

Introduzione a Python

Benvenuti nel mondo della programmazione con Python. Questo corso vi guiderà passo dopo passo.

Imparerete le basi fondamentali per iniziare a programmare con sicurezza.

Obiettivi della Lezione

1 Comprendere Python

Scoprire cos'è Python e i suoi vantaggi principali.

3 Scrivere il primo codice

Creare ed eseguire il vostro primo script Python.

2 Configurare l'ambiente

Installare Python e configurare gli strumenti necessari.

4 Apprendere le basi

Conoscere sintassi, convenzioni e buone pratiche.

Cos'è Python

Linguaggio di Programmazione

Python è un linguaggio semplice e potente. Facile da leggere e imparare.

Ideale per principianti ma usato anche dai professionisti.

Perché Scegliere Python

- Sintassi chiara e intuitiva
- Vasta libreria standard
- Comunità attiva e supportiva
- Versatile per molti progetti

Storia e Comunità

1

1991 - Nascita

Guido van Rossum crea Python nei Paesi Bassi. Nome ispirato ai Monty Python.

2

2000 - Python 2

Rilascio di Python 2.0 con nuove funzionalità. Diventa più popolare.

3

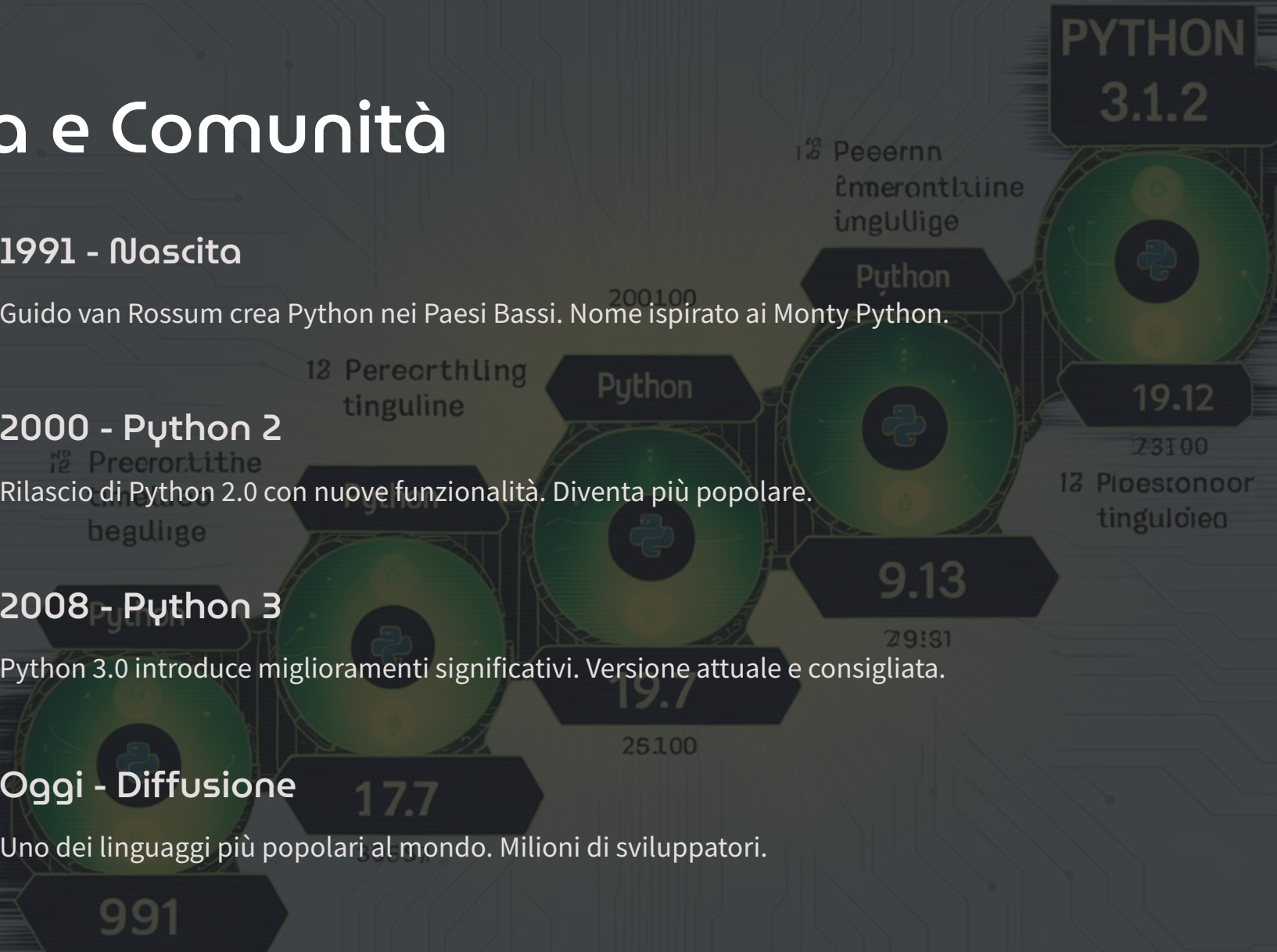
2008 - Python 3

Python 3.0 introduce miglioramenti significativi. Versione attuale e consigliata.

4

Oggi - Diffusione

Uno dei linguaggi più popolari al mondo. Milioni di sviluppatori.



Installazione di Python

Windows

Scaricate da python.org. Eseguite l'installer e selezionate "Add to PATH".

macOS

Usate Homebrew o scaricate da python.org. Terminal: `brew install python`.

Linux

Usate il package manager. Ubuntu: `sudo apt install python3`.



Il REPL di Python

Cos'è il REPL

Read-Eval-Print Loop. Ambiente interattivo per testare codice Python.

Perfetto per esperimenti e apprendimento rapido.

Come Accedere

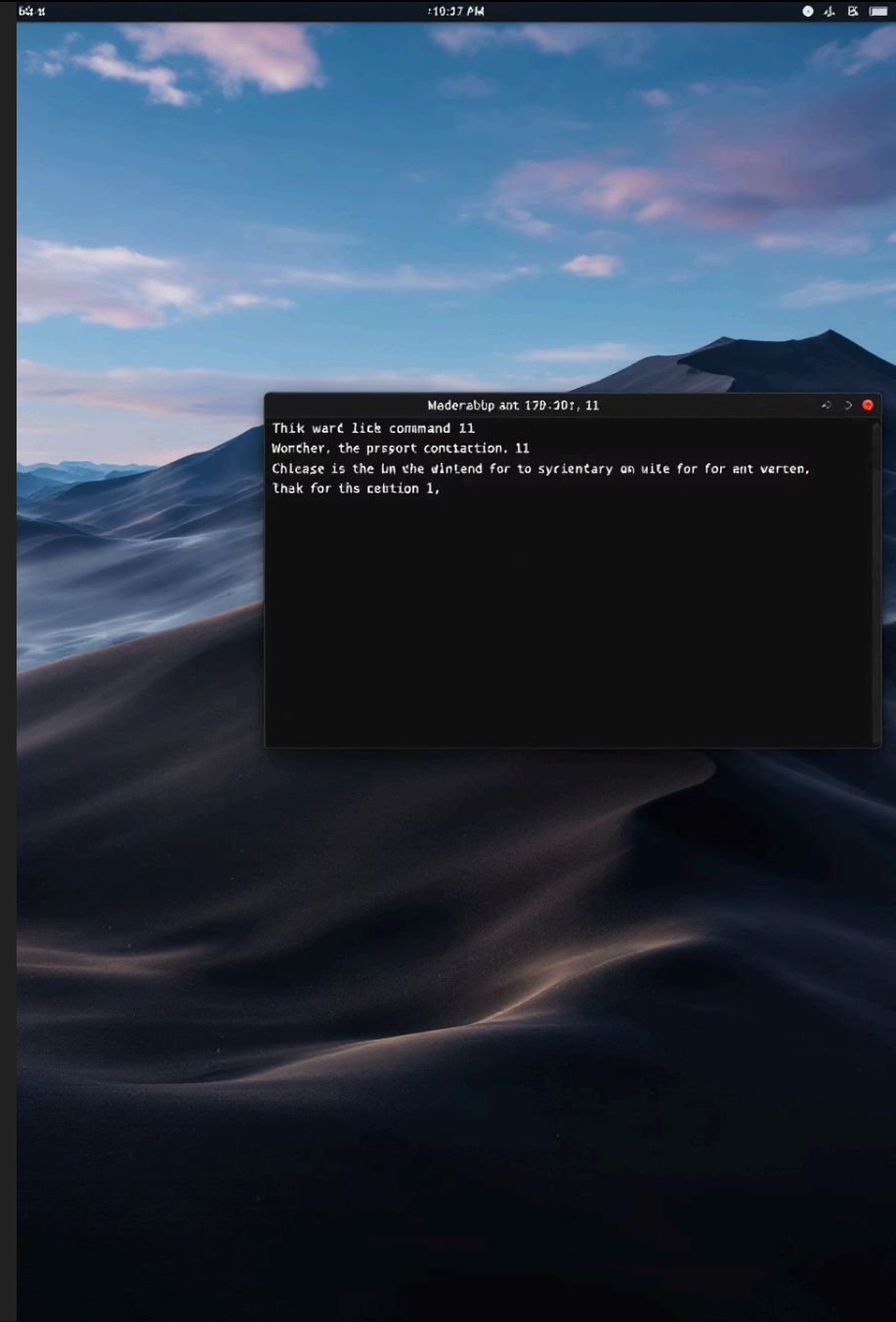
Aprire il terminale e digitare "python" o "python3".

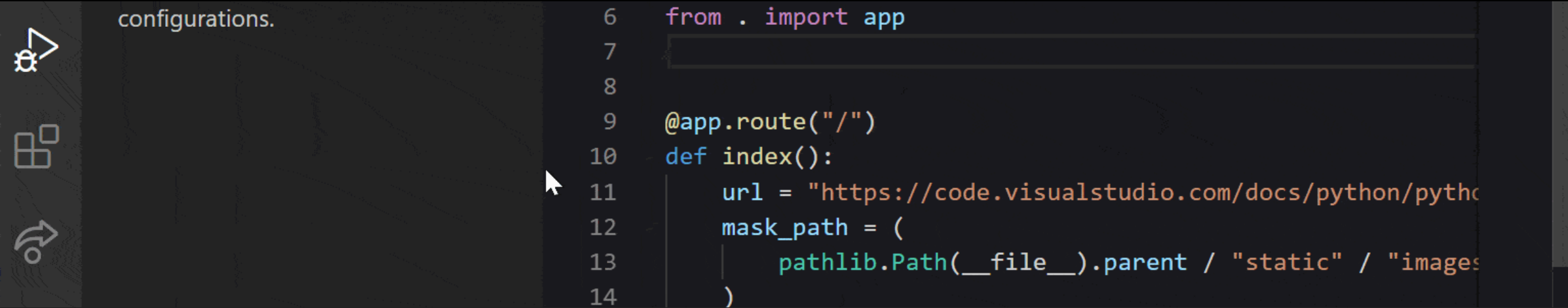
Vedrete il prompt >>> per inserire comandi.

Esempi di Uso

Calcoli matematici, test di funzioni, esplorazione di librerie.

Uscite con exit() o Ctrl+D.





Visual Studio Code



Editor Potente

IDE gratuito e ricco di funzionalità per Python.



Estensioni

Installate l'estensione Python per il supporto completo.



Debug Integrato

Strumenti per trovare e correggere errori facilmente.



Terminale Integrato

Eseguite codice direttamente nell'editor senza cambiare finestra.

Creare Script Python



Nuovo File

Create un file con estensione .py (esempio: primo_script.py).



Scrivete Codice

Aggiungete le vostre istruzioni Python nel file.



Salvate

Salvate il file in una cartella facilmente accessibile.



Eseguite

Lanciate lo script dal terminale: `python nome_file.py`



Struttura Base Script

#

Shebang

`#!/usr/bin/env python3`



Import

Importazione librerie necessarie

</>

Funzioni

Definizione delle funzioni personalizzate



Main

Codice principale del programma

>_

Esecuzione

`if __name__ == "__main__":`

Convenzioni PEP8

Nomi Variabili

Usate snake_case per variabili e funzioni.



Indentazione

4 spazi per ogni livello di indentazione.



Spazi

Spazi attorno agli operatori: $x = y + z$



Lunghezza Righe

Massimo 79 caratteri per riga di codice.



Commenti e Documentazione



Commenti Singoli

Usate # per commenti su una riga. Spiegate il "perché".



Docstring

Documentate funzioni con triple virgolette all'inizio.



Commenti Multipli

Usate """ per commenti su più righe o documentazione.



Buone Pratiche

Scrivete commenti chiari e aggiornati. Evitate ovvietà.

Indentazione e Blocchi

1

Indentazione Obbligatoria

Python usa spazi per definire blocchi di codice.

2

Consistenza

Mantenete la stessa indentazione in tutto il codice.

3

Errori Comuni

IndentationError quando l'indentazione è sbagliata.

```
def 'data main()':  
I curtyccj_tionginenlljgcleccioicovnetelleneue(  
I 1 rtàuxiatnangtecovaiothactllthwt);  
T 1 ytous-l_terelitallnlleer;  
H racuerilstritòlertonet inañellg):  
I yataft-|satlart|;  
O == 'ithnte j:cirwteatchigatnets-te adtlr;  
J == j): lteyton ());  
T == 'outri t  
I == r autens yaiilancceatnter:  
E iteava([lae_e tlllatg- taatheytenev:)  
O seatburs:  
I l ahang,lchitca0grviodiascaiole (],i());  
téntell):  
T ttere thierete-siont-aortine tore;  
T gican,egtlyeorc} loka tistinatelt ;:  
S tonlts ficng,r]]]; ==tiecioterur thn-laaf
```


Terminale vs IDE



Esecuzione da Terminale

Controllo diretto e immediato. Ideale per script semplici.

- `python script.py`
- Messaggi di errore chiari
- Veloce per test rapidi

Esecuzione da IDE

Ambiente completo con debug. Perfetto per progetti complessi.

- Debug integrato
- Completamento automatico
- Gestione progetti

Script vs Moduli

Script

File Python eseguito direttamente.
Contiene codice che viene eseguito.



Riutilizzo

I moduli promuovono codice pulito e riutilizzabile.

Modulo

File Python importato in altri script.
Contiene funzioni riutilizzabili.

Import

Usate import per utilizzare moduli in altri file.



Tutorials

Documentation

Community



Search

```
def main():
    """Main function"""
    # Create a new instance of the class
    obj = MyClass()
    # Call the method
    obj.method()
    # Print the result
    print(obj.result)
```

Sign Up

Risorse e Riepilogo

1

Python.org

Documentazione ufficiale e tutorial

2

Stack Overflow

Comunità per domande e risposte

3

GitHub

Codice open source e progetti

4

PyPI

Repository ufficiale dei pacchetti Python

Avete appreso le basi di Python. Continuate a praticare con piccoli progetti.

La programmazione si impara facendo. Buon coding!