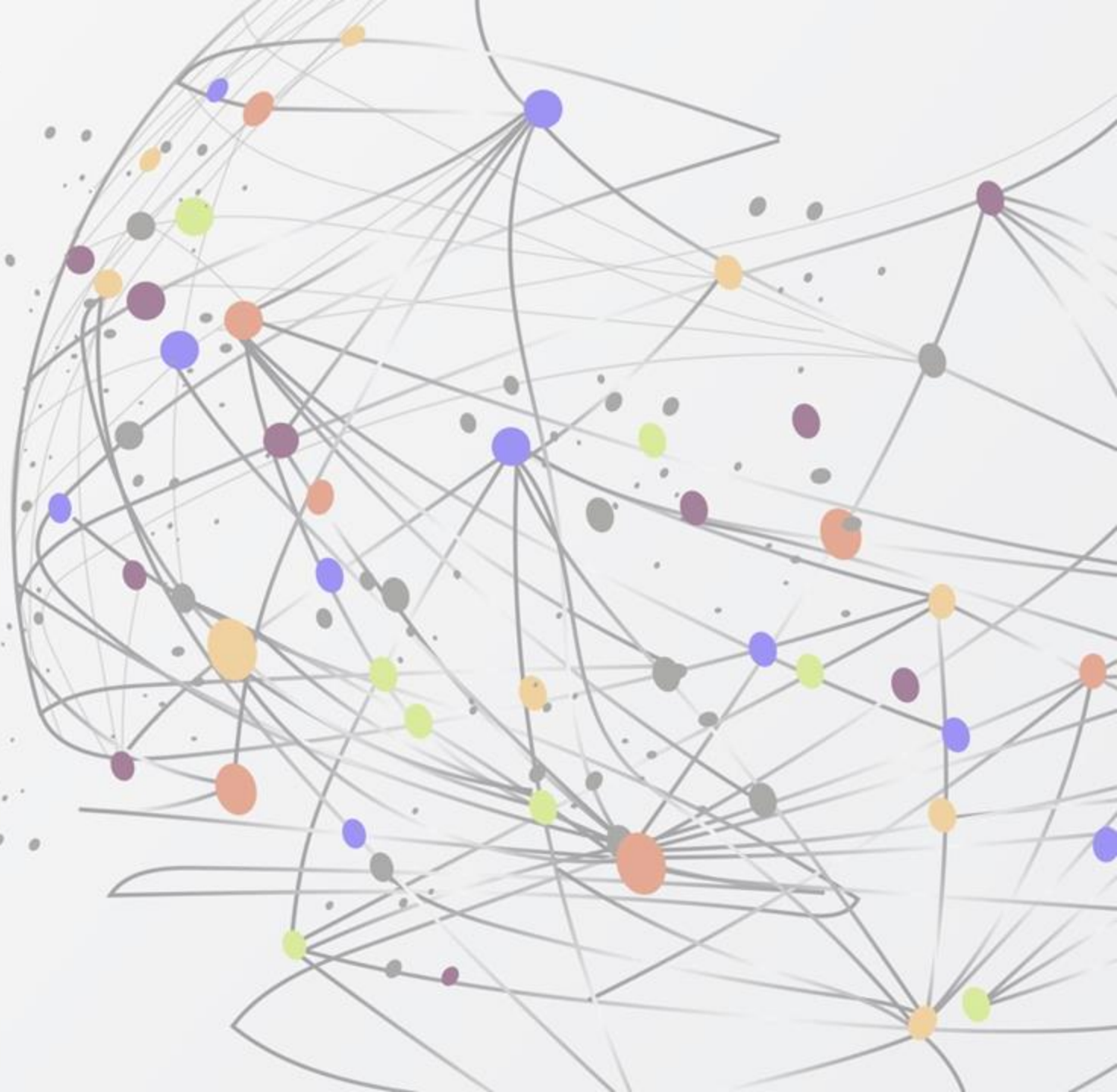


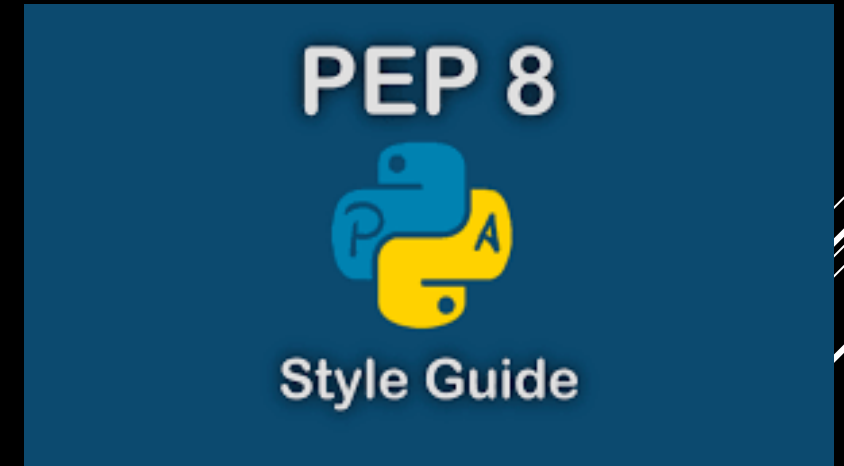
BUONE PRATICHE PER SCRIVERE UN CODICE IN PYTHON

A CURA DI MATTEO ARGENTI



PEP8

PEP 8 è l'acronimo di Python Enhancement Proposal 8. Si tratta di una proposta di miglioramento del linguaggio Python che stabilisce le regole e le raccomandazioni per scrivere codice Python pulito e uniforme. Il PEP 8 è stato scritto da Guido van Rossum, il creatore di Python, e da Barry Warsaw, uno dei principali contributori al progetto Python. PEP-8 è fondamentalmente la guida di stile ufficiale di Python. E' stato creato per migliorare la coerenza del codice scritto da diversi sviluppatori, rendendo i progetti Python più comprensibili.



<https://www.google.com/imgres?>

PERCHÉ USARE IL PEP8

Usare il PEP 8 ha molti vantaggi per noi e per gli altri sviluppatori che lavorano con il nostro codice. Alcuni di questi vantaggi sono:

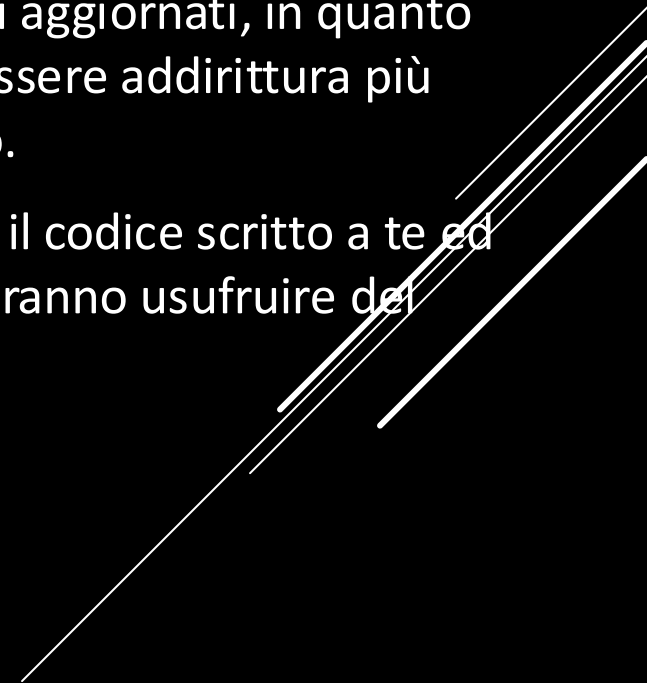
- Migliora la qualità del codice: ci aiuta a scrivere codice più chiaro, più semplice e più facile da mantenere e a evitare errori o bug.
- Migliora la collaborazione: ci aiuta a comunicare meglio con gli altri sviluppatori che usano o modificano il nostro codice e a creare una base di codice uniforme e coerente.
- Migliora la documentazione: ci aiuta a scrivere documentazione più completa e accurata per il nostro codice
- Migliora l'apprendimento: ci aiuta a imparare meglio il linguaggio Python e le sue caratteristiche e a usare al meglio la libreria standard e le librerie esterne.



I COMMENTI IN PYTHON

I commenti sono una parte fondamentale del programma in Python, poiché forniscono informazioni utili e contestuali sul codice. Tuttavia, è cruciale mantenere i commenti aggiornati, in quanto i commenti obsoleti possono essere addirittura più dannosi che non averne affatto.

I commenti servono a spiegare il codice scritto a te ed ad altri programmatori che potranno usufruire del tuo codice



TIPOLOGIE DI COMMENTI

Esistono varie tipologie di commenti che variano a seconda della loro funzione:

- **Commenti in linea:** Sono commenti di breve lunghezza che occupano una linea di codice. Vengono scritti usando il simbolo #
- **Commenti multilinea:** Sono commenti più lunghi rispetto ai primi e vengono usati in codici più complessi per la loro capacità di essere scritti su più righe di codice. Vengono scritti usando i tre apici """

```
"""
This function calculates the area of a circle.
@param radius: The radius of the circle (float or int)
@return: The area of the circle (float)
"""

def area_of_circle(radius):
    pi = 3.141592653589793
    area = pi * (radius ** 2)
    return area

circle_radius = 5
circle_area = area_of_circle(circle_radius)
print("Area of the circle:", circle_area)

Area of the circle: 78.53981633974483 ← Output
```

Using multiline line comments to explain the purpose of our program

<https://www.google.com>

INDENTAZIONE

L'**indentazione** è uno degli aspetti fondamentali per scrivere codice Python leggibile e conforme alle migliori pratiche di programmazione. In Python, l'indentazione non è solo una questione di stile, ma è parte integrante della sintassi del linguaggio, utilizzata per definire la struttura logica del codice.

```
def insert_right(a, x, lo=0, hi=None):  
    """Insert item x in list a, and keep it sorted assuming a is sorted.  
    If x is already in a, insert it to the right of the rightmost x.  
    Optional args lo (default 0) and hi (default len(a)) bound the  
    slice of a to be searched.  
    """  
    if lo < 0:  
        raise ValueError('lo must be non-negative')  
    if hi is None:  
        hi = len(a)  
    while lo < hi:  
        mid = (lo+hi)//2  
        if x < a[mid]: hi = mid  
        else: lo = mid+1  
    a.insert(lo, x)
```

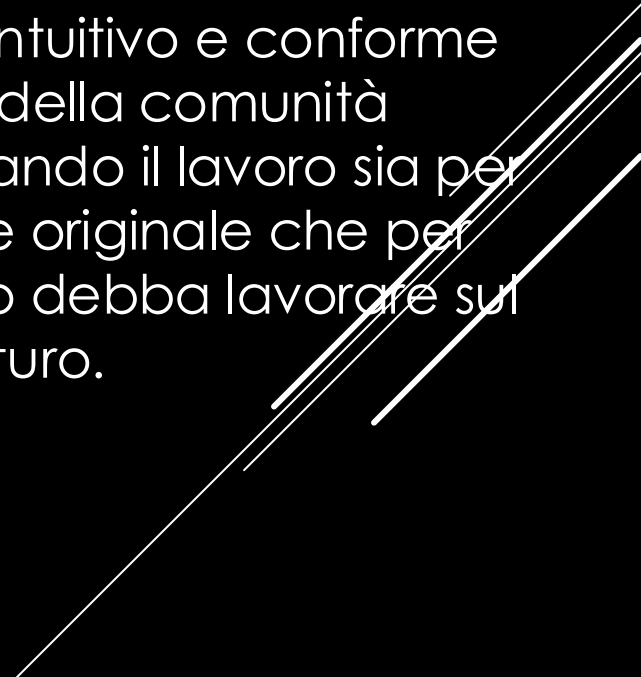
← Primo livello di indentazione

← Secondo livello di indentazione

NOMENCLATURA DELLE VARIABILI

Le convenzioni di nomenclatura sono essenziali per mantenere il codice coerente e facilmente comprensibile. Python definisce delle convenzioni stabilite per la denominazione di variabili, funzioni, classi e moduli nelle linee guida di PEP 8. I nomi dovrebbero essere autoesplicativi e descrivere in modo chiaro le informazioni o la funzionalità associata.

Seguendo queste convenzioni di nomenclatura, il codice diventa più leggibile, intuitivo e conforme agli standard della comunità Python, facilitando il lavoro sia per lo sviluppatore originale che per chiunque altro debba lavorare sul progetto in futuro.



SVILUPPO VIDEOGIOCHI

Anche nel settore dello sviluppo di giochi, Python trova la sua applicazione, principalmente nello scripting. Sebbene non sia il principale linguaggio di programmazione per lo sviluppo di giochi, la sua semplicità e flessibilità lo rendono uno strumento utile per testare idee e meccaniche di gioco in modo rapido ed efficiente.

Python viene molto utilizzato per lo sviluppo delle IA dei personaggi non giocanti e dello scripting in generale.



<https://www.google.com/imgres?>

Vantaggi di Python per lo Sviluppo di Videogiochi

L'uso di Python nello sviluppo di videogiochi è diventato sempre più comune grazie alla sua semplicità, flessibilità e alla vasta gamma di librerie e framework disponibili. Anche se Python non è generalmente scelto per lo sviluppo di giochi tripla A, è molto popolare per prototipazione, giochi educativi e script di supporto.

Facilità d'uso:

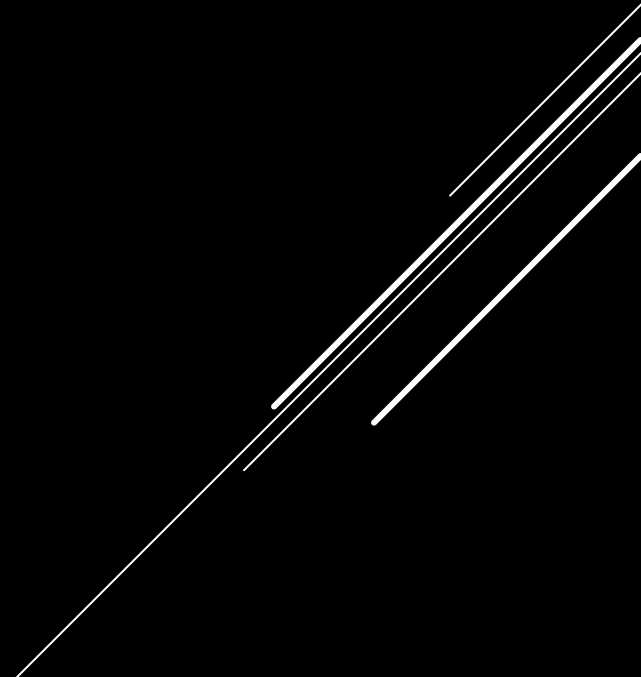
Python ha una sintassi semplice e leggibile, rendendolo ideale per chi si avvicina allo sviluppo di giochi.

•Comunità e Risorse:

Python ha una comunità attiva e una vasta documentazione, che semplifica l'apprendimento e la risoluzione dei problemi.

•Librerie e Framework Ricchi:

Esistono numerosi strumenti specializzati per lo sviluppo di videogiochi.



SITOGRAFIA

<https://www.appacademy.io/blog/python-coding-best-practices>

<https://www.python.it/doc/articoli/pep-8.html>

<https://tecnologia.webyes.com.br/a-cosa-serve-il-pep-8-in-python/>

Appunti e conoscenze sull'argomento

<https://bitcamp.it/blog/corsi-python-blog-apprendimento-linguaggio-di-programmazione/dove-si-utilizza-python/>



GRAZIE PER L'ATTENZIONE

MATTEO ARGENTI

