



Python e l'Open Source

... la rivoluzione della condivisione

 III Liceo Scientifico Biella - Scienze Applicate

 Python Biella Group



Sfida per iniziare

*Cosa, se condivisa, si moltiplica
invece di diminuire?*



Sfida per iniziare

*Cosa, se condivisa, si moltiplica
invece di diminuire?*

La CONOSCENZA!  

Ed è proprio su questo principio che si basa tutto il movimento Open Source



Python Biella Group

"C'è sempre qualcosa da imparare per migliorarci e crescere... insieme!"

Una community (g)local:

- ❖ Inclusivi, raggiungibili e utili - aperti a tutti, dal principiante all'esperto
- 🤝 Aperti alla condivisione e al confronto - ogni idea è benvenuta
- 🎭 Rappresentativi - diverse esperienze e background arricchiscono il gruppo
- 🔬 Innovatori e sperimentatori - non abbiamo paura di provare cose nuove

(G)local = Globali nella visione, Locali nell'azione   



Che cos'è l'Open Source?

Open Source = codice sorgente accessibile e modificabile da chiunque

🔍 Non è solo una licenza software, è una **filosofia**:

💬 Scambio aperto e trasparenza

🤝 Partecipazione collettiva

🏆 Meritocrazia

🌱 Sviluppo della comunità

"Se ho visto più lontano è perché stavo sulle spalle di giganti"

— Isaac Newton (il primo open source developer! 😊)*



FREE

I pilastri dell'Open Source

Libertà fondamentali:

-  **Ridistribuzione libera** - copia, vendi, cedi senza royalties
-  **Codice sorgente incluso** - necessario per modificare e migliorare
-  **Nessuna restrizione** - su come e dove usare il software
-  **Modifiche condivise** - i miglioramenti ritornano alla comunità



Esempi di Open Source

💡 **GNU/Linux** - il sistema operativo che fa girare Internet

🌐 **Mozilla Firefox** - il browser della libertà

📊 **LibreOffice** (ex OpenOffice) - alternativa gratuita a Microsoft Office

🤖 **Android** - il sistema mobile più diffuso al mondo

▀ **VLC Media Player** - riproduce TUTTO

💡 E naturalmente... **Python!** 🐍

Progetti avviati da "persone normali" che avrebbero potuto vendere e arricchirsi subito.

Loro hanno regalato il loro lavoro al mondo... 🌎

e costruito carriere e successi che nessuna vendita avrebbe mai potuto garantire.



Caratteristiche di Python

- 🔒 **Open-source** (*non proprietario*) - codice libero e modificabile
- ⬆️ **High-level** (*non Low-level*) - più vicino al linguaggio umano
- ▶️ **Interpretato** (*non compilato*) - esegui subito, senza compilazione
- 🎯 **Multi-paradigma** - scripting, OOP, programmazione funzionale
- 🌐 **Portabile** (*non platform-dependent*) - scrivi una volta, esegui ovunque
- ⚡ **Estensibile ed embeddable** - integrabile con C/C++ e altri linguaggi



Lo Zen di Python

Digita `import this` nella console Python e appare la "filosofia" del linguaggio:

```
# The Zen of Python, by Tim Peters
```

```
Beautiful is better than ugly.  
Explicit is better than implicit.  
Simple is better than complex.  
Readability counts.  
...e altri principi di saggezza informatica
```

Non sono solo regole di programmazione, sono una filosofia di vita! 🙏



Esplicito è meglio che implicito

✗ Codice criptico - *Cosa diavolo fa questo codice?* 🤔

```
def f(x, y):  
    z = x * y * 0.22  
    return z  
r = f(150, 5)
```

✓ Codice chiaro - *Ah, calcola l'IVA! Chiaro!*💡

```
def calcola_iva(prezzo, quantita):  
    IVA = 0.22  
    totale = prezzo * quantita * IVA  
    return totale  
iva_da_pagare = calcola_iva(150, 5)
```



La leggibilità conta

Il codice si legge molte più volte di quante si scriva!

Fondamentalmente...

*Scrivete codice come se dovesse leggerlo uno psicopatico violento che sa dove abitate.
E quello psicopatico siete voi tra 6 mesi.*

⌚ Leggere codice: **100 volte**

✍️ Scrivere codice: **1 volta**

🔧 Debuggare codice illeggibile: **∞ volte (con sofferenza)**



Semplice è meglio che complesso

Java:

```
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello World!");  
    }  
}
```

Python:

```
print("Hello World!")
```



★ Semplice è meglio che complesso

```
#include <stdio.h>
int main() {
    printf("Hello World!\n");
    return 0;
}
```

Python:

```
print("Hello World!")
```

☕ Uno richiede caffeina. Uno, sacrifici. Uno, solo Python.



La storia di Python

 1989: Guido van Rossum inizia Python durante le vacanze di Natale
(programmatori gonna program)

 1991: Prima versione pubblica (34 anni fa!)

 Nome ispirato ai **Monty Python**, non al serpente!

 Oggi: uno dei linguaggi **PIÙ USATI** al mondo

Linguaggio "vecchio" ma attualissimo: come i jeans, non passa mai di moda! 



🏆 Classifica linguaggi più usati (2024-2025)

- 1 Python 
- 2 JavaScript
- 3 Java
- 4 C/C++
- 5 C#



Python è il RE indiscusso per:

AI/ML • Data Science • Ricerca • Didattica



Python nel mondo accademico

Primo linguaggio insegnato in moltissime università

Sta progressivamente **sostituendo il C** nei corsi introduttivi

💡 Perché?

- Sintassi più semplice e intuitiva
- Permette di concentrarsi sui **concetti**, non sulla sintassi
- Meno frustrante per i principianti (*addio segmentation fault!*)
- Risultati immediati = maggiore motivazione

Il C non è morto, ma Python è il nuovo "primo amore" ❤️



Python nel mondo reale

- 🏛️ **Finanza:** Trading algoritmico, analisi di rischio, blockchain
- 🔬 **Ricerca:** Bioinformatica, fisica delle particelle (CERN!), astronomia
- 🏭 **Industria:** Google, Netflix, Instagram, Spotify, NASA
- 🤖 **AI:** TensorFlow, PyTorch, scikit-learn (*tutti in Python*)
- 💻 **DevOps:** Automazione di tutto l'automatizzabile
- 👤 **Hacker etici:** Penetration testing e security

*Se l'AI sta cambiando il mondo, e l'AI gira su Python... beh, fate voi i conti! 📈



"Batteries Included"

Python viene con una **libreria standard ENORME**:

- ✉ Email • 🌐 Web • 📁 File system • 📈 Matematica • 📅 Date/ora
- 🔒 Crittografia • 📊 Database • ...

E poi ci sono le librerie esterne per tutto...

*NumPy, Pandas, Matplotlib, Django, Flask, BeautifulSoup,
Requests, Pillow, OpenCV, Pygame, Tkinter...*



"Batteries Included"

🎁 Troppe? Forse. Utili? Assolutamente, per quasi ogni problema! 🎁

⌚ C'è anche una libreria che ti fa volare (sul serio):

```
import antigravity
```



SFIDA: Trova l'intruso

Quale di questi NON è un motivo valido per usare Python?

- A) Sintassi pulita e leggibile
- B) Enorme ecosistema di librerie
- C) Performance da Formula 1
- D) Eccellente per prototipazione rapida

Spoiler: Python è fantastico, ma veloce come un F1? Ecco, no.



Come si impara a programmare?

SCRIVENDO CODICE.

Non si impara a programmare guardando tutorial.

Non si impara a programmare leggendo libri.

Non si impara a programmare facendo scrivere codice all'AI.

Si impara FACENDO e SBAGLIANDO. 

Gli errori sono i tuoi migliori maestri!   



L'AI come alleato (non sostituto)

✓ L'AI è PERMESSA per:

- 🔍 Cercare informazioni e documentazione
- 📘 Spiegare e riassumere concetti complessi
- 🐛 Identificare e spiegare errori nel codice
- 💡 Suggerire approcci alternativi

✗ L'AI NON va usata per:

- Scrivere il codice al posto tuo
- Copiare soluzioni senza capirle
- Evitare di ragionare sui problemi



Il programmatore nell'era dell'AI

Il ruolo del programmatore non scompare, si evolve:

-  **Architetto:** organizzare e strutturare soluzioni complesse
-  **Editor:** riscrivere e ottimizzare il codice
-  **Revisore:** verificare qualità e correttezza
-  **Problem solver:** risolvere problemi in modo creativo
-  **Pensatore critico:** valutare e scegliere tra alternative

*L'AI scrive codice, ma sei **TU** che devi capire se è buono!* 

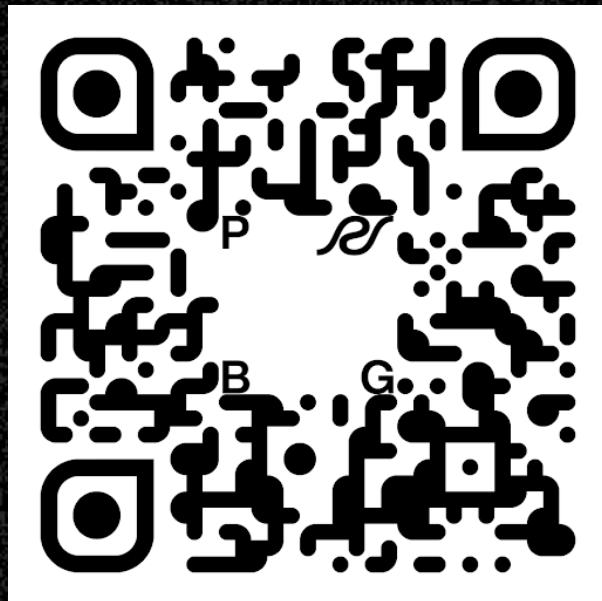


Ricapitolando...

- 🤝 La conoscenza condivisa si moltiplica
- 💻 L'open source ha cambiato il mondo
- 🐍 Python è semplice, potente e ovunque
- 📖 Scrivi codice chiaro e leggibile
- 💪 Si impara SOLO facendo (e sbagliando!)
- 🤖 L'AI è un assistente, non un sostituto

P S
BG

Grazie per l'attenzione...



"C'è sempre qualcosa da imparare per migliorarci e crescere...insieme!"