

Realizzazione di un Web Server minimale in Python e pubblicazione di un sito statico

Questo progetto ha lo scopo di realizzare un semplice **web server HTTP** locale utilizzando Python e il modulo **socket**. Il server è in grado di rispondere a richieste HTTP di tipo **GET** per servire file HTML e altri contenuti statici (immagini, CSS, ecc.) presenti in una directory locale.

Nel file [server.py](#), le funzionalità principali sono le seguenti:

- avvia un server TCP che ascolta sulla porta 8080 (HOST = 127.0.0.1);
- gestisce connessioni in entrata e analizza le richieste HTTP;
- restituisce file statici richiesti dai client (browser), come pagine HTML o immagini;
- mostra nel terminale un log con l'orario, il metodo HTTP, la risorsa richiesta e lo stato (200 o 404).

Se la richiesta è valida e il file esiste, invia il contenuto con intestazione **HTTP 200 OK**; nel caso in cui il file non esiste, viene invece inviata una pagina HTML con errore **404 NOT FOUND**.

Di seguito viene mostrato il funzionamento:

- avvio del server

```
Python 3.12.3 | packaged by conda-forge | (main, Apr 15 2024, 18:20:11) [MSC v.1938 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 8.27.0 -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/crist/Desktop/reti/elaborato/server.py', wdir='C:/Users/crist/Desktop/reti/elaborato')
Server in ascolto su http://127.0.0.1:8080
```

All'accensione del server viene creato un **socket TCP**, successivamente il socket viene associato a **localhost** sulla porta **8080**.

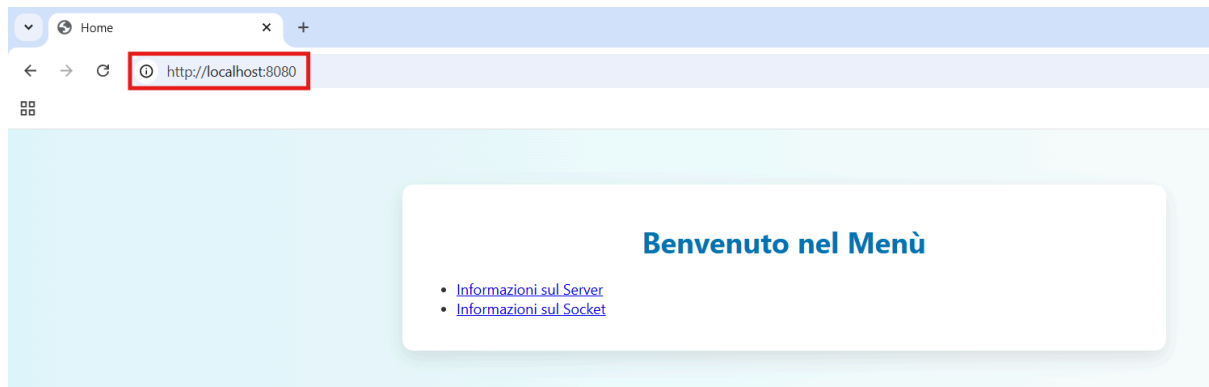
Il server inizia ad ascoltare connessioni in arrivo (`listen(5)`).

Successivamente viene stampato sul terminale:

"Server in ascolto su <http://127.0.0.1:8080>"

N.b. → il file **index.html**, ossia nel nostro caso la homepage, è la pagina principale del sito, caricata quando si visita <http://127.0.0.1:8080/>.

- visita <http://localhost:8080> nel browser



Di seguito il risultato prodotto dal server python:

```
Python 3.12.3 | packaged by conda-forge | (main, Apr 15 2024, 18:20:11) [MSC v.1938 64
bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 8.27.0 -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/crist/Desktop/reti/elaborato/server.py', wdir='C:/Users/
crist/Desktop/reti/elaborato')
Server in ascolto su http://127.0.0.1:8080
[2025-05-19 22:11:21] GET /index.html -> 200
[2025-05-19 22:11:21] GET /style.css -> 200
[2025-05-19 22:11:21] GET /favicon.ico -> 404
```

Dall'immagine possiamo notare che quando visito <http://localhost:8080> nel browser, il browser stesso invia istruzione:

GET /index.html e i relativi file che vengono richiesti.

Se il file esiste:

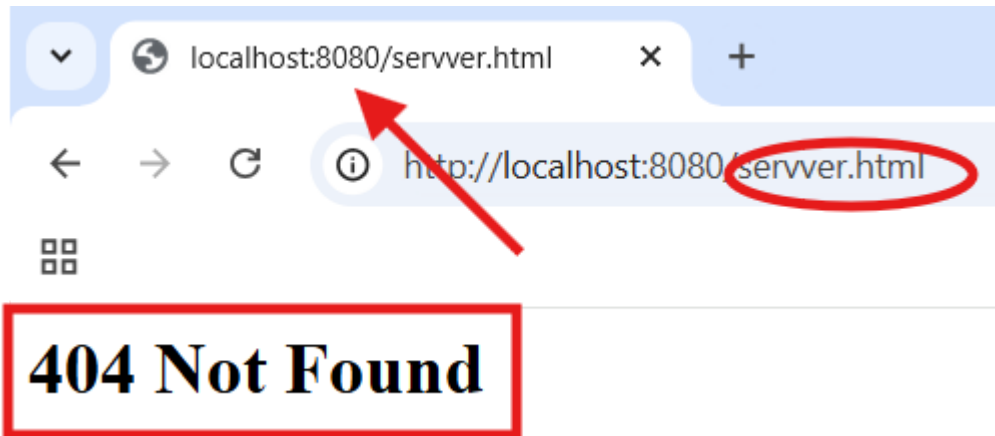
- determina il tipo di MIME (es. text/html);
- costruisce una risposta **HTTP 200 OK**;
- invia il contenuto al browser.

Se il file non esiste invece invia una risposta **404 Not Found** con messaggio HTML.

Dopodiché il server **chiude il socket client** dopo aver risposto, per poi ritornare in ascolto per nuove connessioni.

Caso 404 Not Found

Mostriamo il caso di una pagina che non viene trovata, perché di fatto non esiste. Per farlo modifichiamo il nome del file html, mettendolo appositamente sbagliato nell'url durante la ricerca nel browser.



```
In [1]: runfile('C:/Users/crist/Desktop/reti/elaborato/server.py', wdir='C:/Users/crist/Desktop/reti/elaborato')
Server in ascolto su http://127.0.0.1:8080
[2025-05-19 22:11:21] GET /index.html -> 200
[2025-05-19 22:11:21] GET /style.css -> 200
[2025-05-19 22:11:21] GET /favicon.ico -> 404
[2025-05-19 22:21:23] GET /server.html -> 200
[2025-05-19 22:21:24] GET /style.css -> 200
[2025-05-19 22:21:24] GET /img/server.png -> 200
[2025-05-19 22:21:29] GET /servver.html -> 404
```

Ovviamente ci viene restituito un errore, in quanto la pagina "servver.html" non fa parte del nostro sito statico.

Tecnologie usate per la realizzazione dell'elaborato:

- Python (socket, mimetypes)
- HTML e CSS per il sito statico
- Spyder per realizzare [server.py](#) e visualizzare i risultati prodotti sul terminale
- Visual Studio Code per la realizzazione dei file HTML e CSS