

Cos'è Python?

- Linguaggio di programmazione **interpretato** e **ad alto livello**
- **General-purpose**: usato per web, data science, AI, scripting e altro
- **Sintassi semplice** e leggibile, perfetto per principianti

Vantaggi principali:

- ✓ Facile da imparare
- ✓ Ampia collezione di librerie
- ✓ Comunità attiva e supporto
- ✓ Multiplatforma (Windows, macOS, Linux)

Dove si usa?

- Sviluppo web (Django, Flask)
- Data Science e Machine Learning (Pandas, TensorFlow)
- Automazione e scripting
- Applicazioni scientifiche

2. STRUTTURA DI UN PROGRAMMA PYTHON

Elementi fondamentali:

1. **Commenti**: # commento o """ commento multilinea """
2. **Variabili**: Non serve dichiarare il tipo
3. **Funzioni**: def nome_funzione():
4. **Import**: import modulo

Esempio concettuale:

Copy

Download

1. Definisci una funzione che stampa un saluto
2. Chiama la funzione con un nome
3. Usa una libreria per operazioni matematiche

3. VARIABILI E TIPI DI DATO

Tipi fondamentali:

Tipo	Esempio	Descrizione
int	42, -7	Numeri interi
float	3.14, -0.1	Numeri decimali
str	"ciao"	Stringhe di testo
bool	True, False	Valori booleani
list	[1, 2, 3]	Liste modificabili
tuple	(1, 2, 3)	Liste immutabili
dict	{"nome": "Mario"}	Dizionari

Dichiarazione:

python

Copy

Download

nome = "Alice" # Stringa

eta = 25 # Intero

altezza = 1.75 # Float

attivo = True # Booleano

4. OPERATORI

Aritmetici:

- +, -, *, /, // (divisione intera), % (modulo), ** (potenza)

Comparazione:

- ==, !=, >, <, >=, <=

Logici:

- and, or, not

5. STRUTTURE DI CONTROLLO

Condizionali:

python

Copy

Download

if x > 0:

 print("Positivo")

elif x == 0:

 print("Zero")

else:

 print("Negativo")

Cicli:

python

Copy

Download

For loop

for i in range(5):

 print(i)

While loop

contatore = 0

while contatore < 5:

 print(contatore)

 contatore += 1

6. FUNZIONI

Definizione:

python

Copy

Download

```
def saluta(nome):  
    return f"Ciao, {nome}!"
```

Chiamata:

python

Copy

Download

```
messaggio = saluta("Mario")  
print(messaggio) # Output: Ciao, Mario!
```

7. STRUTTURE DATI AVANZATE

Liste:

python

Copy

Download

```
numeri = [1, 2, 3]  
numeri.append(4) # Aggiunge elemento
```

Dizionari:

python

Copy

Download

```
persona = {  
    "nome": "Luigi",  
    "età": 30,  
    "città": "Roma"  
}  
  
print(persona["nome"]) # Accesso al valore
```

8. OOP IN PYTHON

Classi e Oggetti:

python

Copy

Download

class Persona:

```
def __init__(self, nome, eta):  
    self.nome = nome  
    self.eta = eta  
def saluta(self):  
    print(f"Ciao, sono {self.nome}")
```

Creazione oggetto

p1 = Persona("Anna", 25)

p1.saluta()

9. GESTIONE DEI FILE

Lettura:

python

Copy

Download

with open("file.txt", "r") as file:

```
    contenuto = file.read()
```

Scrittura:

python

Copy

Download

with open("file.txt", "w") as file:

```
    file.write("Ciao Mondo!")
```

10. ERRORI COMUNI

✗ Dimenticare i : dopo if/for/def

✗ Errori di indentazione

- ✗ Confondere = con ==
- ✗ Modificare una lista durante l'iterazione

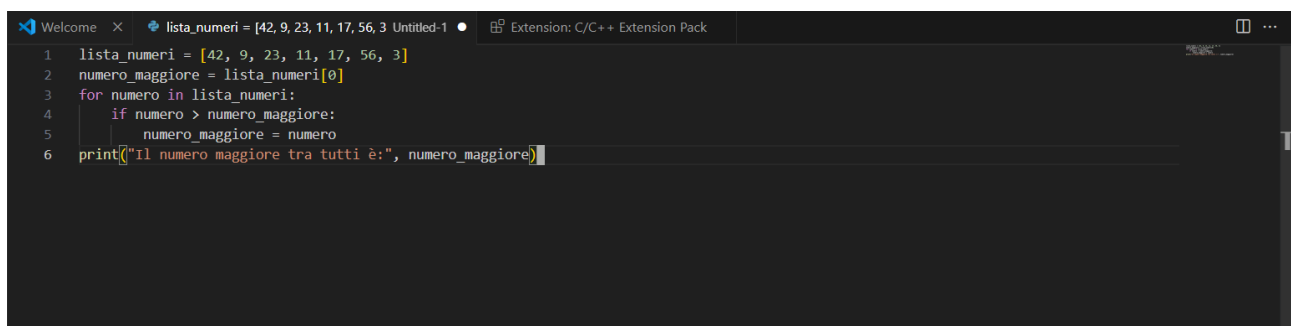
11. BEST PRACTICES

- ✓ Usa nomi descrittivi per variabili e funzioni
- ✓ Mantieni le righe sotto 79 caratteri (PEP 8)
- ✓ Scrivi docstring per documentare le funzioni
- ✓ Usa i context manager (with) per i file

12. DOVE ANDARE ORA?

1. **Esercitati** con progetti semplici
2. **Esplora** librerie popolari:
 - NumPy per calcoli scientifici
 - Pandas per data analysis
 - Requests per HTTP
3. **Approfondisci**:
 - Decoratori
 - Generatori
 - Programmazione asincrona

Ecco un esempio di python:



```
1 lista_numeri = [42, 9, 23, 11, 17, 56, 3]
2 numero_maggiore = lista_numeri[0]
3 for numero in lista_numeri:
4     if numero > numero_maggiore:
5         numero_maggiore = numero
6 print("Il numero maggiore tra tutti è:", numero_maggiore)
```