw-content-text

div

table.wikitable.sortable.jquery-tablesorter

Styles

Event Listeners

DOM Breakpoints

Properties

Accessibility

Filter

:hov .cls

- Nella immagine precedente (pagina web wikipedia sulle capitali nel mondo) sono le capitali di nazioni nel mondo
- Risulta molto chiaro il concetto di scatole nidificate (o scatole cinesi)
- Si nota infatti come la scatola che contiene tutti i paesi risulti un table di classe wikitable sortable jquery-tablesorter
- Inoltre le informazioni per ciascun paese sono a loro volta inserite in una riga tr
- E le singole informazioni sono inserite in celle td

# Scraping WEB

---

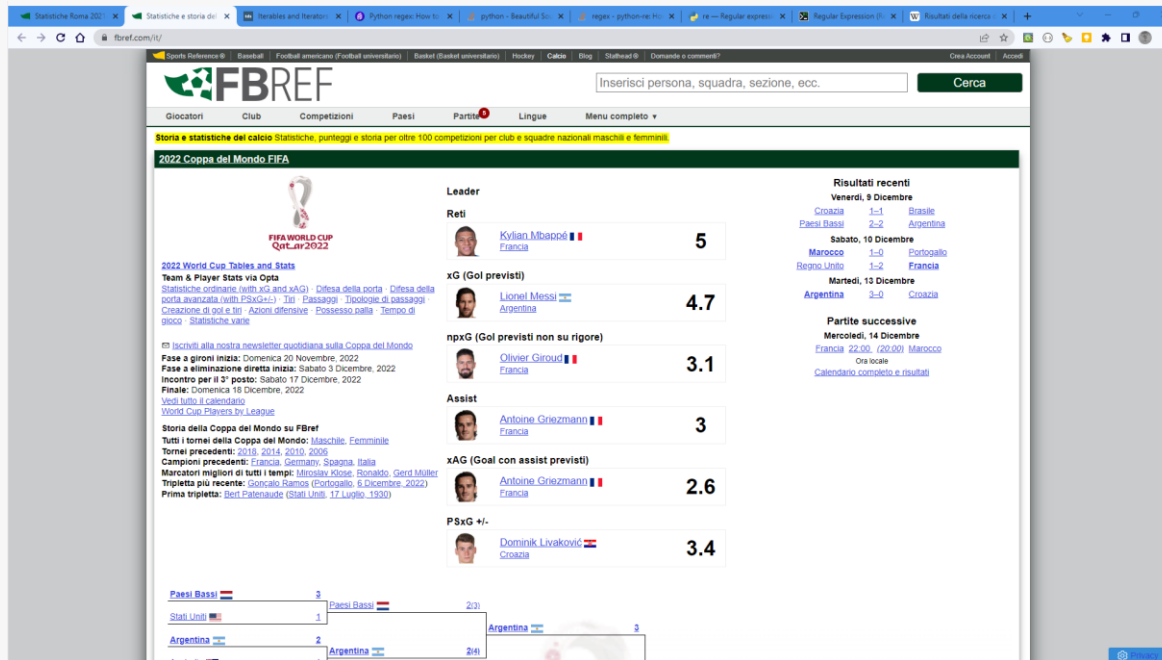
- BeautifulSoup è in grado di trovare tutte queste scatole semplicemente passandogli il tipo (ad esempio `div`) e un qualche attributo (ad esempio `class="xyz"`) che la renda possibilmente univoca nel testo HTML
- Ad esempio cercare tutti i `table` di classe `wikitable` ci permetterà di isolare queste particolari scatole
- Una volta isolate le scatole dovremo estrarre le informazioni utilizzando gli strumenti messi a disposizione dalla libreria

- I metodi che l'oggetto BeautifulSoup mette a disposizione sono vari
- I più usati sono:
  - `.find()` trova una scatola di un certo tipo con specifici attributi
  - `.find_all()` trova tutte le scatole di un certo tipo con specifici attributi (lista)
  - `.text` ritorna il puro testo della scatola
  - `.attrs[]` (dizionario di attributi) tutte le coppie chiave valore della scatola

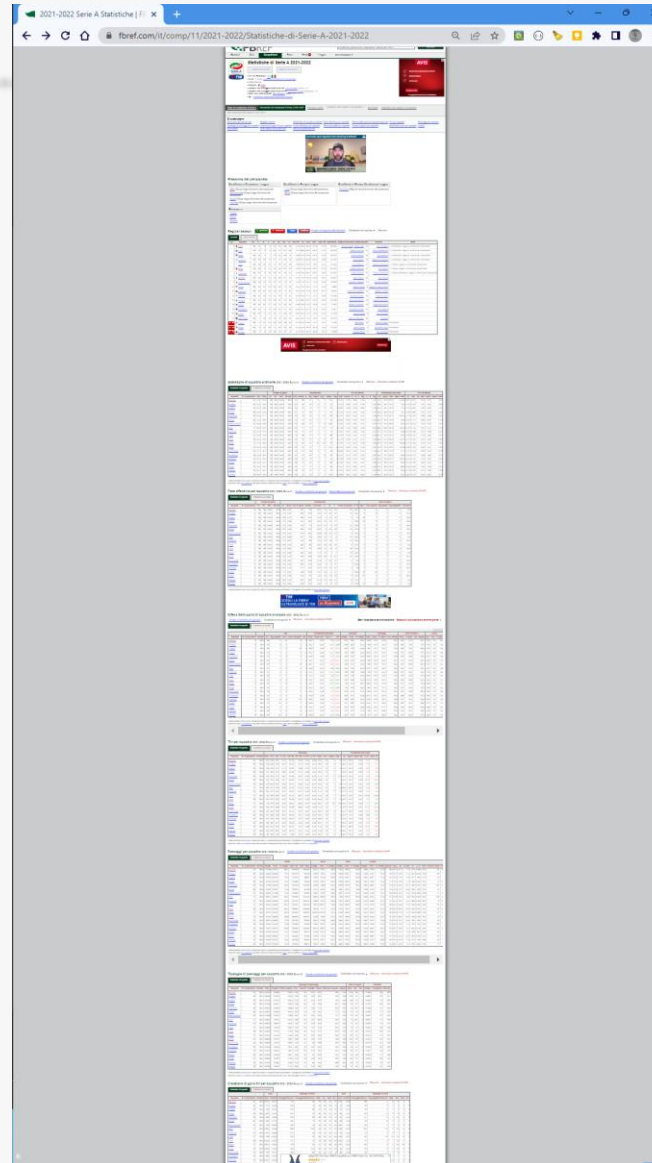


- Metodo `find()` e `find_all()`
  - Signature: `find(name, attrs, recursive, string)`
  - Signature: `find_all(name, attrs, recursive, string, limit)`
    - `attrs` è un dizionario con gli attributi della scatola `name`

- **Storia e statistiche del calcio** Statistiche, punteggi e storia per oltre 100 competizioni per club e squadre nazionali maschili e femminili.



# Molte tabelle



The screenshot shows a web browser window with a single tab titled "2021-2022 Serie A Statistiche | F...". The address bar displays the URL "fbref.com/it/comp/11/2021-2022/Statistiche-di-Serie-A-2021-2022". The page content is a vertical stack of several tables, each representing different statistical data for the 2021-2022 Serie A season. The tables are separated by small horizontal dividers and include various headers and data rows. The browser's interface, including the address bar and navigation buttons, is visible at the top of the window.

Condividere ed esportare ▼ Glossario

Generali		Casa/Trasferta																		
Pos.	Squadra	PG	V	N	P	Rf	Rs	DR	Pt	Pts/MP	xG	xGA	xGD	xGD/90	Spettatori	Miglior marcatore della squadra	Portiere	Note		
1	 <a href="#">Milan</a>	38	26	8	4	69	31	+38	86	2,26	68.2	40.4	+27.8	0,73	44.015	<a href="#">Olivier Giroud</a> , <a href="#">Rafael Leão</a> - 11	<a href="#">Mike Maignan</a>	→ Champions League al termine del campionato		
2	 <a href="#">Inter</a>	38	25	9	4	84	32	+52	84	2,21	88.4	44.1	+44.3	+1,17	44.473	<a href="#">Lautaro Martínez</a> - 21	<a href="#">Samir Handanović</a>	→ Champions League al termine del campionato		
3	 <a href="#">Napoli</a>	38	24	7	7	74	31	+43	79	2,08	66.3	36.8	+29.6	0,78	28.119	<a href="#">Victor Osimhen</a> - 14	<a href="#">David Ospina</a>	→ Champions League al termine del campionato		
4	 <a href="#">Juventus</a>	38	20	10	8	57	37	+20	70	1,84	57.3	42.8	+14.5	0,38	22.621	<a href="#">Paulo Dybala</a> - 10	<a href="#">Wojciech Szczęsny</a>	→ Champions League al termine del campionato		
5	 <a href="#">Lazio</a>	38	18	10	10	77	58	+19	64	1,68	60.2	51.9	+8.3	0,22	23.263	<a href="#">Ciro Immobile</a> - 27	<a href="#">Thomas Strakosha</a>	→ Europa League al termine del campionato		
6	 <a href="#">Roma</a>	38	18	9	11	59	43	+16	63	1,66	70.3	41.8	+28.5	0,75	41.929	<a href="#">Tammy Abraham</a> - 17	<a href="#">Rui Patrício</a>	→ Europa League al termine del campionato		
7	 <a href="#">Fiorentina</a>	38	19	5	14	59	51	+8	62	1,63	65.6	46.1	+19.5	0,51	21.107	<a href="#">Dušan Vlahović</a> - 17	<a href="#">Pietro Terracciano</a>	→ Europa Conference League al termine del campionato		
8	 <a href="#">Atalanta</a>	38	16	11	11	65	48	+17	59	1,55	71.2	50.4	+20.8	0,55	10.447	<a href="#">Mario Pašalić</a> - 13	<a href="#">Juan Musso</a>			
9	 <a href="#">Hellas Verona</a>	38	14	11	13	65	59	+6	53	1,39	57.6	53.7	+3.9	0,10	13.894	<a href="#">Giovanni Simeone</a> - 17	<a href="#">Lorenzo Montipò</a>			
10	 <a href="#">Torino</a>	38	13	11	14	46	41	+5	50	1,32	53.1	43.7	+9.3	0,25	9.846	<a href="#">Andrea Belotti</a> - 8	<a href="#">Vanja Milinković-Savić</a>			
11	 <a href="#">Sassuolo</a>	38	13	11	14	64	66	-2	50	1,32	62.0	72.4	-10.3	-0,27	8.362	<a href="#">Gianluca Scamacca</a> - 16	<a href="#">Andrea Consigli</a>			
12	 <a href="#">Udinese</a>	38	11	14	13	61	58	+3	47	1,24	58.2	57.1	+1.1	0,03	12.144	<a href="#">Gerard Deulofeu</a> - 13	<a href="#">Marco Silvestri</a>			
13	 <a href="#">Bologna</a>	38	12	10	16	44	55	-11	46	1,21	48.1	59.7	-11.6	-0,31	14.158	<a href="#">Marko Arnautović</a> - 14	<a href="#">Łukasz Skorupski</a>			
14	 <a href="#">Empoli</a>	38	10	11	17	50	70	-20	41	1,08	49.4	76.2	-26.8	-0,71	6.356	<a href="#">Andrea Pinamonti</a> - 13	<a href="#">Guglielmo Vicario</a>			
15	 <a href="#">Sampdoria</a>	38	10	6	22	46	63	-17	36	0,95	41.2	62.2	-21.0	-0,55	9.417	<a href="#">Francesco Caputo</a> - 11	<a href="#">Emil Audero</a>			
16	 <a href="#">Spezia</a>	38	10	6	22	41	71	-30	36	0,95	42.3	72.2	-29.9	-0,79	6.709	<a href="#">Daniele Verde</a> - 8	<a href="#">Ivan Provedel</a>			
17	 <a href="#">Salernitana</a>	38	7	10	21	33	78	-45	31	0,82	42.2	71.2	-29.0	-0,76	15.073	<a href="#">Federico Bonazzoli</a> - 10	<a href="#">Vid Belec</a>			
▼ 18	 <a href="#">Cagliari</a>	38	6	12	20	34	68	-34	30	0,79	42.9	64.6	-21.7	-0,57	9.718	<a href="#">João Pedro</a> - 13	<a href="#">Alessio Cragno</a>	Retrocessa		
▼ 19	 <a href="#">Genoa</a>	38	4	16	18	27	60	-33	28	0,74	41.9	58.3	-16.4	-0,43	12.326	<a href="#">Mattia Destro</a> - 9	<a href="#">Salvatore Sirigu</a>	Retrocessa		
▼ 20	 <a href="#">Venezia</a>	38	6	9	23	34	69	-35	27	0,71	40.1	80.9	-40.8	-1,07	6.648	<a href="#">Thomas Henry</a> - 9	<a href="#">Niki Maenpää</a>	Retrocessa		





































# Team - Overall

Statistiche ordinarie 2021-2022 Roma: Serie A 

Condividere ed esportare

Glossario

Interruttore statistiche Per90

Giocatore	Nazione	Ruolo	Età	Tempo di gioco				Rendimento							Per 90 minuti							Prestazione prevista				Per 90 minuti				Partite
				PG	Tit	Min	90 min	Reti	Assist	R - Rig	Rigori	Rig T	Amm.	Esp.	Reti	Assist	R + A	R - Rig	R + A	R - Rig	xG	npG	xAG	npG+xAG	xG	npG	xAG	npG+xAG		
Rui Patrício		Por	33	38	38	3.420	38,0	0	0	0	0	2	0	0,00	0,00	0,00	0,00	0,00	0,0	0,0	0,0	0,0	0,0	0,00	0,00	0,00	0,00	0,00	Partite	
Tammy Abraham		Att	23	37	36	3.084	34,3	17	4	14	3	3	9	0	0,50	0,12	0,61	0,41	0,53	19,2	16,8	4,9	21,8	0,56	0,14	0,70	0,49	0,63	Partite	
Gianluca Mancini		Dif	25	33	33	2.878	32,0	0	0	0	0	0	16	1	0,00	0,00	0,00	0,00	0,00	3,0	2,9	0,4	3,3	0,09	0,01	0,11	0,09	0,10	Partite	
Roger Ibanes		Dif	22	34	32	2.889	32,1	3	0	3	0	0	9	0	0,09	0,00	0,09	0,09	0,09	2,5	2,5	0,6	3,1	0,08	0,02	0,10	0,08	0,10	Partite	
Rick Karsdorp		Dif,Att	26	36	32	2.883	32,0	0	2	0	0	0	9	1	0,00	0,06	0,06	0,00	0,06	0,0	0,0	5,3	5,3	0,00	0,16	0,17	0,00	0,17	Partite	
Bryan Cristante		Cen	26	34	29	2.678	29,8	2	0	2	0	0	9	0	0,07	0,00	0,07	0,07	0,07	3,0	3,0	2,8	5,8	0,10	0,09	0,19	0,10	0,19	Partite	
Hennikh Mchitarjan		Cen,Att	32	31	29	2.495	27,7	5	6	5	0	0	4	1	0,18	0,22	0,40	0,18	0,40	4,7	4,7	5,4	10,1	0,17	0,19	0,37	0,17	0,37	Partite	
Lorenzo Pellegrini		Cen	25	28	27	2.289	25,4	9	3	7	2	3	8	1	0,35	0,12	0,47	0,28	0,39	10,1	7,6	7,9	15,5	0,40	0,31	0,71	0,30	0,61	Partite	
Jordan Veretout		Cen	28	36	26	2.319	25,8	4	8	3	1	2	6	0	0,16	0,31	0,47	0,12	0,43	4,9	3,3	6,4	9,7	0,19	0,25	0,44	0,13	0,38	Partite	
Chris Smalling		Dif	31	27	23	2.091	23,2	3	0	3	0	0	0	0	0,13	0,00	0,13	0,13	0,13	2,5	2,5	0,6	3,1	0,11	0,02	0,13	0,11	0,13	Partite	
Nicolò Zaniolo		Att,Cen	22	28	23	1.971	21,9	2	1	2	0	0	12	2	0,09	0,05	0,14	0,09	0,14	6,2	6,2	3,6	9,8	0,28	0,16	0,45	0,28	0,45	Partite	
Martin Villa		Dif,Att	23	26	18	1.565	17,4	0	1	0	0	0	2	0	0,00	0,06	0,06	0,00	0,06	0,9	0,9	1,3	2,1	0,05	0,07	0,12	0,05	0,12	Partite	
Mansur Kumbulla		Dif	21	17	13	1.025	11,4	0	0	0	0	0	6	0	0,00	0,00	0,00	0,00	0,00	0,9	0,9	0,1	1,0	0,08	0,01	0,09	0,08	0,09	Partite	
Sérgio Oliveira		Cen	29	14	13	892	9,9	2	0	1	1	1	5	0	0,20	0,00	0,20	0,10	0,10	1,4	0,7	1,8	2,4	0,14	0,18	0,32	0,07	0,25	Partite	
Nicola Zalewski		Dif	19	16	9	793	8,8	0	0	0	0	0	1	0	0,00	0,00	0,00	0,00	0,00	0,6	0,6	0,6	1,1	0,07	0,06	0,13	0,07	0,13	Partite	
Stoichan El Shaarawy		Att,Dif	28	27	8	1.055	11,7	3	0	3	0	0	1	0	0,26	0,00	0,26	0,26	0,26	3,6	3,6	3,3	6,9	0,30	0,28	0,59	0,30	0,59	Partite	
Ainslie Maitland-Niles		Dif	23	8	7	487	5,4	0	0	0	0	0	0	0	0,00	0,00	0,00	0,00	0,00	0,4	0,4	0,6	1,0	0,07	0,11	0,18	0,07	0,18	Partite	
Felix Afena-Gyan		Att,Cen	18	17	6	668	7,4	2	0	2	0	0	5	1	0,27	0,00	0,27	0,27	0,27	1,5	1,5	0,7	2,2	0,20	0,10	0,30	0,20	0,30	Partite	
Eldor Shomurodov		Att,Cen	26	28	5	784	8,7	3	4	3	0	0	2	0	0,34	0,46	0,80	0,34	0,80	4,6	4,6	2,8	7,5	0,53	0,33	0,86	0,53	0,86	Partite	
Carlos Pérez		Cen,Att	23	19	3	554	6,2	1	1	1	0	0	2	0	0,16	0,16	0,32	0,16	0,32	1,9	1,9	1,3	3,2	0,30	0,22	0,52	0,30	0,52	Partite	
Riccardo Calafiori		Dif	19	6	2	174	1,9	0	1	0	0	0	2	0	0,00	0,52	0,52	0,00	0,52	0,0	0,0	0,6	0,6	0,00	0,32	0,32	0,00	0,32	Partite	
Amadou Diawara		Cen	24	4	2	170	1,9	0	0	0	0	0	0	0	0,00	0,00	0,00	0,00	0,00	0,0	0,0	0,1	0,1	0,02	0,03	0,05	0,02	0,05	Partite	
Leonardo Spinazzola		Dif	28	3	2	127	1,4	0	0	0	0	0	1	0	0,00	0,00	0,00	0,00	0,00	0,1	0,1	0,1	0,2	0,05	0,09	0,15	0,05	0,15	Partite	
Borja Mayoral		Att,Dif	24	5	1	96	1,1	0	0	0	0	0	0	0	0,00	0,00	0,00	0,00	0,00	0,3	0,3	0,0	0,3	0,25	0,00	0,25	0,25	0,25	Partite	
Ebrima Darboe		Cen	20	1	1	63	0,7	0	0	0	0	0	0	0	0,00	0,00	0,00	0,00	0,00	0,1	0,1	0,0	0,1	0,12	0,00	0,12	0,12	0,12	Partite	
Edoardo Bove		Cen,Dif	19	11	0	68	0,8	1	0	1	0	0	0	0	1,32	0,00	1,32	1,32	1,32	0,0	0,0	0,0	0,0	0,06	0,00	0,06	0,06	0,06	Partite	
Cristian Volpato		Att,Cen	17	3	0	38	0,4	1	0	1	0	0	0	0	2,37	0,00	2,37	2,37	2,37	0,1	0,1	0,1	0,2	0,28	0,13	0,41	0,28	0,41	Partite	
Bryan Reynolds		Dif	20	1	0	2	0,0	0	0	0	0	0	0	0	0,00	0,00	0,00	0,00	0,00	0,0	0,0	0,0	0,0	0,00	0,00	0,00	0,00	0,00	Partite	
Dimitrios Keramiitis		Cen	17	1	0	1	0,0	0	0	0	0	0	0	0	0,00	0,00	0,00	0,00	0,00	0,0	0,0	0,0	0,0	0,00	0,00	0,00	0,00	0,00	Partite	
Daniel Fuzato		Por	24	0	0			0	0	0	0	0	0	1															Partite	
Pietro Boer		Por	19	0	0																									Partite
Davide Mastrantonio		Por	17	0	0																									Partite
Filippo Missiroli		Dif	17	0	0																									Partite
Maissa Ndiaye		Dif	19	0	0																									Partite
Jan Oliveras		Dif	17	0	0																									Partite
Filippo Trini		Cen	19	0	0																									Partite
Gonzalo Villar		Cen	23	0	0																									Partite
Totale di squadra				26,4	38	418	3.420	38,0	58	31	51	7	9	111	8	1,53	0,82	2,34	1,34	2,16	70,3	63,3	50,5	113,8	1,85	1,33	3,18	1,67	2,99	
Totale avversario				26,8	38	418	3.420	38,0	41	23	36	5	6	106	8	1,08	0,61	1,68	0,95	1,55	41,8	37,7	29,1	66,9	1,10	0,77	1,87	0,99	1,76	


















I totali potrebbero non essere completi per tutte le competizioni professionistiche. Ti consigliamo di consultare la [nota sulla copertura](#).

# Player - Overall

## Statistiche ordinarie: Campionati nazionali

[Registri dei goal](#) Condividere ed esportare ▾ [Glossario](#) [Interruttore statistiche Per90](#)

[Scorri verso destra per altre statistiche](#) · [Passa alla visualizzazione a schermo grande](#) ►

						Tempo di gioco				Rendimento						Per 90 minuti						Prestazione prevista				Per 90		
Stagione	Età	Squadra	Paese	Competizione	Piazzamento	PG	Tit	Min	90 min	Reti	Assist	R - Rig	Rigori	Rig T	Amm.	Esp.	Reti	Assist	R + A	R - Rig	R + A - Rig	xG	npG	xAG	npG+xAG	xG	xAG	xG+xAG
2006-2007	18	<a href="#">Sporting CP</a>	 POR	<a href="#">1. Primeira Liga</a>	2º	1	1	90	1.0	0	0	0	0	0	0	0	0,00	0,00	0,00	0,00	0,00							
2007-2008	19	<a href="#">Sporting CP</a>	 POR	<a href="#">1. Primeira Liga</a>	2º	20	20	1.800	20.0	0	0	0	0	0	0	0	0,00	0,00	0,00	0,00	0,00							
2008-2009	20	<a href="#">Sporting CP</a>	 POR	<a href="#">1. Primeira Liga</a>	2º	26	26	2.340	26.0	0	0	0	0	0	1	0	0,00	0,00	0,00	0,00	0,00							
2009-2010	21	<a href="#">Sporting CP</a>	 POR	<a href="#">1. Primeira Liga</a>	4º	30	30	2.700	30.0	0	0	0	0	0	1	0	0,00	0,00	0,00	0,00	0,00							
2010-2011	22	<a href="#">Sporting CP</a>	 POR	<a href="#">1. Primeira Liga</a>	3º	30	30	2.700	30.0	0	0	0	0	0	4	0	0,00	0,00	0,00	0,00	0,00							
2011-2012	23	<a href="#">Sporting CP</a>	 POR	<a href="#">1. Primeira Liga</a>	4º	28	28	2.519	28.0	0	0	0	0	0	3	0	0,00	0,00	0,00	0,00	0,00							
2012-2013	24	<a href="#">Sporting CP</a>	 POR	<a href="#">1. Primeira Liga</a>	7º	30	30	2.700	30.0	0	0	0	0	0	3	0	0,00	0,00	0,00	0,00	0,00							
2013-2014	25	<a href="#">Sporting CP</a>	 POR	<a href="#">1. Primeira Liga</a>	2º	30	30	2.700	30.0	0	0	0	0	0	3	0	0,00	0,00	0,00	0,00	0,00							
2014-2015	26	<a href="#">Sporting CP</a>	 POR	<a href="#">1. Primeira Liga</a>	3º	33	33	2.970	33.0	0	0	0	0	0	3	0	0,00	0,00	0,00	0,00	0,00							
2015-2016	27	<a href="#">Sporting CP</a>	 POR	<a href="#">1. Primeira Liga</a>	2º	34	34	2.996	33.3	0	0	0	0	0	3	1	0,00	0,00	0,00	0,00	0,00							
2016-2017	28	<a href="#">Sporting CP</a>	 POR	<a href="#">1. Primeira Liga</a>	3º	31	31	2.790	31.0	0	0	0	0	0	2	0	0,00	0,00	0,00	0,00	0,00							
2017-2018	29	<a href="#">Sporting CP</a>	 POR	<a href="#">1. Primeira Liga</a>	3º	34	34	3.060	34.0	0	0	0	0	0	4	0	0,00	0,00	0,00	0,00	0,00							
2018-2019	30	<a href="#">Wolves</a>	 ENG	<a href="#">1. Premier League</a>	7º	37	37	3.329	37.0	0	0	0	0	0	1	0	0,00	0,00	0,00	0,00	0,00	0,0	0,0	0,0	0,0	0,00	0,00	0,00
2019-2020	31	<a href="#">Wolves</a>	 ENG	<a href="#">1. Premier League</a>	7º	38	38	3.420	38.0	0	0	0	0	0	0	0	0,00	0,00	0,00	0,00	0,00	0,0	0,0	0,0	0,0	0,00	0,00	0,00
2020-2021	32	<a href="#">Wolves</a>	 ENG	<a href="#">1. Premier League</a>	13º	37	37	3.329	37.0	0	0	0	0	0	1	0	0,00	0,00	0,00	0,00	0,00	0,0	0,0	0,0	0,0	0,00	0,00	0,00
2021-2022	33	<a href="#">Roma</a>	 ITA	<a href="#">1. Serie A</a>	6º	38	38	3.420	38.0	0	0	0	0	0	2	0	0,00	0,00	0,00	0,00	0,00	0,0	0,0	0,0	0,0	0,00	0,00	0,00
2022-2023	34	<a href="#">Roma</a>	 ITA	<a href="#">1. Serie A</a>	7º	15	15	1.350	15.0	0	0	0	0	0	1	0	0,00	0,00	0,00	0,00	0,00	0,0	0,0	0,0	0,0	0,00	0,00	0,00
17 stagioni		3 club		3 campionati		492	492	44.213	491.3	0	0	0	0	0	32	1	0,00	0,00	0,00	0,00	0,00	0,0	0,0	0,0	0,0	0,00	0,00	0,00
			Paese	Competizione	Piazzamento	PG	Tit	Min	90 min	Reti	Assist	R - Rig	Rigori	Rig T	Amm.	Esp.	Reti	Assist	R + A	R - Rig	R + A - Rig	xG	npG	xAG	npG+xAG	xG	xAG	xG+xAG
Sporting CP (12 stagioni)				1 campionato		327	327	29.365	326.3	0	0	0	0	0	27	1	0,00	0,00	0,00	0,00	0,00							
Wolves (3 stagioni)				1 campionato		112	112	10.078	112.0	0	0	0	0	0	2	0	0,00	0,00	0,00	0,00	0,00	0,0	0,0	0,0	0,0	0,00	0,00	0,00
Roma (2 stagioni)				1 campionato		53	53	4.770	53.0	0	0	0	0	0	3	0	0,00	0,00	0,00	0,00	0,00	0,0	0,0	0,0	0,0	0,00	0,00	0,00
Primeira Liga (12 stagioni)						327	327	29.365	326.3	0	0	0	0	0	27	1	0,00	0,00	0,00	0,00	0,00							
Premier League (3 stagioni)						112	112	10.078	112.0	0	0	0	0	0	2	0	0,00	0,00	0,00	0,00	0,00	0,0	0,0	0,0	0,0	0,00	0,00	0,00
Serie A (2 stagioni)						53	53	4.770	53.0	0	0	0	0	0	3	0	0,00	0,00	0,00	0,00	0,00	0,0	0,0	0,0	0,0	0,00	0,00	0,00

# Scraping Soccer

- Vediamo con qualche semplice esempio come estrarre alcune tabelle dal sito di FBREF
  - Estrazione di tutte le tabelle presenti in una pagina:

```
def get_tables(url):  
    html = requests.get(url).text  
    soup = BeautifulSoup(html)  
    tables = soup.find_all("table", {"class": "stats_table"})  
    return tables
```

# Scraping Soccer

- Estrazione dati da una tabella in una lista di dizionari

```
def get_table(t, skip=0):
    trs = t.find_all("tr")
    # header
    i_trs = iter(trs)
    for i in range(skip):
        next(i_trs)
    head = next(i_trs)
    ths = head.find_all("th")
    keys = []
    for th in ths:
        s = th.text
        s = re.sub(r"(\s)|(/)|(\.)", "_", s)
        s = re.sub("(à)|(è)|(ì)|(ò)|(ù)|(é)", "_", s)
        keys.append(s)
    data = []

    for tr in i_trs:
        tds = tr.find_all(re.compile("(th)|(td)"))
        record = {}
        for i, td in enumerate(tds):
            record[keys[i]] = td.text.replace(",", ".")
        data.append(record)
    return data
```



# Scraping Soccer

- Estrazione di url utili

```
def get_url(t, col=0, skip=0):
    urls = {}
    trs = t.find_all("tr")
    i_trs = iter(trs)
    for i in range(skip):
        next(i_trs)
    next(i_trs)
    for tr in i_trs:
        tds = tr.find_all(re.compile("(th)|(td)"))
        if len(tds) > 0:
            href = tds[col].find("a")
            if href is not None:
                urls[href.text] = BASE_URL + href.attrs["href"]
            else:
                urls[tds[col].text] = None
    return urls
```

# DB – Creazione tabelle

- Useremo i nomi delle colonne come attributi (li abbiamo ‘ripuliti’ apposta)

```
def create_table(db, table, data):
    fields = ""
    for k in d[0].keys():
        fields += f"\n\t{k} TEXT,"
    query = f"CREATE TABLE IF NOT EXISTS {table} ({fields[:-1]})"
    db.execute(query)
    db.commit()
```

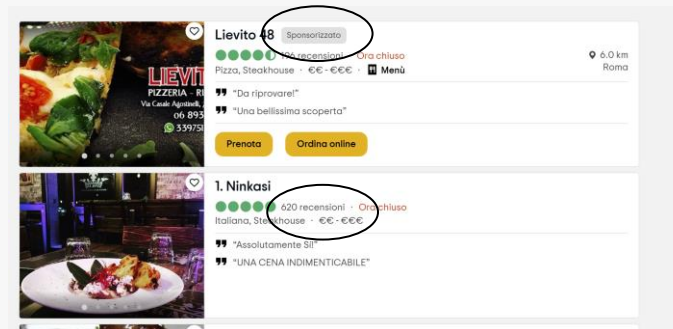
# DB – Inserimento dati

- Il 90% delle tabelle presenti rispetta lo standard sfruttato qui:

```
def to_db(db, table, data):
    for d in data:
        fields = ""
        values = ""
        for k in d.keys():
            fields += f"{k},"
            values += f"{d[k]}',"
        query = f"INSERT INTO {table} ({fields[:-1]}) VALUES ({values[:-3]})"
        db.execute(query)
    db.commit()
```

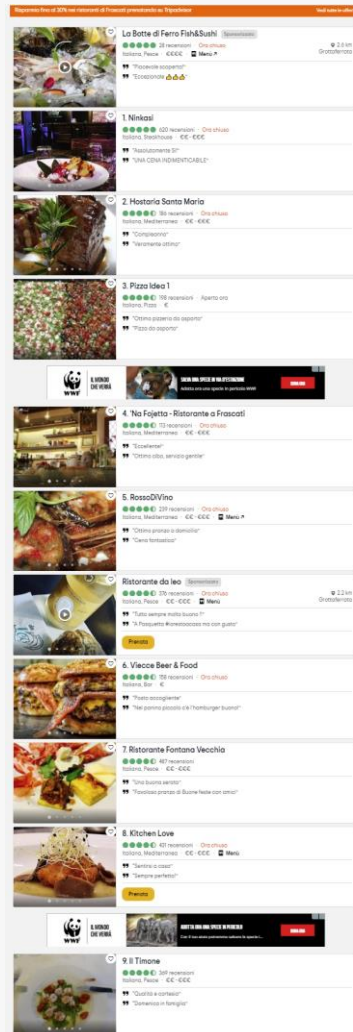
- Alcune tabelle hanno delle righe di riepilogo che andrebbero ignorate (es. `data[:-2]` per la tabella Statistiche ordinarie per squadra)

- Case study
  - Raccogliere recensioni per ristoranti di una città
    - Frascati
    - 10 recensioni per ristorante (light version)
  - Dati
    - URL: [https://www.tripadvisor.it/Restaurants-g229462-Frascati\\_Province\\_of\\_Rome\\_Lazio.html](https://www.tripadvisor.it/Restaurants-g229462-Frascati_Province_of_Rome_Lazio.html)



```
<!--trkP:eateries-->
▼<div class="react-container component-widget " id="component_2" data-component-props:
component="@ta/restaurants.list" data-component-init="data-component-init">
  ▶<div class="restaurants-list-List__wrapper--3PzDL">...</div> == $0
</div>
<!--etk-->
▶<script type="text/javascript">...</script>
  <script type="text/javascript"> injektReviewsContent(); </script>
</div>
```

```
<div class="react-container component-widget" id="component_2" data-component-props="page-manifest" data-component>
  <div class="restaurants-list-List_wrapper--3PzDL">
    <div data-test="SL_list_item" class="_1llCuDZj">...</div>
    <div data-test="1_list_item" class="_1llCuDZj">...</div>
    <div data-test="2_list_item" class="_1llCuDZj">...</div>
    <div data-test="3_list_item" class="_1llCuDZj">...</div>
    <div class="restaurants-list-CPMAAds__cellContainerWrapper--wQdtU">...</div>
    <div data-test="4_list_item" class="_1llCuDZj">...</div>
    <div data-test="5_list_item" class="_1llCuDZj">...</div>
    <div data-test="SL_list_item" class="_1llCuDZj">...</div>
    <div data-test="6_list_item" class="_1llCuDZj">...</div>
    <div data-test="7_list_item" class="_1llCuDZj">...</div>
    <div data-test="8_list_item" class="_1llCuDZj">...</div>
    <div class="restaurants-list-CPMAAds__cellContainerWrapper--wQdtU">...</div>
    <div data-test="9_list_item" class="_1llCuDZj">...</div>
    <div data-test="10_list_item" class="_1llCuDZj">...</div>
    <div data-test="SL_list_item" class="_1llCuDZj">...</div>
    <div data-test="11_list_item" class="_1llCuDZj">...</div>
    <div data-test="12_list_item" class="_1llCuDZj">...</div>
    <div data-test="13_list_item" class="_1llCuDZj">...</div>
    <div class="restaurants-list-CPMAAds__cellContainerWrapper--wQdtU">...</div>
    <div data-test="14_list_item" class="_1llCuDZj">...</div>
    <div data-test="15_list_item" class="_1llCuDZj">...</div>
    <div data-test="SL_list_item" class="_1llCuDZj">...</div>
    <div data-test="16_list_item" class="_1llCuDZj">...</div>
    <div data-test="17_list_item" class="_1llCuDZj">...</div>
    <div data-test="18_list_item" class="_1llCuDZj">...</div>
    <div class="restaurants-list-CPMAAds__cellContainerWrapper--wQdtU">...</div>
    <div data-test="19_list_item" class="_1llCuDZj">...</div>
    <div data-test="20_list_item" class="_1llCuDZj">...</div>
    <div data-test="SL_list_item" class="_1llCuDZj">...</div>
    <div data-test="21_list_item" class="_1llCuDZj">...</div>
    <div data-test="22_list_item" class="_1llCuDZj">...</div>
    <div data-test="23_list_item" class="_1llCuDZj">...</div>
    <div class="restaurants-list-CPMAAds__cellContainerWrapper--wQdtU">...</div>
    <div data-test="24_list_item" class="_1llCuDZj">...</div>
    <div data-test="25_list_item" class="_1llCuDZj">...</div>
    <div data-test="SL_list_item" class="_1llCuDZj">...</div>
    <div data-test="26_list_item" class="_1llCuDZj">...</div>
    <div data-test="27_list_item" class="_1llCuDZj">...</div>
    <div data-test="28_list_item" class="_1llCuDZj">...</div>
    <div data-test="29_list_item" class="_1llCuDZj">...</div>
    <div data-test="30_list_item" class="_1llCuDZj">...</div>
    <div data-test="SL_list_item" class="_1llCuDZj">...</div>
  </div>
</div>
```




## Ninkasi

Profilo richiesto

620 recensioni #1 di 166 ristoranti a Frascati €€-€€€, Italiana, Steakhouse, Mediterranea

Via Di Passo Lombardo 13/15, 00133 Frascati Italia +39 06 7265 0935 Sito web Ora chiuso: Vedi tutti gli orari

Certificato di Eccellenza



Tutte le foto (555)

### Punteggi e recensioni

5,0 620 recensioni

N. 1 di 141 Italiana a Frascati  
N. 1 di 166 Ristoranti a Frascati

Certificato di Eccellenza Vincitore 2019

PUNTEGGI	
Cucina	5.0
Servizio	5.0
Qualità/prezzo	4.5

### Cibo e atmosfera

Cucina locale, Italiana, Steakhouse, Mediterranea, Romana, Laziale

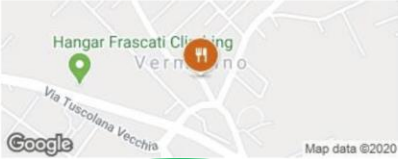
... con faraona ripiena di porcini **filetto** al pistacchio e tagliata al barolo...

... ricette sorprendenti dall'**antipasto** al dolce, ti lasciano la voglia di to...

... vi alzerete soddisfatti del loro **filetto** in tutte le salse oppure di un b...

Ho anche potuto assaggiare una forchettata della gustosissima **cacio e pepe**, s...

### Località e contatti



Via Di Passo Lombardo 13/15, 00133 Frascati Italia

Sito web E-mail

+39 06 7265 0935

Migliora questo profilo

what else?



## Recensioni (620)

Scrivi una recensione

### Valutazione



### Tipo di viaggiatore

- ☐ Famiglie
- ☐ In coppia
- ☐ Da solo
- ☐ Affari
- ☐ Amici

### Periodo dell'anno

- ☐ Mar-Mag
- ☐ Giu-Ago
- ☐ Set-Nov
- ☐ Dic-Feb

### Lingua

- ☐ Tutte le lingue
- ☒ Italiano (609)
- ☐ Inglese (7)
- ☐ Francese (1)
- Altre lingue ▾

### Scopri i commenti dei viaggiatori:



Cerca recensioni



aqmanu  
1 recensione



Recensito il 12 marzo 2020 ☐ da dispositivo mobile

### Assolutamente Sì!

Locale molto bello ed accogliente, personale gentile e disponibile...i piatti si presentano molto bene e la cucina è di qualità (tutto molto equilibrato e curato), ottima anche la scelta dei vini...parcheeggio privato... Consigliato!

Data della visita: febbraio 2020

Utile?



1

what else?



Recensito il 12 marzo 2020 ☐ da dispositivo mobile

### UNA CENA INDIMENTICABILE



Recensito il 12 marzo 2020  da dispositivo mobile

## UNA CENA INDIMENTICABILE

cena con la famiglia in un ambiente caldo e accogliente dove le proprietarie Anna e Beatrice ti accolgono facendoti sentire a casa. E poi si inizia con il menù che offre una accurata selezione di piatti a cominciare dagli antipasti preparati con tanta cura che... Più



Recensito il 12 marzo 2020  da dispositivo mobile

## UNA CENA INDIMENTICABILE

cena con la famiglia in un ambiente caldo e accogliente dove le proprietarie Anna e Beatrice ti accolgono facendoti sentire a casa. E poi si inizia con il menù che offre una accurata selezione di piatti a cominciare dagli antipasti preparati con tanta cura che è un peccato mangiarli. si continua con la pasta fatta in casa direttamente dallo chef Davide, il quale per mio figlio ha preparato delle pappardelle al ragù di cervo mai gustate così buone nemmeno in trentino. Per finire la tagliata di altissima qualità. Sul finire della serata bellissima sorpresa la musica dal vivo suonata da DJ GIGITHEFOX che in console è sempre una garanzia.

**Mostra meno**

.....

```
▼<p class="partial_entry">  
  "cena con la famiglia in un ambiente caldo e accogliente dove le proprietarie Anna e Beatrice ti accolgono  
  accurata selezione di piatti a cominciare dagli antipasti preparati con tanta cura che..."  
  <span class="taLnk ulBlueLinks" onclick="widgetEvCall('handlers.clickExpand',event,this);">Più</span>  
  ,
```

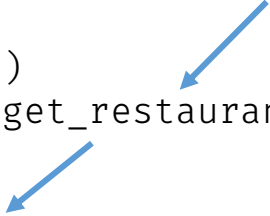
- Analisi - 1
  - Caricare la prima pagina web
    - TripAdvisor mostra 30 ristoranti per pagina
  - Verificare se esiste una prossima pagina
    - Bottone Avanti in fondo alla pagina
      - Se esiste prelevare il link
      - Altrimenti siamo all'ultima pagina
  - Caricare tutti i box dei ristoranti nelle macro scatole
    - `div class="_1lICuDZj"`
    - Escludendo gli sponsorizzati
      - Presenza della scatola
        - `div class="_376lhJeB"`

- Analisi – 2

- Prelevare il link di ogni ristorante
  - `<a href="/Restaurant_Review-g229462-d14188599-Reviews-FiloLogico_Restaurant-Frascati_Province_of_Rome_Lazio.html" class="_15_ydu6b" target="_blank">`
- Aprire la pagina del ristorante (!!!)
- Prelevare le informazioni selezionate
  - Nome
    - `<h1 class="restaurants-detail-top-info-TopInfo__restaurantName--1KBe ui-header h1">`
  - Totale recensioni
    - `<a class="restaurants-detail-overview-cards-RatingsOverviewCard__ratingCount--DFxkG" href="#REVIEWS">620 recensioni</a>`
  - Rating
    - `<span class="restaurants-detail-overview-cards-RatingsOverviewCard__overallRating--nohTI">5,0<!-- -->&nbsp;</span>`
  - Non più di 10 recensioni
    - `<p class="partial_entry">Piacetvolissima esperienza...`

- Caricare pagina web

```
def openCityRestaurants(url):  
    while url is not None:  
        page = get_page(url)  
        rests, next_page = get_restaurants(page)  
  
        for rest in rests:  
            print(get_reviews(rest))  
  
        if next_page is None:  
            url = None  
        else:  
            url = BASE + next_page.attrs["href"]
```




```
if __name__ == "__main__":  
    BASE = "https://www.tripadvisor.it/"  
    city = "Restaurants-g229462-Frascati_Province_of_Rome_Lazio.html"  
    openCityRestaurants(BASE + city)
```

```
from bs4 import BeautifulSoup

def get_restaurants(html):
    soup = BeautifulSoup(html, "html.parser")
    next_page = soup.find("a", {"class": "next"})
    restaurants = soup.find_all("div", {"class": "_1llCuDZj"})
    for restaurant in restaurants:
        if restaurant.find("div", {"class": "_376lhJeB"}):
            restaurants.remove(restaurant)
    return restaurants, next_page
```

```
def get_reviews(rest):
    a = rest.find("a", {"class": "_15_ydu6b"})
    rest = open_rest(a.attrs["href"])
    return rest
```





```
def open_rest(url):  
    text = get_reviews_page(BASE + url)  
    soup = BeautifulSoup(text, "html.parser")  
  
    name = get_name(soup)  
    nrev = get_nrev(soup)  
    rating = get_rating(soup)  
    reviews_text = get_up_to_10_reviews(soup)  
  
    # and many others  
  
    return {"name": name,  
            "reviews_count": nrev,  
            "rating": rating,  
            "ten_reviews_text": reviews_text}
```