

Queste slides fanno parte del corso *Crea il Tuo Blog Personale con Python e Flask* di programmareinpython.it, il Sito Web Italiano dedicato al linguaggio di programmazione Python.

Per conoscere tutte le news e gli aggiornamenti in tempo reale sui video corsi, i tutorial gratuiti, gli esercizi e gli articoli dal blog, puoi seguirmi sui Canali Ufficiali:

- ▶ YouTube: <https://www.youtube.com/c/pymike>
- ▶ Telegram: <https://t.me/programmareinpython>
- ▶ Facebook: <https://www.facebook.com/programmareinpython.it>
- ▶ Instagram: <https://www.instagram.com/programmareinpython.it/>
- ▶ Twitter: <https://www.twitter.com/pymike00>
- ▶ GitHub: <https://github.com/pymike00>
- ▶ Newsletter: <http://bit.ly/2ywWCCh>



Crea il Tuo Blog Personale con Python e Flask

Panoramica del Corso



Panoramica del Corso

Anzitutto: **grazie mille per esserti iscritto/a al corso!**

In questa lezione faremo una breve panoramica che ti permetterà di ottenere il massimo da questa esperienza di apprendimento.

Panoramica del Corso

In questo corso imparerai tutto ciò che devi sapere per iniziare da subito a creare siti, web app o microservizi con **Flask**!

Per fare ciò useremo questo **Web Framework** per costruire assieme un Blog moderno con funzionalità CRUD complete, sistema di autenticazione e permessi, supporto alla sintassi markdown, upload di immagini...

Questo ci permetterà di prendere familiarità da subito con tutti i concetti chiave e i principali package dell'ecosistema Flask, tramite la creazione di un caso d'uso reale.

Panoramica del Corso

Vedremo anche tutti gli step necessari a collegare il Blog ad un dominio web e come generare un certificato SSL per connetterci via HTTPS.

Inoltre vedremo anche come creare un tema grafico responsivo su base Bootstrap, in modo da disporre di un layout personalizzato professionale.

Terminato questo corso dunque, oltre ad aver appreso cosa sia e come usare il Web Framework Flask nel migliore dei modi, disporrai anche di un Blog che volendo, potrai usare da subito come tuo Blog Personale.

Panoramica del Corso

Questo corso è suddiviso in tre sezioni:

- Sezione Introduttiva: dove setteremo l'ambiente di sviluppo, faremo un'introduzione al Framework e creeremo il nostro primo progetto Flask
- Sezione Principale: dove svilupperemo il nostro progetto spiegando passo passo tutti gli aspetti principali di Flask e come usarlo al meglio
- Terza Sezione: dedicata alla messa online del Blog, dall'allestimento del server al collegamento del dominio alla creazione del certificato

Panoramica del Corso

È molto importante seguire l'ordine delle lezioni, ma sei tu a scegliere quando e come seguirle.

Volendo puoi anche modificare la velocità di riproduzione dalle impostazioni del Video Player, e ciascuna lezione può essere rivista quante volte vuoi.

Panoramica del Corso

Qualora il video dovesse risultare sfocato, assicurati di aver settato la risoluzione delle lezioni al massimo disponibile o su *automatico*, che scalerà automaticamente il video in base alla tua attuale velocità di connessione fino a un massimo di 1080p.

Per qualsiasi problema di natura tecnica puoi contattare il Team di Assistenza di Udemy all'indirizzo **support@udemy.com** o aprendo un ticket di supporto sul sito.

Panoramica del Corso

Come ottenere il materiale didattico del corso?

Puoi scaricare tutte le slides e il codice utilizzato per ciascuna lezione dalle risorse scaricabili abbinate alla lezione stessa, o puoi scaricare un archivio con tutto il materiale scaricabile del corso dalla lezione successiva.

Panoramica del Corso

Come ottenere aiuto durante il corso?

Panoramica del Corso

Nella gran parte dei casi la soluzione più rapida è fare una cara vecchia ricerca su Google!

I messaggi di errore per framework famosi come Flask sono spesso molto comuni, e questo ti permetterà di ottenere una risposta in tempi brevissimi da siti dedicati come *stackoverflow.com*, e inoltre imparare a debuggare le proprie app in autonomia è una skill fondamentale!

Panoramica del Corso

Tieni inoltre a mente che come sempre, nel mondo della programmazione anche una sola virgola fuori posto può causare il crash dell'applicazione!

Effettuare un raffronto / sostituzione col codice scaricabile si dimostra in questi casi una metodologia di risoluzione efficace e rapida.

Panoramica del Corso

Nella sezione Domande e Risposte, assicurati innanzitutto che la domanda non sia già stata posta e risposta, cercando tramite l'apposito input testuale.

Quando poni una domanda assicurati di:

- Scrivere un titolo esplicativo del problema
- Far riferimento o aggiungere il titolo della lezione relativa
- Descrivi l'errore con attenzione e posta sempre i messaggi di errore
- Aggiungi uno screenshot o il codice che pensi stia scatenando l'evento!

Panoramica del Corso

Grazie!



Crea il Tuo Blog Personale con Python e Flask

Preparazione Ambiente di Sviluppo



Preparazione Ambiente di Sviluppo

In questa lezione allestiremo il nostro ambiente di sviluppo per lavorare comodamente con Python e Flask.

In questo corso viene utilizzato **Visual Studio Code**, un potente editor per codice open source sviluppato da Microsoft, che come vedremo ci mette a disposizione tante funzionalità programmer-friendly già pronte all'uso.

Se già utilizzate Visual Studio Code o se preferite usare un altro editor come Atom o Sublime Text potete passare direttamente alla lezione successiva.

Preparazione Ambiente di Sviluppo

Per quanto riguarda la versione di Python da impiegare è invece molto importante usare una versione non precedente a Python 3.6, per evitare problemi di compatibilità col codice che scriveremo e le varie librerie che verranno utilizzate.

Iniziamo Subito!

Preparazione Ambiente di Sviluppo

<https://code.visualstudio.com/>



Crea il Tuo Blog Personale con Python e Flask

Introduzione a Flask e Primo Progetto



Introduzione a Flask e Primo Progetto

In questa lezione faremo un'introduzione a Flask e creeremo il nostro primo progetto con questo Framework.

Questo ci permetterà di comprenderne il contesto di sviluppo e acquisire familiarità con gli aspetti propri e caratteristici di Flask, in modo da creare una base da cui partire per introdurre concetti sempre più avanzati mentre creiamo il nostro Blog Personale!

Introduzione a Flask e Primo Progetto

Prima di iniziare con la scrittura del codice è bene rispondere ad alcune domande:

Introduzione a Flask e Primo Progetto

Prima di iniziare con la scrittura del codice è bene rispondere ad alcune domande:

- ★ Che cos'è esattamente Flask?

Introduzione a Flask e Primo Progetto

Prima di iniziare con la scrittura del codice è bene rispondere ad alcune domande:

- ★ Che cos'è esattamente Flask?
- ★ Cosa distingue Flask dagli altri Web Framework?

Introduzione a Flask e Primo Progetto

Prima di iniziare con la scrittura del codice è bene rispondere ad alcune domande:

- ★ Che cos'è esattamente Flask?
- ★ Cosa distingue Flask dagli altri Web Framework?
- ★ Chi usa Flask, e per cosa?

Introduzione a Flask e Primo Progetto

Flask è un Web Framework Open Source scritto con Python caratterizzato da flessibilità, leggerezza e semplicità d'uso.

Fa parte della categoria dei **micro-framework** per via del suo approccio allo sviluppo minimalista non opinionato, che lascia agli sviluppatori la possibilità di scegliere *quando* e soprattutto *come* implementare ogni aspetto delle proprie web applications a partire dalla struttura del progetto, scegliendo quali funzionalità è davvero necessario implementare, quanti e quali file/moduli creare, le convenzioni di sviluppo a cui fare riferimento e così via.

Introduzione a Flask e Primo Progetto

Nel caso di Flask, *micro-framework* significa quindi possedere un *core semplice e performante* che mette a disposizione funzionalità fondamentali quali server di sviluppo e debugger, routing, supporto unit testing integrato, protezione contro cross-site scripting (XSS) e l'impiego di Jinja 2 come template engine.

A differenza dei web framework full stack, Flask non include ad esempio una database API, un sistema di autenticazione, upload o di validazione dei form.

Questo genere di funzionalità vengono incluse in una web app o sito scritto con Flask tramite l'utilizzo di *estensioni dedicate*, che una volta integrate nel progetto potranno essere usate facilmente come se facessero parte del framework stesso!

Introduzione a Flask e Primo Progetto

Nel sito ufficiale di Flask è presente un vero e proprio **registro delle estensioni** moderato dagli sviluppatori del framework e aggiornato regolarmente, in modo da garantire un alto livello di qualità per tutte le estensioni più importanti.

In fase di creazione del nostro Blog Personale impiegheremo molte di queste estensioni tra cui ad esempio **Flask-Login** per l'aggiunta di un sistema di autenticazione, **Flask-SQLAlchemy** e **Flask-Migrate** per lavorare comodamente col database, **Flask-WTF** per la gestione dei form ed altre ancora.

Introduzione a Flask e Primo Progetto

Flask può quindi essere utilizzato per qualsiasi tipologia di progetto, sta in cima alla lista dei web framework più utilizzati assieme a Django, ed entrambi possono vantare ben oltre 40.000 stelle su GitHub a prova del loro grande valore nella Community Python.

Tra i suoi utilizzatori celebri vanta **LinkedIn** e **Pinterest**, ed è impiegato per la creazione di siti, web app complete e microservizi in progetti e aziende di ogni livello.

Grazie al suo approccio allo sviluppo flessibile, che incoraggia la personalizzazione del codice partendo da una base minimalista, facilita la creazione di componenti e servizi altamente personalizzati.

Introduzione a Flask e Primo Progetto

Può essere installato come un qualsiasi altro package Python tramite pip, col comando “pip install flask”

Iniziamo ora con la creazione del nostro Primo Progetto Flask!

Introduzione a Flask e Primo Progetto

<http://flask.pocoo.org/docs/dev/foreword/>

Estensioni Flask (nuovo link aggiornato): <https://pypi.org/search/?c=Framework+%3A%3A+Flask>


<https://www.youtube.com/watch?v=OXN3wuHUBP0>

<https://www.quora.com/What-challenges-has-Pinterest-encountered-with-Flask/answer/Steve-Cohen>



Crea il Tuo Blog Personale con Python e Flask

La Struttura del Nostro Blog



La Struttura del Nostro Blog

Abbiamo visto come una delle caratteristiche più distintive di Flask sia il suo approccio allo sviluppo non opinionato, che lascia a noi sviluppatori la libertà di strutturare i nostri progetti scegliendo quali file e moduli creare in base alle esigenze del caso o alle nostre preferenze.

Per quanto sia di fatto possibile scrivere tutto il codice per un'app Flask anche complessa all'interno di un unico file Python, difficilmente questa scelta verrà mai operata da developers esperti per la creazione di casi d'uso reali.

La Struttura del Nostro Blog

Una buona struttura base per ogni progetto software concreto terrà infatti a mente il principio di design noto come **Separation of Concerns (SoC)**, traducibile come *Separazione delle Competenze* o *Separazione dei Ruoli*.

Risulta molto utile suddividere il codice in maniera logica su più file in base al ruolo svolto nell'applicazione, creando una base robusta e mantenibile su cui poter sviluppare man mano tutti gli aspetti dell'applicazione.

La Struttura del Nostro Blog

Come il titolo suggerisce, in questa lezione creeremo la struttura base del nostro blog Flask che andremo poi a migliorare ed espandere durante tutto il corso, e introdurremo anche un package molto utile chiamato *python-dotenv*.

Siete Pronti? Iniziamo!



Crea il Tuo Blog Personale con Python e Flask

Templates e File Statici



Templates e File Statici

Fin'ora abbiamo visto come sia possibile usare delle funzioni, che abbiamo chiamato *views*, per ricevere richieste HTTP su degli indirizzi definiti tramite il decoratore `@app.route()`, ed elaborare delle risposte da fornire agli utenti.

Generare tutto il codice HTML per il nostro Blog tramite le *views* può però portare a confusione, e in generale come abbiamo accennato è sempre bene mantenere una buona *Separazione dei Ruoli* a livello software, in questo caso tra la parte puramente logica e quella propriamente grafica.

Templates e File Statici

In questa lezione vedremo cosa sono e come utilizzare i **Templates** in Flask tramite l'impiego del Template Engine **Jinja 2**.

I Templates ci permettono di racchiudere la veste grafica del nostro Blog in file HTML dedicati da renderizzare utilizzando le views, e il bello dei Templates è che permettono anche l'interpolazione testuale e l'uso di logica base dedicata!

Vedremo inoltre come utilizzare file statici quali file .css ed immagini, in modo da poter creare un tema grafico professionale per il nostro Blog Personale.


Passiamo su VS Code!

Templates e File Statici

<http://jinja.pocoo.org/>


<http://flask.pocoo.org/docs/1.0/quickstart/#rendering-templates>

<http://flask.pocoo.org/docs/1.0/quickstart/#static-files>



Crea il Tuo Blog Personale con Python e Flask

Creazione Layout Grafico



Creazione Layout Grafico

In questa lezione creeremo il Layout Grafico del nostro Blog, utilizzando i Templates di Flask, CSS e Bootstrap!

Volendo potete scaricare tutto il codice HTML e CSS utilizzato e integrarlo da voi secondo quanto abbiamo visto nella lezione precedente.

Iniziamo Subito!



Crea il Tuo Blog Personale con Python e Flask

Database, Modelli e Migrazioni



Database, Modelli e Migrazioni

In questa lezione vedremo come integrare un database nel nostro progetto in modo da poter finalmente aggiungere e salvare Post nel nostro Blog!

Ricordando che come abbiamo spiegato nella lezione introduttiva, Flask non dispone di un'interfaccia di supporto nativa per database, e considerando che si tratta di una delle parti più sensibili di ogni Web App, per fare ciò useremo delle estensioni dedicate, e l'ecosistema Flask ne mette a disposizione diverse!

Database, Modelli e Migrazioni

Tra le varie scelte possibili, noi vedremo come usare una delle combinazioni di estensioni tra le più sicure e di facile impiego per questo scopo:

Flask-SQLAlchemy e *Flask-Migrate*.

Di che si tratta esattamente?

Database, Modelli e Migrazioni

Flask-SQLAlchemy è un'estensione che semplifica l'utilizzo su Flask di SQLAlchemy, un Object Relational Mapper (ORM) molto famoso per Python.

SQLAlchemy permette la gestione del database tramite entità di alto livello come classi, oggetti e metodi invece di tabelle e SQL: in pratica offre *un'astrazione* che ci permette di interagire col database utilizzando Python.

SQLAlchemy supporta tutti i principali database relazionali quali PostgreSQL, SQLite e MySQL.

Database, Modelli e Migrazioni

Flask-Migrate permette invece di gestire facilmente i cambiamenti allo schema del database, ad esempio per creare una nuova tabella o aggiungere una colonna ad una tabella già presente.

Database, Modelli e Migrazioni

Spiegheremo tutti i passaggi principali del loro utilizzo nel dettaglio e torneremo su questi concetti più volte durante il corso, così da poter capire a fondo il loro funzionamento.

Iniziamo Subito!

Database, Modelli e Migrazioni

<https://flask-sqlalchemy.palletsprojects.com/en/2.x/>

<https://www.sqlalchemy.org/>

<https://flask-migrate.readthedocs.io/en/latest/>

[https://it.wikipedia.org/wiki/Structured Query Language](https://it.wikipedia.org/wiki/Structured_Query_Language)

<https://stackoverflow.com/questions/1279613/what-is-an-orm-and-where-can-i-learn-more-about-it#1279678>

https://en.wikipedia.org/wiki/Schema_migration



Crea il Tuo Blog Personale con Python e Flask

Database Queries e User Password Hashing



Database Queries e User Password Hashing

In questa lezione impareremo ad effettuare query sul database del nostro Blog utilizzando Flask-SQLAlchemy e SQLAlchemy.

Aggiungeremo anche due metodi al nostro modello User per implementare un sistema di *password hashing*, in modo da salvare delle versioni crittografate delle password nel database, per ragioni di sicurezza.

Iniziamo!

Database Queries e User Password Hashing

<https://docs.sqlalchemy.org/en/13/orm/query.html>

<https://flask-sqlalchemy.palletsprojects.com/en/2.x/queries/>

<https://www.fastweb.it/internet/password-hashing-cos-e-e-cosa-serve/>

https://it.wikipedia.org/wiki/Funzione_crittografica_di_hash

[https://it.wikipedia.org/wiki/Sale_\(crittografia\)](https://it.wikipedia.org/wiki/Sale_(crittografia))



Crea il Tuo Blog Personale con Python e Flask

HomePage e Renderizzazione della Lista di Posts



HomePage e Renderizzazione della Lista di Posts

Ora che abbiamo imparato come ottenere il queryset dei Posts presenti nel database, è arrivato il momento di mostrare questo elenco nell'homepage del nostro Blog.

Passiamo su VS Code!



Crea il Tuo Blog Personale con Python e Flask

Come Renderizzare Post Singoli



Come Renderizzare Post Singoli

In questa lezione creeremo una nuova route per mostrare i dettagli relativi a un singolo Post del Blog, e nel fare ciò impareremo come passare parametri via URL alle nostre views, in modo da poter usare la chiave primaria di ogni Post per reperirne i dettagli dal database.

Iniziamo!



Crea il Tuo Blog Personale con Python e Flask

Autenticazione Utenti: Login e Logout



Autenticazione Utenti: Login e Logout

Il nostro Blog Personale sta migliorando sempre più, e a breve aggiungeremo la possibilità di creare nuovi Post tramite un'interfaccia web dedicata.

Prima di fare ciò dobbiamo però implementare un Sistema di Autenticazione, così che solo gli amministratori del Blog possano avere la capacità di aggiungere nuovi articoli o modificare e cancellare quelli già presenti nel database.

Autenticazione Utenti: Login e Logout

Nel fare ciò vedremo ora come usare due nuove estensioni per Flask, **Flask-Login** e **Flask-WTF**.

Flask-Login gestirà le fasi di login e logout degli utenti, mentre Flask-WTF ci permetterà di creare i form di login necessari in maniera sicura e rapida.

Iniziamo Subito!

Autenticazione Utenti: Login e Logout

<https://flask-login.readthedocs.io/en/latest/>

<https://flask-wtf.readthedocs.io/en/stable/>



Crea il Tuo Blog Personale con Python e Flask

Autenticazione Utenti: Settaggio Login Manager





Crea il Tuo Blog Personale con Python e Flask

Interfaccia per la Creazione di Nuovi Post



Interfaccia per la Creazione di Nuovi Post

Ora che il Sistema di Autenticazione è stato implementato, possiamo finalmente occuparci dell'aggiunta di nuovi Post!

Useremo Flask-WTF per la creazione di un nuovo apposito form, e introdurremo una nuova estensione chiamata **Flask-Misaka** per aggiungere il supporto a *Markdown*, un linguaggio di markup che ci permetterà di formattare gli articoli del nostro blog per aggiungere ad esempio intestazioni, corsivo, grassetto ecc.

Tra i link di referenza per questa lezione è presente un link a un'introduzione alla sintassi Markdown. - **Iniziamo Subito!**

Interfaccia per la Creazione di Nuovi Post

<https://flask-misaka.readthedocs.io/en/latest/>

<https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet>



Crea il Tuo Blog Personale con Python e Flask

Implementazione funzionalità di Aggiornamento



Implementazione funzionalità di Aggiornamento


In questa lezione vedremo come aggiungere la possibilità di **aggiornare** gli articoli già pubblicati nel nostro Blog.

Passiamo su VS Code!



Crea il Tuo Blog Personale con Python e Flask


Implementazione funzionalità di Cancellazione



Implementazione funzionalità di Cancellazione


Completando l'aggiunta delle funzionalità CRUD al nostro Blog Personale, in questa lezione vedremo come aggiungere la possibilità di **cancellare** gli articoli già pubblicati.

Iniziamo Subito!



Crea il Tuo Blog Personale con Python e Flask

Aggiunta campo Slug per l'ottimizzazione SEO



Aggiunta campo Slug per l'ottimizzazione SEO

Fin'ora per reperire i dettagli di una singola istanza di Post abbiamo utilizzato l'ID dell'istanza, ed essendo l'ID passato come parametro via URL, il risultato è che al momento i link del nostro Blog sono poco espliciti e informativi, il che rappresenta una penalizzazione dal punto di vista SEO (Search Engine Optimization).

È molto importante per quanto possibile ottimizzare il proprio sito in modo che questo possa essere scoperto e visitato più facilmente.

Aggiunta campo Slug per l'ottimizzazione SEO

In questa lezione modificheremo il modello Post aggiungendo un nuovo campo **slug** basato sul titolo del Post stesso, così da poter usare questo campo per reperire i dettagli dell'istanza dal database e come conseguenza ottimizzare ogni pagina di dettaglio del nostro Blog dal punto di vista di questo aspetto SEO.

Iniziamo Subito!



Crea il Tuo Blog Personale con Python e Flask

Come gestire l'Upload di Immagini in Flask



Come gestire l'Upload di Immagini in Flask

In questa lezione apporteremo un'ulteriore modifica al modello Post aggiungendo un campo immagine!

Per comprenderne a pieno il funzionamento, creeremo da noi il sistema necessario alla gestione dell'upload, usando una libreria Python per la gestione delle immagini chiamata Pillow.

Iniziamo subito!



Crea il Tuo Blog Personale con Python e Flask

Rifinitura Interfaccia Grafica





Crea il Tuo Blog Personale con Python e Flask


Implementazione della Paginazione Posts



Implementazione della Paginazione Posts


In questa lezione vedremo come implementare un sistema di paginazione dei Post, così da poter migliorare l'esperienza d'uso per i nostri utenti.

Iniziamo!



Crea il Tuo Blog Personale con Python e Flask

Creazione Pagina di Contatto



Creazione Pagina di Contatto

In questa lezione aggiungeremo una nuova pagina simile ad about, dove aggiungere i vari link ai nostri profili social e una mail di contatto.

Passiamo su VS Code e iniziamo!



Crea il Tuo Blog Personale con Python e Flask

Gestione degli Errori e Pagine Dedicate



Gestione degli Errori e Pagine Dedicate

In questa lezione vedremo come fare in modo che il nostro Blog possa gestire eventuali errori innescati da parte degli utenti, come ad esempio l'errore 404 che si manifesta quando viene richiesta una pagina che non esiste, a causa ad esempio di un link copiato male.

Iniziamo Subito!



Crea il Tuo Blog Personale con Python e Flask

Deploying Parte I - Digital Ocean VPS



Deploying Parte I - Digital Ocean VPS

Siamo finalmente pronti per la messa online del nostro Blog, e faremo ciò usando i servizi offerti da un provider professionale: Digital Ocean.

In questa lezione vedremo come creare e configurare un Droplet su base Ubuntu, allestendo tutto ciò di cui avremo bisogno per poter iniziare a lavorare con Flask sul server dalla lezione successiva.

Tra i link di referenza a questa lezione ne troverete uno che vi permetterà di ottenere del credito gratuito se già non disponete di un account Digital Ocean.

Deploying Parte I - Digital Ocean VPS

Tutta la configurazione verrà effettuata tramite il protocollo SSH, che ci permetterà di connetterci direttamente alla shell del server in maniera sicura.

Su Linux e Mac OS è già presente un client SSH dedicato accessibile tramite terminale, e anche su Windows 10 dovrete aver accesso ad un client SSH tramite PowerShell.

Qualora non disponiate di un client SSH su Windows, potete usare *Putty*.


Per trasferire tutti i file del nostro progetto sul server useremo invece *FileZilla*.

Deploying Parte I - Digital Ocean VPS

FileZilla: <https://filezilla-project.org>


Digital Ocean: <https://m.do.co/c/5a0beb860638>

Putty: www.putty.org



Crea il Tuo Blog Personale con Python e Flask

Deploying Parte II - Supervisor ed Nginx



Deploying Parte II - Supervisor ed Nginx

Ora che il server e l'ambiente virtuale sono pronti, avendo verificato il corretto funzionamento di Gunicorn, in questa lezione vedremo come configurare Supervisor e Nginx.

Supervisor è un tool che garantisce l'operatività del nostro sito monitorando il corretto funzionamento di Gunicorn, riavviandolo quando necessario.

Nginx ci permette di esporre il nostro sito al pubblico facendo da proxy per la configurazione locale di Gunicorn e garantendo una buona fruizione dei contenuti statici da parte degli utenti.


Tra le risorse scaricabili potete trovare i file di configurazione necessari.



Crea il Tuo Blog Personale con Python e Flask


Deploying Parte III - Dominio e Certificato HTTPS





Crea il Tuo Blog Personale con Python e Flask

Ringraziamenti e Sguardo al Futuro



 PROGRAMMAREINPYTHON.IT

YouTube: PyMike

Facebook: [programmareinpython.it](https://www.facebook.com/programmareinpython.it)

Telegram: [programmareinpython](https://www.telegram.com/programmareinpython)

GitHub, Twitter: [pymike00](https://www.github.com/pymike00)