



BERGAMO **FAB**ULOUS **LAB**ORATORY

Python in pratica II

Pascal Brunot

09/01/2026

FabLab Bergamo

Regole del corso



ORARI

15,30 – 16,30
pausa 15 minuti
16.45 – 18.00

Le domande sono molto gradite, a qualunque momento. Alzate la mano se non vi noto.

Usate il PC e fare gli esercizi/quizz man mano, la pratica è fondamentale per imparare.

Piano

Lezione 2

- ❖ Review lezione 1
 - ❖ Variabili, Espressioni, Istruzioni di controllo if/while
- ❖ Elementi di linguaggio Python
 - ❖ Funzioni, Range
 - ❖ Ciclo FOR
- ❖ Liste e dizionari
- ❖ Usare librerie Python
 - ❖ Fare richieste API WEB
 - ❖ Manipolare immagini
 - ❖ Creare un oggetto 3D

Recap

```
nome = "Alex"
```

```
età = 25
```

```
prezzo = 19.99
```

```
attivo = True
```

Colab

<https://colab.research.google.com/drive/1ptdQNCBAFVPvFmXRD9co32-WBo0I-oRi?usp=sharing>

Funzioni

- ❖ Un blocco di codice che svolge un compito specifico
 - ❖ Puoi usarla più volte senza riscrivere il codice
 - ❖ Rende il codice più organizzato e pulito
 - ❖ Facilita la correzione degli errori



Esempi

- ❖ Funzione che non ritorna valori

```
def saluta(nome):  
    print(f"Ciao {nome} !")  
  
saluta("Alex")  
saluta("Isabelle")
```

- ❖ Funzione che ritorna valori

```
def applica_sconto(prezzo, percentuale):  
    sconto = prezzo * percentuale / 100  
    prezzo_finale = prezzo - sconto  
    return prezzo_finale  
  
prezzo_maglietta = 50  
nuovo_prezzo =  
applica_sconto(prezzo_maglietta, 20)  
print(nuovo_prezzo) #Stampa: 40.0
```

Range

◆ RANGE(A)

- ◆ I numeri interi fra 0 e A escluso

RANGE(10) = [0,1,2,3,4,5,6,7,8,9]

Sono 10 elementi

◆ RANGE(A,B)

- ◆ I numeri interi fra A (incluso) e B (escluso)

RANGE(1,10) = [1,2,3,4,5,6,7,8,9]

Sono 9 elementi !

◆ RANGE(A, B, C)

- ◆ I numeri interi fra A (incluso) e B (escluso) a passo di C

◆ RANGE(5,51,5) = la tabella del cinque

[5,10,15,20,25,30,35,40,45,50]

Provate in Colab:

```
list(range(5))
```

```
list(range(1,20))
```

```
list(range(1,51, 5))
```

Range - domande



Se voglio i primi 100 numeri partendo da 0, cosa scrivo ?



Se voglio i primi 100 numeri dispari partendo da 1, cosa scrivo ?

Istruzione For

•
For consente di
ENUMERARE (ITERARE)
na sequenza di oggetti : una
lista, un range...

Esempio

```
frutti = ["mela", "banana",  
"pesca"]  
  
for frutto in frutti:  
    print(f"mangio una {frutto}")
```

Spiegazione

- Dichiaraione di una variabile **frutti** di tipo lista con 3 elementi.

- **for frutto in frutti:** prendi ogni elemento da **frutti**

- **frutto** = la variabile che cambia ad ogni iterazione

- Stampa: mangio una mela → mangio una banana → mangio una pesca

Esercizio con ciclo for

Stampare i numeri interi fra 1 e 10 inclusi.

- Output atteso:

1
2
3
...
10

Stampare i numeri pari fra 0 e 20 inclusi.

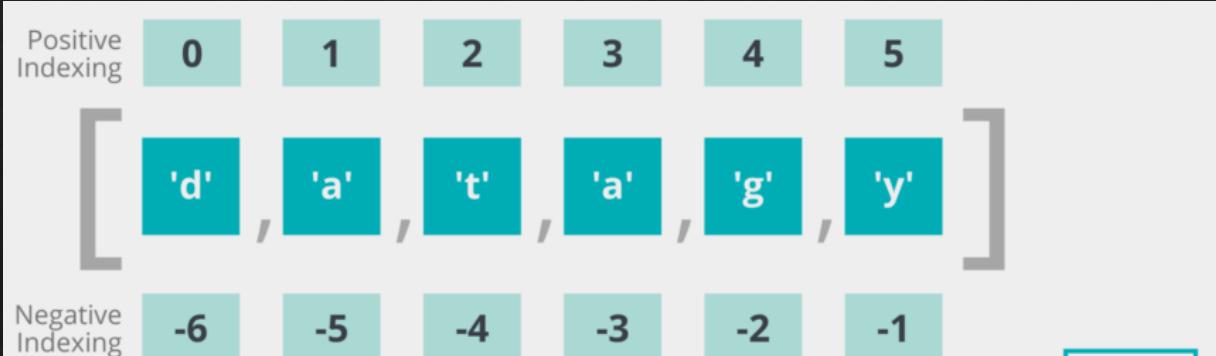
- Output atteso:

0
2
4
...
20

Liste

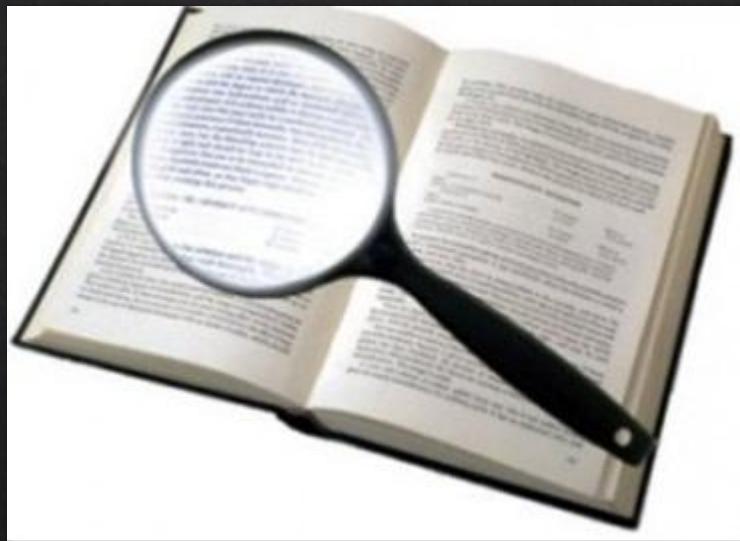
- ❖ Parentesi quadre
- ❖ Accesso agli elementi con indice intero
`l[1]` , `l[-1]`

```
# Lo zaino del videogiocatore
zaino = ["spada", "pozione", "mappa",
"torcia"]
print(zaino[0]) # spada - il primo oggetto!
```



Dizionari

- ❖ Parentesi graffe { } (ALTGR SHIFT [])
- ❖ Accesso agli elementi con la loro **chiave**



```
# Rubrica  
contatti = {  
    "pizzeria": "035-123456",  
    "mamma": "333-1234567",  
    "meccanico": "349-7654321"  
}  
  
# Estriamo il contatto di emergenza  
print(contatti["pizzeria"]) # Emergenza pizza!
```

Dizionari

- ❖ Un dizionario è un insieme di **valori** indicizzati da **chiavi** univoche

```
dizionario = {  
    chiave1: valore1,  
    chiave2: valore2,  
    ...  
}
```

- ❖ Non posso avere chiavi **duplicate** nel dizionario.

- ❖ Cancellare un'entrata del dizionario
`del(contatti["pizzeria"])`

- ❖ Aggiungere un elemento
`contatti["papa"] = "3311234564"`

- ❖ Enumerare le entrate
`for nome in contatti:
 print(f"{nome} = {contatti[nome]}")`

- ❖ Enumerare entrate e valori
`for nome, telefono in contatti.values():
 print(f"{nome} : {telefono}")`

Dizionari di liste



```
# Gli ingredienti per ricetta
ricette = {
    "carbonara": ["pasta", "uova", "guanciale", "pecorino"],
    "margherita": ["farina", "pomodoro", "mozzarella", "basilico"],
    "tiramisù": ["savoiardi", "caffè", "mascarpone", "cacao"]
}

print(f"Per la carbonara serve: {ricette['carbonara']}")
```

Le librerie Python



Estendono il linguaggio con molte operazioni oltre quelle che abbiamo visto fin'ora

Rendono semplici operazioni molto elaborate

Cos'è un libreria?

- ❖ Un pacchetto già pronto per realizzare operazioni specifiche
- ❖ Una libreria aggiunge «parole» e funzionalità a Python.
- ❖ Come molti linguaggi moderni Python ha un gestore delle librerie integrato (pip).
 - ❖ Concetto non unico a Python, esistono in altri linguaggi (cargo/Rust, nuget/dotnet)
 - ❖ Esistono gestori di librerie più performanti come uv, ma noi stiamo a livello base.
- ❖ Si installano in Colab con un istruzione di riga di comando (inizia con !)
- ❖ Ad esempio per installare la libreria pandas:
`!pip install pandas`

Usare librerie Python

Trovare

Chiedere all'AI («proponi una libreria Python semplice per fare <....> con un esempio di installazione e uso per farmi capire le funzioni chiave»)

Alcune sono molto molto diffuse e li vedremo nel corso : pandas, numpy, PIL,...

Installare

```
!pip install <nome>
```

PIP è un gestore di pacchetti che scarica la libreria su Internet e la installa per Python.

Importare nel programma

```
import <libreria>
```

```
import numpy as np
```

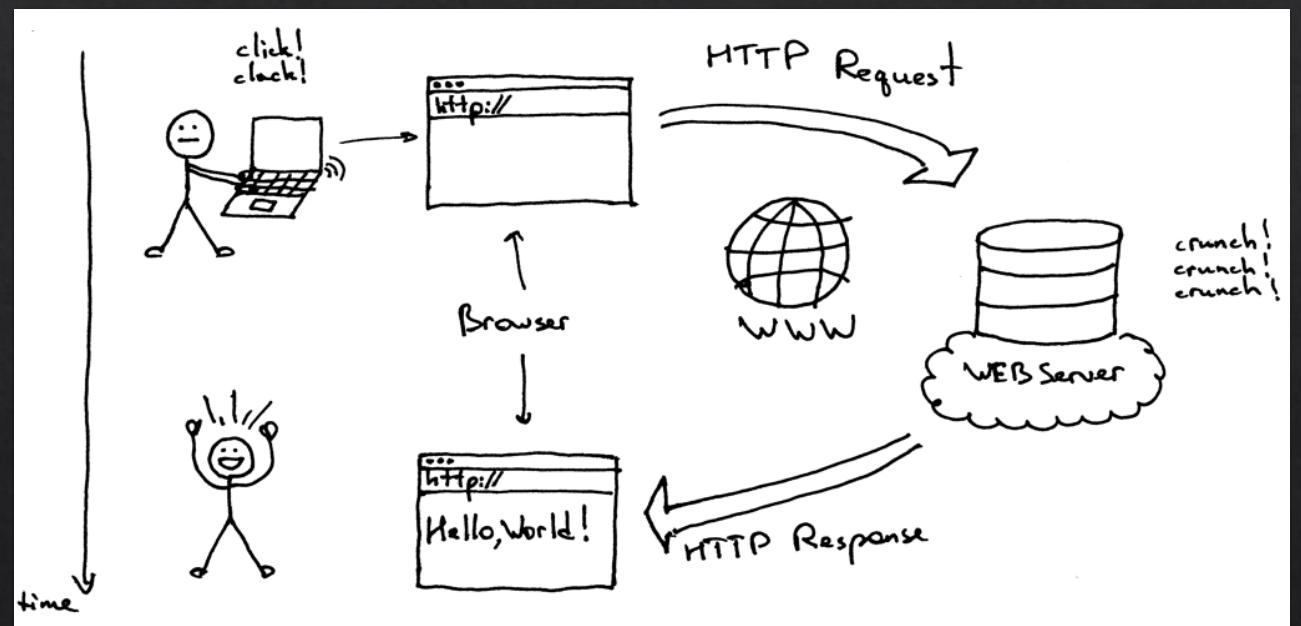
Import parziale

```
from PIL import Image  
import matplotlib.pyplot as plt
```

1° libreria : Requests

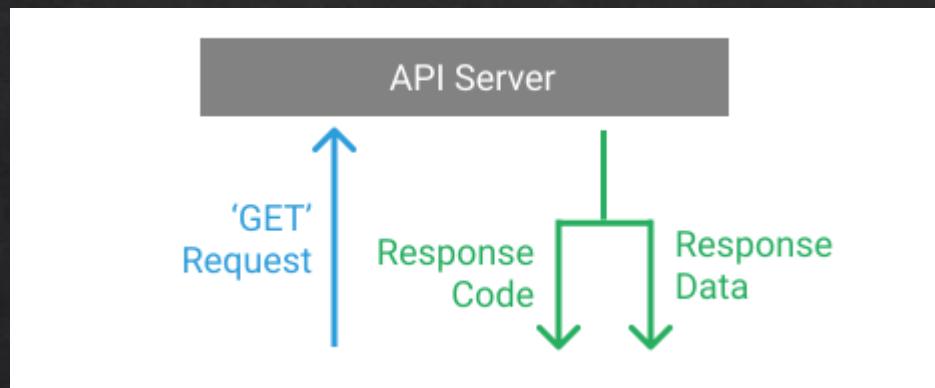
- ❖ Serve a mandare una richiesta ad un servizio http su Internet.
- ❖ E' già installata in Colab
 - ❖ non serve !pip install requests

```
import requests  
response = requests.get(url=..., params={})
```



API e Request

- ❖ Principio di REQUESTS



- ❖ API SERVER

- ❖ METEO

- ❖ Esempi di API Server

- ❖ LLM : ChatGPT, Claude, ...
 - ❖ Google search
 - ❖ Tassi di cambio
 - ❖ Meteo
 - ❖ Text-to-voice
 - ❖ Voice-to-text

Esercizio

- ❖ Trovate un servizio web che ritorna immagini di gatti.
- ❖ Usando requests, mandare una richiesta a un servizio, recuperare l'indirizzo URL dell'immagine del gatto nella risposta, e farlo visualizzare con display()

```
import requests  
from IPython.display import Image, display  
  
## IL VOSTRO CODICE QUA  
  
# Mostrare l'immagine direttamente  
display(Image(url=url_immagine))
```

Altre librerie

PILLOW

Trattamento di immagine

Ridimensionamento

Modifiche immaginei

Altre Librerie

Trimesh

Generare delle forme ed esportare la forma come file STL / GLB