

Master

Sviluppo in Python



tech global
university



Master Sviluppo in Python

- » Modalità: online
- » Durata: 12 mesi
- » Titolo: TECH Global University
- » Accreditamento: 60 ECTS
- » Orario: a tua scelta
- » Esami: online

Accesso al sito web: www.techtitute.com/it/informatica/master/master-sviluppo-python



Indice

01

Presentazione

pag. 4

02

Obiettivi

pag. 8

03

Competenze

pag. 12

04

Direzione del corso

pag. 16

05

Struttura e contenuti

pag. 20

06

Metodologia

pag. 30

07

Titolo

pag. 38

01

Presentazione

Python si distingue come linguaggio di programmazione versatile ed efficiente. Innanzitutto, la sua sintassi chiara e leggibile rende facile la scrittura e la comprensione del codice, accelerando il processo di sviluppo e riducendo la probabilità di errori. Inoltre, l'ampia varietà di librerie e *Frameworks* disponibili consente agli sviluppatori di accedere a strumenti potenti e preesistenti, permettendo uno sviluppo più rapido ed efficiente. La sua natura open source e l'attiva comunità di sviluppatori contribuiscono inoltre al costante miglioramento e aggiornamento del linguaggio, garantendo un ambiente di sviluppo dinamico e moderno. Per questo motivo, TECH ha sviluppato un programma completo, volto a formare i professionisti con gli ultimi progressi nello sviluppo di Python, con una metodologia 100% online.



66

Python è riconosciuto per la sua adattabilità, essendo utilizzato in applicazioni che vanno dallo sviluppo web all'Intelligenza Artificiale. Cosa aspetti a iscriverti?"

Python è un linguaggio di Programmazione di alto livello, molto utilizzato dagli informatici, in quanto dispone di un'ampia gamma di librerie e framework che semplificano le attività più comuni, consentendo agli sviluppatori di concentrarsi sulla logica delle loro applicazioni invece di perdere tempo con i dettagli tecnici. Inoltre, la sua versatilità è un altro dei suoi vantaggi significativi, in quanto può essere utilizzato in una varietà di ambienti, dallo sviluppo web all'analisi dei dati e all'apprendimento automatico.

Nasce così questo Master in Sviluppo in Python, un programma che affronta la gestione avanzata dei dati e dei tipi in Python, esplorando gli identificatori, le parole chiave, i tipi integrali e booleani, nonché la formattazione e la codifica avanzata delle *strings*. Inoltre, verranno esaminate collezioni come tuple, liste e dizionari, oltre a tecniche di iterazione e funzioni lambda, fornendo una solida base di fondamenti del linguaggio.

Verrà inoltre approfondito lo sviluppo di applicazioni Python, con particolare attenzione alle best practice e alle metodologie moderne. Dall'architettura dell'applicazione al deployment e alla manutenzione, verranno trattati aspetti come la progettazione e la modellazione avanzata, la gestione delle dipendenze, i design pattern, il *testing* e il *debugging*, l'ottimizzazione delle prestazioni, le strategie di deployment e di distribuzione.

Allo stesso modo, l'informatico si addentrerà nello sviluppo web e mobile con Python, coprendo frameworks, come Django e Flask, nonché lo sviluppo di API e servizi web. Si concentrerà anche sulla progettazione di interfacce e sull'esperienza utente (UI/UX), dall'uso di strumenti di progettazione al miglioramento dell'accessibilità e dell'usabilità. Infine, si affronterà la gestione e l'analisi dei dati, utilizzando Python e strumenti come NumPy, Pandas e Matplotlib.

Questo titolo di studio si presenta quindi come un'opportunità unica, attraverso una proposta accademica completamente online e adattabile. Grazie a questo approccio, i professionisti potranno godere di una maggiore libertà nella gestione del proprio tempo di accesso, consentendo loro di armonizzare gli impegni quotidiani personali e lavorativi.

Questo **Master in Sviluppo in Python** possiede il programma più completo e aggiornato del mercato. Le caratteristiche principali del programma sono:

- ◆ Lo sviluppo di casi di studio presentati da esperti del settore
- ◆ Contenuti grafici, schematici ed eminentemente pratici che forniscono informazioni teoriche e pratiche sulle discipline essenziali per l'esercizio della professione
- ◆ Esercizi pratici che offrono un processo di autovalutazione per migliorare l'apprendimento
- ◆ Speciale enfasi sulle metodologie innovative
- ◆ Lezioni teoriche, domande all'esperto e/o al tutor, forum di discussione su questioni controverse e compiti di riflessione individuale
- ◆ Contenuti disponibili da qualsiasi dispositivo fisso o mobile dotato di connessione a internet



Un percorso formativo completo e specializzato nello sviluppo in Python, che ti preparerà ad affrontare le sfide reali nel mondo dello sviluppo software"

“

Attraverso questo Master, 100% online, approfondirai la progettazione di Frontend e Backend, la gestione di database e le strategie di pubblicazione negli app shop”

Il personale docente del programma comprende rinomati specialisti del settore e altre aree correlate, che forniscono agli studenti le competenze necessarie a intraprendere un percorso di studio eccellente.

Contenuti multimediali, sviluppati in base alle ultime tecnologie educative, forniranno al professionista un apprendimento coinvolgente e localizzato, ovvero inserito in un contesto reale.

La creazione di questo programma è incentrata sull'Apprendimento Basato su Problemi, mediante il quale il professionista deve cercare di risolvere le diverse situazioni che gli si presentano durante il corso. Lo studente potrà usufruire di un innovativo sistema di video interattivi creati da esperti di rinomata fama.

Si affronta tutto ciò che va dalle basi, come le variabili e i tipi di dati, alle tecniche avanzate di visualizzazione dei dati e strategie di ottimizzazione dello storage. Iscriviti subito!

Approfondirai la Programmazione Orientata agli Oggetti (POO) e argomenti quali classi, ereditarietà, polimorfismo, creazione di classi astratte ed eccezioni personalizzate.



02

Obiettivi

Questo Master è stato progettato con l'obiettivo di formare professionisti altamente competenti e specializzati nel vasto universo della programmazione in questo linguaggio ad alte prestazioni. Attraverso moduli completi, gli studenti si immergeranno nella sintassi e nelle funzionalità di Python, dalla gestione avanzata dei dati alla padronanza della Programmazione Orientata agli Oggetti (POO) e alla progettazione efficiente di applicazioni web e mobili. Grazie a un approccio pedagogico rigoroso e aggiornato, questo programma garantirà un'esperienza educativa unica che posizionerà gli informatici come esperti ricercati nel settore dello sviluppo Software. Preparati a superare i tuoi limiti e a intraprendere un viaggio educativo che non solo eleva le tue competenze in Python, ma ti fornisce anche gli strumenti per eccellere e fare da guida nel dinamico mondo dello sviluppo tecnologico.



66

Diventa un leader nel campo dello sviluppo Software, dotandoti di conoscenze avanzate e competenze specialistiche per affrontare le sfide tecnologiche contemporanee”

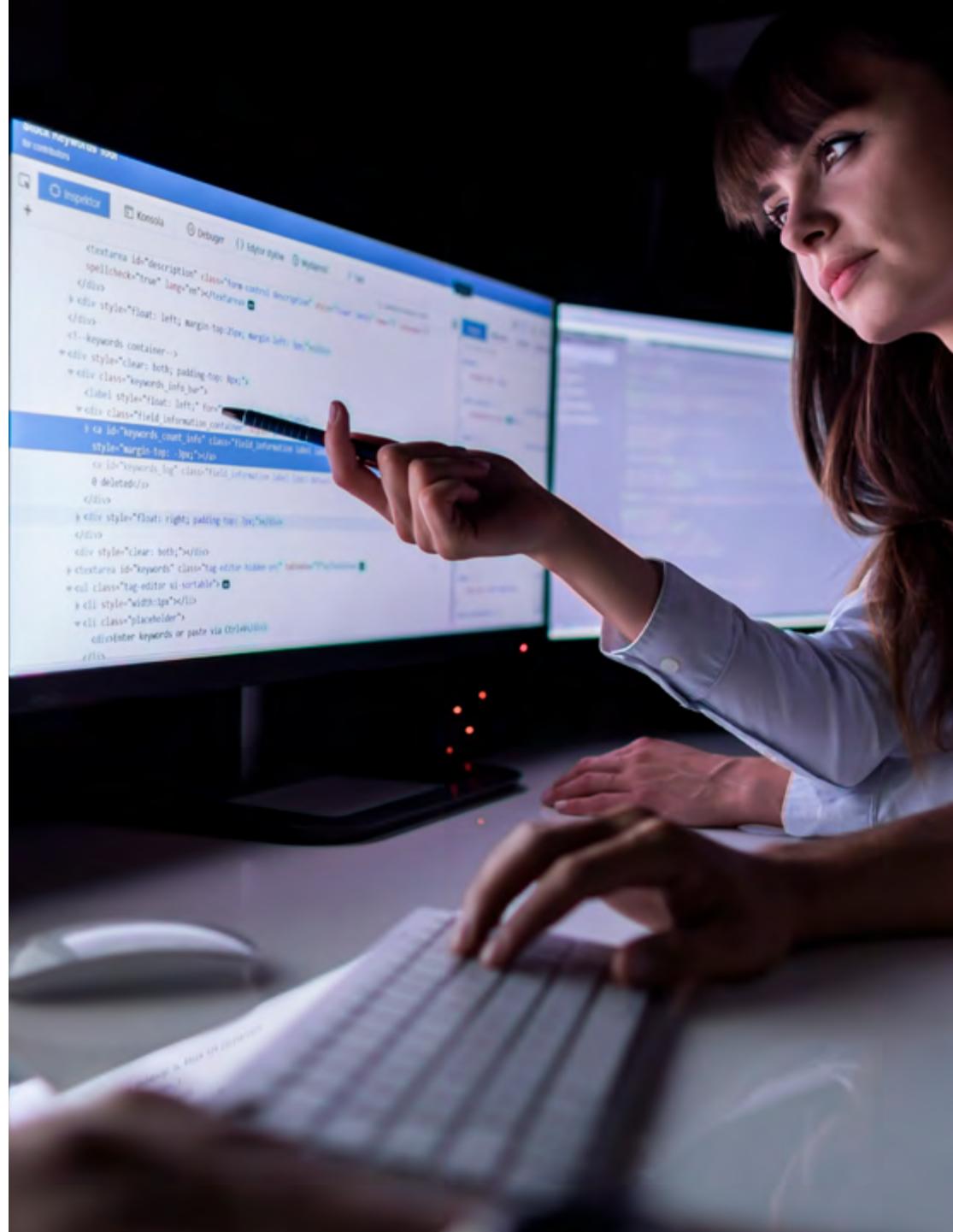


Obiettivi generali

- Fornire una comprensione completa di Python
- Abilitare la gestione avanzata dei dati e dei tipi in Python
- Applicare i principi della Programmazione Orientata agli Oggetti (POO) in Python
- Incoraggiare l'uso delle migliori pratiche e delle moderne metodologie di sviluppo del Software
- Fornire una preparazione completa sullo sviluppo web e mobile con Python
- Integrare i principi UI/UX nello sviluppo del Software
- Formare alla configurazione e all'utilizzo di strumenti e ambienti di sviluppo dati
- Approfondire l'uso di strutture dati e funzioni in Python
- Preparazione a tecniche avanzate di visualizzazione dei dati con Matplotlib
- Preparazione alle strategie di archiviazione dei dati e di ottimizzazione delle prestazioni

“

Grazie all'ampia biblioteca offerta da questo titolo di studio, ricca delle più innovative risorse multimediali, acquisirai gli strumenti necessari per distinguerti e primeggiare nel dinamico mondo dello sviluppo tecnologico”





Obiettivi specifici

Modulo 1. Programmazione in Python

- ◆ Abilitare la configurazione e l'uso efficace dell'ambiente di sviluppo Python
- ◆ Comprendere concetti di Programmazioni avanzate

Modulo 2. Dati avanzati e controllo del flusso con Python

- ◆ Padroneggiare le convenzioni e le pratiche per la gestione degli identificatori e delle parole chiave
- ◆ Applicare strutture dati complesse e le loro operazioni

Modulo 3. Programmazione Orientata agli Oggetti (POO) in Python

- ◆ Padroneggiare la creazione e l'uso di classi e oggetti in Python
- ◆ Applicare l'ereditarietà e il polimorfismo in Python

Modulo 4. Sviluppo delle applicazioni in Python

- ◆ Specializzati nella progettazione e nella modellazione di applicazioni avanzate
- ◆ Istruzione per l'ottimizzazione, l'implementazione e la manutenzione delle applicazioni

Modulo 5. Sviluppo web e mobile con Python

- ◆ Utilizzo dei *framework* web più diffusi in Python
- ◆ Preparazione allo sviluppo e alla pubblicazione di applicazioni mobili

Modulo 6. Interfaccia utente ed esperienza utente con Python

- ◆ Istruire le tecniche di design reattivo e adattivo
- ◆ Preparare test di usabilità e analisi del comportamento dell'utente

Modulo 7. Elaborazione dei dati e *Big Data* con Python

- ◆ Gestire le tecniche di controllo del flusso e le funzioni per la gestione dei dati
- ◆ Promuovere le migliori pratiche di codifica e gestione degli errori in Python

Modulo 8. Strutture dei dati e funzioni in Python

- ◆ Creare e utilizzare funzioni avanzate
- ◆ Leggere e scrivere file e la loro elaborazione

Modulo 9. Gestione dei dati in Python con NumPy e Pandas

- ◆ Creazione e manipolazione Arrays con NumPy
- ◆ Promuovere la competenza nella visualizzazione dei dati con Matplotlib

Modulo 10. Tecniche avanzate e applicazioni pratiche in NumPy e Pandas

- ◆ Sviluppo di conoscenze specialistiche nel caricamento e nell'archiviazione di dati da e verso varie fonti
- ◆ Istruzione su strategie avanzate di pulizia e trasformazione dei dati

03

Competenze

Il programma non si limita a impartire conoscenze tecniche, ma si concentra anche sulla coltivazione di competenze che trasformeranno i laureati in eccellenti professionisti nel campo dello sviluppo del Software. Questo comprenderà la capacità di sviluppare soluzioni innovative ed efficienti attraverso una padronanza esperta di Python, dalla gestione avanzata dei dati alla creazione di applicazioni web e mobili ad alte prestazioni. Inoltre, gli informatici saranno dotati di competenze specialistiche nella Programmazione Orientata agli Oggetti (POO), nella progettazione dell'interfaccia utente e dell'esperienza (UI/UX) e nell'analisi avanzata dei dati con librerie come NumPy e Pandas.



66

Sarà incoraggiato lo sviluppo di soft skills cruciali come il pensiero critico, il problem solving, la collaborazione di gruppo e l'adattabilità”

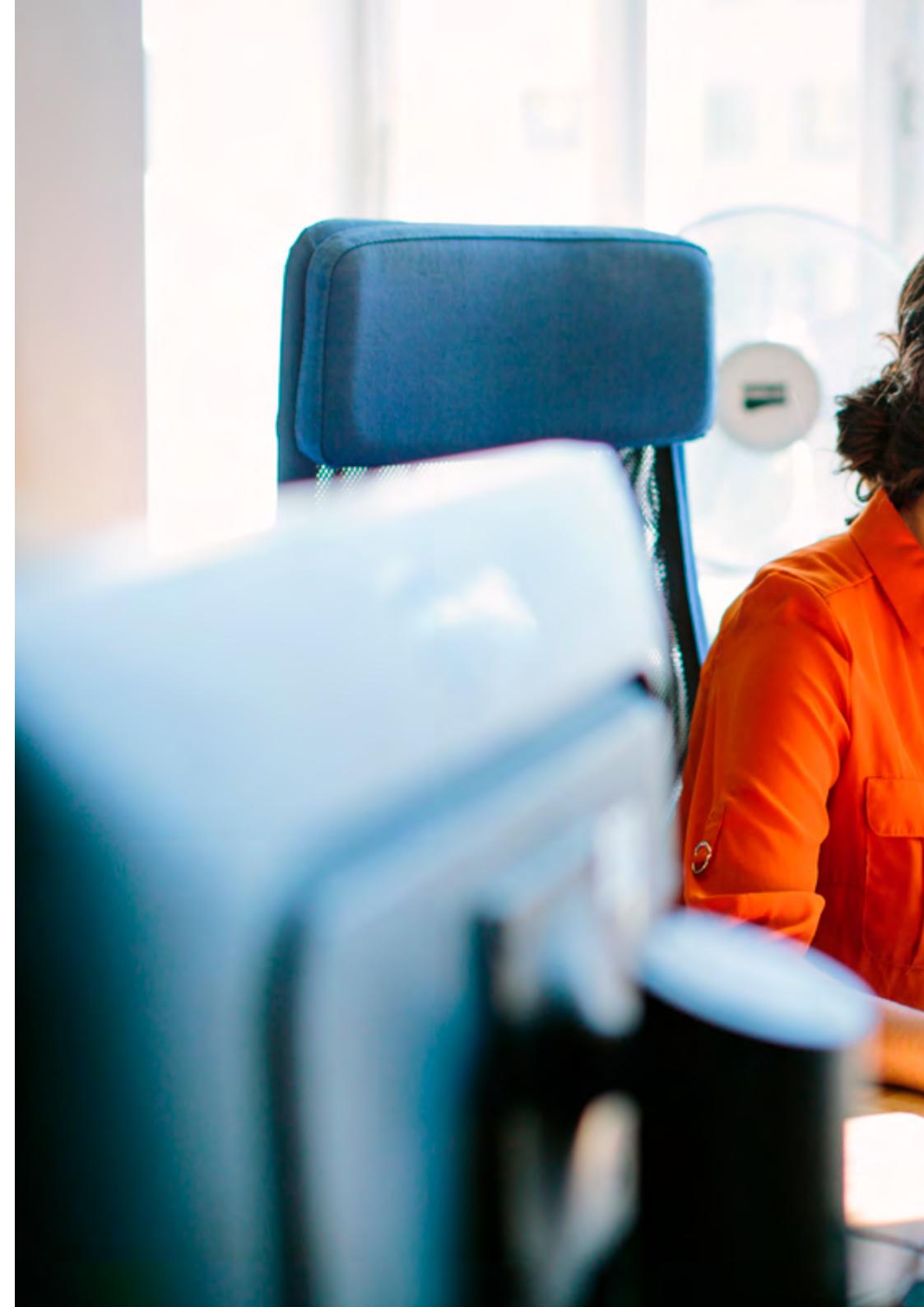


Competenze generali

- ◆ Sviluppare competenze pratiche di Programmazione
- ◆ Acquisire competenze avanzate nel controllo del flusso dei programmi
- ◆ Gestire la progettazione e l'implementazione di Software Orientato agli Oggetti
- ◆ Diventare esperti nello sviluppo completo di applicazioni Python
- ◆ Dominare la progettazione e la gestione di applicazioni web e mobili
- ◆ Padroneggiare la progettazione dell'interfaccia utente e dell'esperienza utente utilizzando Python
- ◆ Sviluppare competenze nella gestione e nell'analisi dei dati con Python
- ◆ Acquisire competenze nella gestione avanzata dei file e nella modellazione in Python
- ◆ Sviluppare competenze nella gestione avanzata dei dati con NumPy e Pandas
- ◆ Approfondire la gestione avanzata dei dati con NumPy e Pandas

“

Affronterai sfide reali ed eccellerai in un mercato del lavoro competitivo e in continua evoluzione. Scegli TECH!"





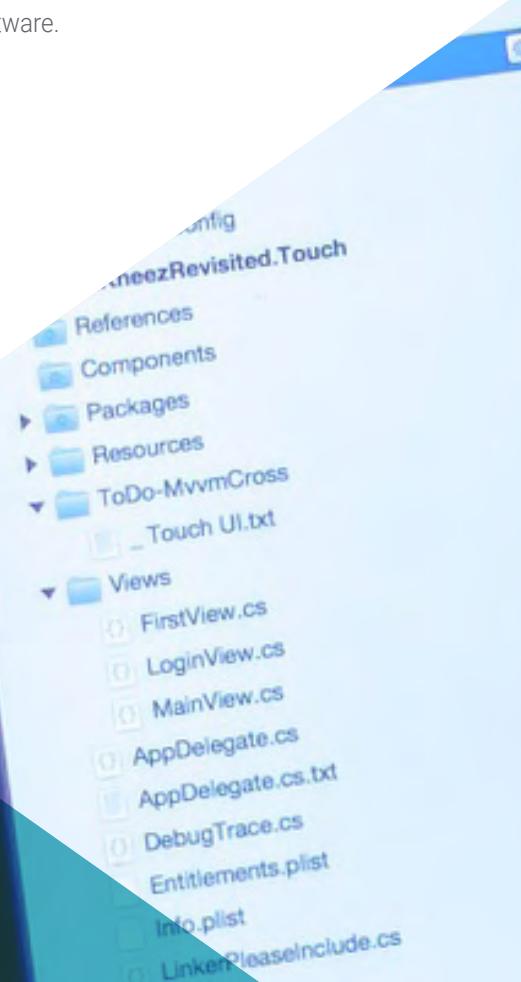
Competenze specifiche

- ◆ Imparare a gestire i dati in modo avanzato in Python
- ◆ Gestire l'uso avanzato delle funzioni in Python
- ◆ Implementare concetti avanzati di Programmazione Orientata agli Oggetti (POO) come classi astratte ed eccezioni personalizzate
- ◆ Controllare test e il *Debug*
- ◆ Sviluppare API e servizi web
- ◆ Padroneggiare l'uso di strumenti di progettazione UI/UX con Python
- ◆ Utilizzare le librerie di dati essenziali in Python
- ◆ Applicare in modo pratico diverse strutture di dati
- ◆ Utilizzare Pandas per la gestione di dati strutturati
- ◆ Promuovere competenze nell'analisi e nella manipolazione di serie temporali e dati complessi

04

Direzione del corso

I docenti di questo Master sono esperti del settore, con una vasta esperienza pratica nello sviluppo di Software e una comprovata esperienza di progetti di successo. Coniugando la teoria con l'applicazione pratica, questi tutor non solo impartiranno conoscenze avanzate su Python, ma condivideranno anche la loro esperienza nel mondo reale, fornendo agli studenti una preziosa visione delle migliori pratiche e delle tendenze attuali del settore. Inoltre, il suo approccio pedagogico si distingue per l'impegno a fornire ai laureati la preparazione tecnica più aggiornata per eccellere nel dinamico campo dello sviluppo Software.



```
1 // BeezKneezRevisited.Core
2
3 public class MainViewModel : MvxViewModel
4 {
5     public MainViewModel ()
6     {
7     }
8
9
10    private string _hello = "Hello MOFO";
11
12    public string Hello
13    {
14        get { return _hello; }
15        set { set/_hello = value; RaisePropertyChanged("Hello"); }
16    }
17
18    }
19
20 }
21
22 }
```

The image shows a code editor with C# code for a 'MainViewModel' class. The code uses the MVVM pattern with MvvmCross. It includes a private field '_hello' and a public property 'Hello' with both get and set accessors. The set accessor triggers a 'RaisePropertyChanged' event with the name 'Hello'. A tooltip for the 'SetProperty' method is visible on the right.



A screenshot of a code editor window showing a portion of a C# class definition. The code includes a constructor that calls a base constructor with a parameter of type Hello, and a protected method SetProperty that takes a ref T storage, a T value, and a string propertyName, with a null default for propertyName.

```
    PropertyChangedEventArgs() : base("Hello") { }

    protected bool SetProperty<T> (ref T storage,
        T value,
        string propertyName = null)
```

“

Avrai successo con i migliori e acquisirai le conoscenze e le competenze necessarie per intraprendere il settore dello Sviluppo in Python”

Direzione



Dott. Matos Rodríguez, Dionis

- *Data Engineer* presso Wide Agency Sadexo
- *Data Consultant* presso Tokiota
- *Data Engineer* presso Devoteam
- *BI Developer* presso Ibermática
- *Applications Engineer* presso Johnson Controls
- *Database Developer* presso Suncapital España
- *Senior Web Developer* presso Deadlock Solutions
- *QA Analyst* presso Metaconzept
- Master in *Big Data & Analytics* presso EAE Business School
- Master in Analisi e Progettazione di Sistemi
- Laurea in Ingegneria Informatica presso l'Università APEC

Personale docente

Dott. Villar Valor, Javier

- ◆ Direttore e socio fondatore di Impulsa2
- ◆ *Chief Operations Officer (COO)* presso Summa Insurance Brokers
- ◆ Direttore della trasformazione e dell'eccellenza professionale presso Johnson Controls
- ◆ Master in *Coaching Profesional*
- ◆ Executive MBA conseguito presso Emylon Business School, Francia
- ◆ Master in Gestione della Qualità presso EOI
- ◆ Ingegneria Informatica presso l'Universidad Acción Pro-Educación y Cultura (UNAPEC)

Dott. Gil Contreras, Armando

- ◆ *Lead Big Data Scientist* presso Jhonson Controls
- ◆ *Data Scientist-Big Data* presso Opensistemas S.A.
- ◆ Revisore dei fondi in Creatività e Tecnologia S.A. (CYTSA)
- ◆ Revisore del settore pubblico presso PricewaterhouseCoopers Auditores
- ◆ Master in *Data Science* presso il Centro Universitario di Tecnología e Arte
- ◆ Master MBA in Relazioni Internazionali e Business presso il Centro di Studi Finanziari (CEF)
- ◆ Laurea in Economia presso l'Istituto Tecnologico di Santo Domingo

Dott.ssa Gil Contreras, Milagros

- ◆ Content Creator in MPCTech LLC
- ◆ Manager di Progetti
- ◆ Freelance *IT Writer*
- ◆ MBA presso l'Università Complutense di Madrid
- ◆ Diplomata/Laureata in Amministrazione di Imprese presso l'Istituto Tecnologico di Santo Domingo

Dott.ssa Delgado Feliz, Benedit

- ◆ Assistente Amministrativa e Operatrice di Sorveglianza Elettronica presso la Direzione Nazionale del Dipartimento di controllo della droga (DNCD)
- ◆ Servizio Clienti a Cáceres e Attrezzature
- ◆ Reclami e servizio clienti presso Express Parcel Services (EPS)
- ◆ Specialista in Microsoft Office presso la Scuola Nazionale di Informatica
- ◆ Comunicatrice Sociale dell'Università Cattolica di Santo Domingo

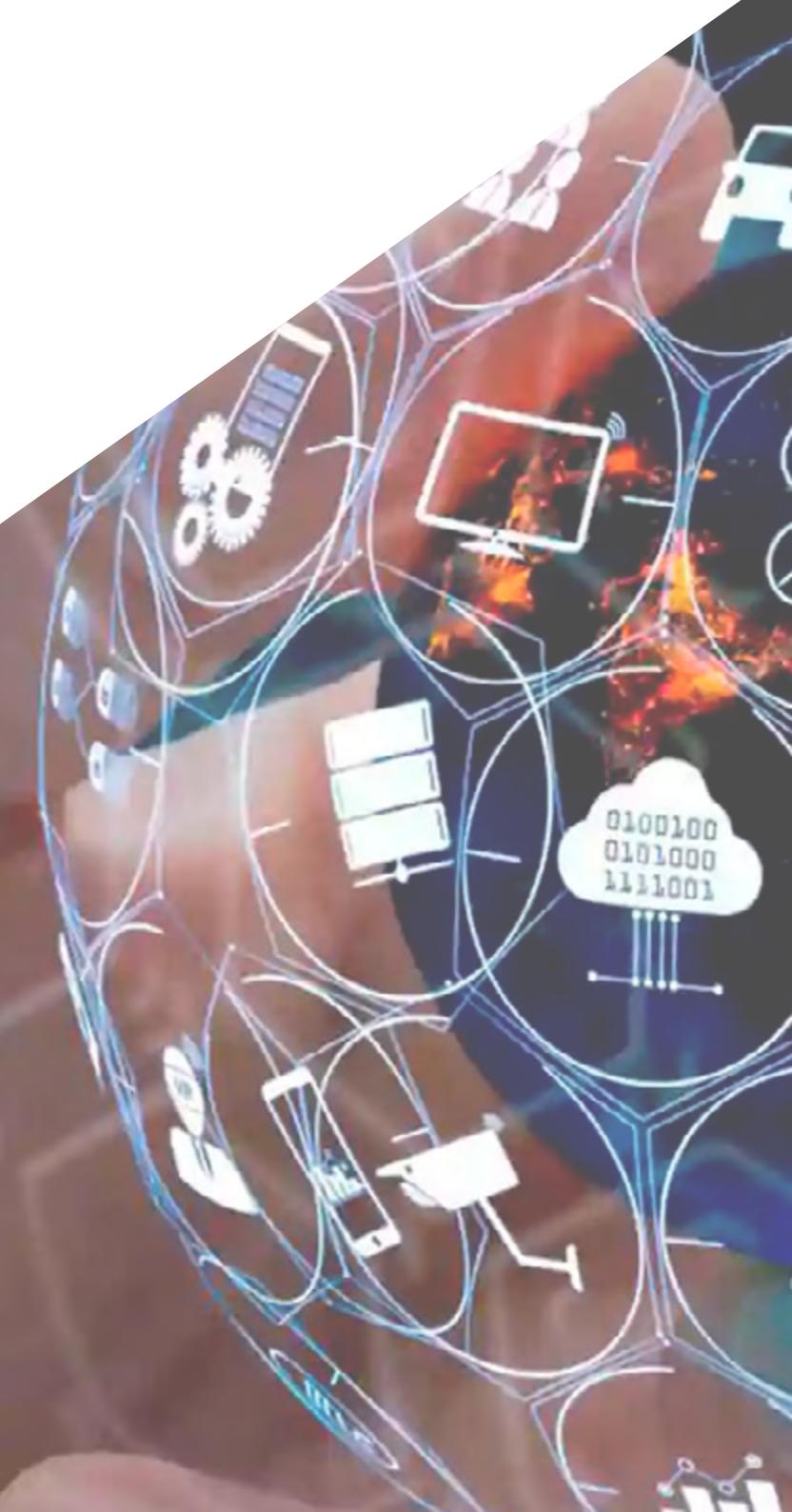
“

Cogli l'occasione per conoscere gli ultimi sviluppi in materia e applicali alla tua pratica quotidiana”

05

Struttura e contenuti

Il programma è stato meticolosamente progettato per fornire un'esperienza completa ai laureati. Da un'immersione profonda nella sintassi e nelle funzionalità avanzate di Python, alla specializzazione nella Programmazione Orientata agli Oggetti (POO), alla progettazione di applicazioni web e mobile, alla gestione esperta dei dati con librerie come NumPy e Pandas, ogni modulo sarà attentamente strutturato per fornire solide conoscenze e competenze pratiche. Inoltre, verranno esplorati argomenti cruciali come la progettazione dell'interfaccia utente e dell'esperienza utente (UI/UX), la gestione avanzata dei dati e l'ottimizzazione delle prestazioni e dello storage.





66

*Non solo acquisirai una padronanza approfondita
in Python, ma ti preparerai anche ad affrontare con
sicurezza le sfide dinamiche del settore tecnologico”*

Modulo 1. Programmazione in Python

- 1.1. Creazione ed esecuzione di programmi in Python
 - 1.1.1. Configurazione degli ambienti di sviluppo
 - 1.1.2. Esecuzione scripts en Python
 - 1.1.3. Strumenti di Sviluppo Integrato (IDE)
- 1.2. Dati in Python
 - 1.2.1. Tipi primitivi (*int, float, str*)
 - 1.2.2. Conversione e *casting* dei tipi di dati in Python
 - 1.2.3. Immutabilità e memorizzazione dei dati in Python
- 1.3. Riferimenti agli oggetti in Python
 - 1.3.1. Riferimenti in memoria
 - 1.3.2. Identità v Uguaglianza
 - 1.3.3. Gestione dei referral e raccolta dei rifiuti
- 1.4. Raccolta dati in Python
 - 1.4.1. Elenchi e operazioni comuni
 - 1.4.2. Le tuple e la loro immutabilità
 - 1.4.3. Dizionari e accesso ai dati
- 1.5. Operazioni logiche in Python
 - 1.5.1. Operatori booleani
 - 1.5.2. Espressioni condizionali
 - 1.5.3. *Short-Circuit Evaluation*
- 1.6. Operazioni aritmetiche in Python
 - 1.6.1. Operazioni aritmetiche in Python
 - 1.6.2. Operatori di divisione
 - 1.6.3. Precedenza e associatività
- 1.7. Ingresso/uscita in Python
 - 1.7.1. Lettura di dati da input standard
 - 1.7.2. Scrittura di dati su standard output
 - 1.7.3. Gestione dei file
- 1.8. Creazione e chiamata di funzioni in Python
 - 1.8.1. Sintassi delle funzioni
 - 1.8.2. Parametri e argomenti
 - 1.8.3. Valori di ritorno e funzioni anonime



- 1.9. Uso di *strings* in Python
 - 1.9.1. Manipolazione e formattazione delle *strings*
 - 1.9.2. Metodi comuni di *strings*
 - 1.9.3. Interpolazione e *F-strings*
- 1.10. Gestione degli errori e delle eccezioni in Python
 - 1.10.1. Tipi comuni di eccezioni
 - 1.10.2. Blocchi *try-except*
 - 1.10.3. Creare eccezioni personalizzate

Modulo 2. Dati avanzati e controllo del flusso con Python

- 2.1. Identificatori e parole chiave in Python
 - 2.1.1. Regole per i nomi delle variabili
 - 2.1.2. Parole riservate in Python
 - 2.1.3. Convenzioni di denominazione
- 2.2. Tipi integrali e booleani in Python
 - 2.2.1. Tipi integrali
 - 2.2.2. Operazioni booleane specifiche
 - 2.2.3. Conversioni e rappresentazioni
- 2.3. Tipi in virgola mobile e numeri complessi in Python
 - 2.3.1. Precisione e rappresentazione
 - 2.3.2. Operazioni in virgola mobile
 - 2.3.3. Uso dei numeri complessi nei calcoli
- 2.4. Formattazione e codifica delle *strings* in Python
 - 2.4.1. Metodi di formattazione avanzatiMetodologia formattazione dei avanzata
 - 2.4.2. Codifiche *Unicode* e UTF-8
 - 2.4.3. Lavoro con i caratteri speciali
- 2.5. Collezioni: Tuple, Elenchi e Dizionari in Python
 - 2.5.1. Confronto e contrasto tra tipi
 - 2.5.2. Metodi specifici per ogni tipo
 - 2.5.3. Efficienza e selezione del tipo adeguato
- 2.6. Sets y Frozen Sets in Python
 - 2.6.1. Creazione e operazioni in Sets
 - 2.6.2. Frozen Sets
 - 2.6.3. Applicazioni pratiche e prestazioni

- 2.7. Iterazione e copia di collezioni in Python
 - 2.7.1. Loop *for* e comprensione delle liste
 - 2.7.2. Copia superficiale vs. Profonda
 - 2.7.3. Iteratori e generatori
- 2.8. Uso delle funzioni *Lambda* in Python
 - 2.8.1. Sintassi e creazione di funzioni *Lambda*
 - 2.8.2. Applicazioni in filtri e mappe
 - 2.8.3. Limitazioni e buone pratiche
- 2.9. Strutture di controllo: Condizionali e loop in Python
 - 2.9.1. Strutture *if-else* y *elif*
 - 2.9.2. Loop *while* e *for*
 - 2.9.3. Controllo di flusso con *break*, *continue* e *else*
- 2.10. Funzioni e metodi in Python avanzati
 - 2.10.1. Funzioni ricorsive
 - 2.10.2. Funzioni di ordine superiore
 - 2.10.3. Decoratori di funzioni

Modulo 3. Programmazione Orientata agli Oggetti (POO) in Python

- 3.1. Programmazione Orientata agli Oggetti (POO) in Python
 - 3.1.1. Classi e oggetti
 - 3.1.2. Incapsulamento e astrazione
 - 3.1.3. Programmazione Orientata agli Oggetti (POO) in Python
- 3.2. Creare classi e oggetti in Python
 - 3.2.1. Le classi in Python POO
 - 3.2.2. Metodi di istanziazione e inizializzazione
 - 3.2.3. Attributi e metodi
- 3.3. Attributi e metodi in Python
 - 3.3.1. Attributi di istanza vs. Classe
 - 3.3.2. Metodi di istanza, di classe e statici
 - 3.3.3. Incapsulamento e occultamento delle informazioni
- 3.4. Eredità e il polimorfismo in Python
 - 3.4.1. Eredità singola e multipla
 - 3.4.2. Sovrascrittura ed estensione dei metodi
 - 3.4.3. Polimorfismo e *Duck Typing*

- 3.5. Proprietà e accesso agli attributi in Python
 - 3.5.1. Getters e Setters
 - 3.5.2. Decoratore `@property`
 - 3.5.3. Controllo dell'accesso e validazione
- 3.6. Classi e collezioni personalizzate in Python
 - 3.6.1. Creazione di tipi di collezioni
 - 3.6.2. Metodi speciali (`__len__`, `__getitem__`)
 - 3.6.3. Iteratori personalizzati
- 3.7. Aggregazione e composizione nelle classi in Python
 - 3.7.1. Relazioni tra lezioni
 - 3.7.2. Aggregazione vs. Composizione
 - 3.7.3. Gestione del ciclo di vita obiettivi
- 3.8. Uso dei decoratori nelle classi in Python
 - 3.8.1. Decoratori per metodi
 - 3.8.2. Decoratori di classe
 - 3.8.3. Applicazioni e casi d'uso
- 3.9. Classi e metodi astratti in Python
 - 3.9.1. Classi astratte
 - 3.9.2. Metodi astratti e implementazione
 - 3.9.3. Uso dell'ABC (*Abstract Base Class*)
- 3.10. Eccezioni e gestione degli errori in Python POO
 - 3.10.1. Eccezioni personalizzate nelle classi
 - 3.10.2. Gestione delle eccezioni nei metodi
 - 3.10.3. Buone pratiche in materia di eccezioni e POO
- 4.3. Gestione delle dipendenze e librerie in Python
 - 4.3.1. Gestione dei pacchetti con Pip
 - 4.3.2. Utilizzo di ambienti virtuali
 - 4.3.3. Risolvere i conflitti di dipendenza
- 4.4. Modelli di progettazione nello sviluppo in Python
 - 4.4.1. Modelli creativi, strutturali e comportamentali
 - 4.4.2. Applicazione pratica dei modelli
 - 4.4.3. Refactoring e modelli
- 4.5. Test e Debugging nelle applicazioni in Python
 - 4.5.1. Strategie di *Testing* (Unitario, Integrazione")
 - 4.5.2. Uso di Frameworks di prova
 - 4.5.3. Tecniche e strumenti di Debugging
- 4.6. Sicurezza e autenticazione in Python
 - 4.6.1. Sicurezza in applicazioni
 - 4.6.2. Implementazione dell'autenticazione e autorizzazione
 - 4.6.3. Prevenzione delle vulnerabilità
- 4.7. Ottimizzazione e prestazioni delle applicazioni in Python
 - 4.7.1. Analisi delle prestazioni
 - 4.7.2. Tecniche di ottimizzazione del codice
 - 4.7.3. Gestione efficiente delle risorse e dei dati
- 4.8. Implementazione e distribuzione delle applicazioni in Python
 - 4.8.1. Strategie di implementazione
 - 4.8.2. Uso di container e orchestratori
 - 4.8.3. Distribuzione e aggiornamenti continui
- 4.9. Manutenzione e aggiornamenti in Python
 - 4.9.1. Gestione del ciclo di vita del Software
 - 4.9.2. Strategie di manutenzione e refactoring
 - 4.9.3. Aggiornamento e migrazione dei sistemi
- 4.10. Documentazione e supporto in Python
 - 4.10.1. Creare documentazione efficacemente
 - 4.10.2. Strumenti di documentazione
 - 4.10.3. Strategie per il supporto e la comunicazione con gli utenti

Modulo 4. Sviluppo delle applicazioni in Python

- 4.1. Architettura dell'applicazione in Python
 - 4.1.1. Progettazione del Software
 - 4.1.2. Modelli architettonici comuni
 - 4.1.3. Valutazione dei requisiti e delle esigenze
- 4.2. Progettazione e modellazione di applicazioni Python
 - 4.2.1. Uso di UML e diagrammi
 - 4.2.2. Modellazione del flusso di dati e informazioni
 - 4.2.3. Principi SOLID e progettazione modulare

Modulo 5. Sviluppo web e mobile con Python

- 5.1. Sviluppo web con Python
 - 5.1.1. Struttura e componenti di una web
 - 5.1.2. Tecnologie di sviluppo web
 - 5.1.3. Tendenze nello sviluppo web
- 5.2. Framework web più diffusi Cono Python
 - 5.2.1. *Django, Flask* e altre opzioni
 - 5.2.2. Confronto e selezione dei Frameworks
 - 5.2.3. Integrazione con *Frontend*
- 5.3. Sviluppatore *FrontEnd*: HTML, CSS y JavaScript con Python
 - 5.3.1. HTML e CSS
 - 5.3.2. JavaScript e manipolazione del DOM
 - 5.3.3. Frameworks e biblioteche *Frontend*
- 5.4. Backend e database con Python
 - 5.4.1. Sviluppo di *Backend* con Python
 - 5.4.2. Gestione di database relazionali e non relazionali
 - 5.4.3. Integrazione *Backend-Frontend*
- 5.5. API e servizi web con Python
 - 5.5.1. Progettazione di API *RESTful*
 - 5.5.2. Implementazione e documentazione di API
 - 5.5.3. Consumo e sicurezza delle API
- 5.6. Sviluppo mobile con Python
 - 5.6.1. Piattaforme di sviluppo mobile (Native, Ibride)
 - 5.6.2. Strumenti e ambienti di sviluppo
 - 5.6.3. Adattamento di applicazioni per dispositivi mobili
- 5.7. Piattaforme di sviluppo mobile con Python
 - 5.7.1. Android e IOS
 - 5.7.2. Frameworks per lo sviluppo incrociato
 - 5.7.3. Test e Deployment su dispositivi mobili
- 5.8. Design e UX nelle applicazioni mobili con Python
 - 5.8.1. Progettazione dell'interfaccia mobile
 - 5.8.2. Usability utente ed esperienza utente con Python
 - 5.8.3. Strumenti di prototipazione e progettazione
- 5.9. Test e depurazione nelle applicazioni in Python
 - 5.9.1. Strategie di *Testing* su dispositivi mobili
 - 5.9.2. Strumenti di depurazione e monitoraggio
 - 5.9.3. Automazione dei test
- 5.10. Pubblicazione nei negozi di applicazioni con Python
 - 5.10.1. Processo di pubblicazione su App Store e Google Play
 - 5.10.2. Conformità e politiche delle app
 - 5.10.3. Strategie di Marketing e promozione

Modulo 6. Interfaccia utente ed esperienza utente con Python

- 6.1. Progettazione dell'interfaccia utente con Python
 - 6.1.1. Disegno UI con Python
 - 6.1.2. Interazione utente-computer con Python
 - 6.1.3. Progettazione centrata sull'utente con Python
- 6.2. Strumenti di progettazione UI/UX con Python
 - 6.2.1. Software di progettazione e prototipazione
 - 6.2.2. Strumenti di collaborazione e *Feedback*
 - 6.2.3. Integrazione della progettazione nel processo di sviluppo
- 6.3. Progettazione reattiva e adattiva con Python
 - 6.3.1. Tecniche di progettazione reattiva
 - 6.3.2. Adattamento a diversi dispositivi e schermi
 - 6.3.3. Test e garanzia della qualità
- 6.4. Animazioni e transizioni con Python
 - 6.4.1. Creare animazioni efficaci con Python
 - 6.4.2. Strumenti e librerie per le animazioni
 - 6.4.3. Impatto su UX e prestazioni
- 6.5. Accessibilità e usabilità con Python
 - 6.5.1. Accessibilità web
 - 6.5.2. Strumenti e tecniche di valutazione
 - 6.5.3. Implementazione delle migliori pratiche
- 6.6. Prototipazione e *Wireframes* con Python
 - 6.6.1. Creazione di *Wireframes* e Mockups
 - 6.6.2. Strumenti di prototipazione rapido
 - 6.6.3. Test di usabilità e *Feedback*

- 6.7. Test di usabilità con Python
 - 6.7.1. Metodi e tecniche di test di usabilità
 - 6.7.2. Analisi e miglioramento basati sui risultati
 - 6.7.3. Strumenti per il test di usabilità
- 6.8. Analisi dell'interfaccia utente con Python
 - 6.8.1. Tecniche di analisi e *Tracking*
 - 6.8.2. Interpretazione dei dati e alle metriche
 - 6.8.3. Miglioramento continua basata sui dati
- 6.9. Miglioramenti basati sul *Feedback* con Python
 - 6.9.1. Gestione e analisi del *Feedback*
 - 6.9.2. Cicli *Feedback* e miglioramento continuo
 - 6.9.3. Strategie per l'attuazione di un cambiamento efficace
- 6.10. Tendenze future dell'UI/UX con Python
 - 6.10.1. Innovazioni e tendenze emergenti
 - 6.10.2. Impatto delle nuove tecnologie nell' UI/ UX
 - 6.10.3. Prepararsi al futuro del design

Modulo 7. Elaborazione dei dati e *Big Data* con Python

- 7.1. Uso di Python sui dati
 - 7.1.1. Python nella scienza dei dati e nell'analisi
 - 7.1.2. Librerie di dati essenziali
 - 7.1.3. Applicazioni ed esempi
- 7.2. Configurazione degli ambienti di sviluppo con Python
 - 7.2.1. Installazione e strumenti Python
 - 7.2.2. Configurazione di ambienti virtuali
 - 7.2.3. Strumenti di Sviluppo Integrato (IDE)
- 7.3. Variabili, tipi di dati e operatori in Python
 - 7.3.1. Variabili e tipi di dati primitivi
 - 7.3.2. Struttura dei dati
 - 7.3.3. Operatori aritmetici e logici
- 7.4. Controllo del Flusso: Condizionali e loop
 - 7.4.1. Strutture di controllo condizionali (*if, else, elif*)
 - 7.4.2. Loops (*for, while*) e controllo di flusso
 - 7.4.3. Comprensione di liste ed espressioni generatrici

- 7.5. Funzioni e modularità con Python
 - 7.5.1. Uso delle funzioni
 - 7.5.2. Parametri, argomenti e valori di ritorno
 - 7.5.3. Modularità e riutilizzo del codice
- 7.6. Gestione degli errori e delle eccezioni con Python
 - 7.6.1. errori ed eccezioni
 - 7.6.2. Gestione delle eccezioni con `try-except`
 - 7.6.3. Creare eccezioni personalizzate
- 7.7. Strumento *IPython*
 - 7.7.1. Strumento *IPython*
 - 7.7.2. Uso degli *IPython* per analisi dei dati
 - 7.7.3. Differenze con l'interprete *Python* standard
- 7.8. *Jupyter Notebooks*
 - 7.8.1. *Jupyter Notebooks*
 - 7.8.2. Uso dei quaderni per analisi dei dati
 - 7.8.3. Pubblicazione dei quaderni *Jupyter*
- 7.9. Migliori pratiche di codifica Python
 - 7.9.1. Stile e convenzioni (PEP 8)
 - 7.9.2. Documentazione e commenti
 - 7.9.3. Strategie di test e debug
- 7.10. Risorse e comunità Python
 - 7.10.1. Risorse e documentazione online
 - 7.10.2. Comunità e forum
 - 7.10.3. Apprendimento e aggiornamenti in Python

Modulo 8. Strutture dei dati e funzioni in Python

- 8.1. Gli insiemi in Python
 - 8.1.1. Operazioni e metodi
 - 8.1.2. Differenze e applicazioni pratiche
 - 8.1.3. Iterazione e comprensione
- 8.2. Dizionari e loro utilizzo in Python
 - 8.2.1. Creazione e manipolazione di dizionari
 - 8.2.2. Accesso e gestione dei dati
 - 8.2.3. Modelli e tecniche avanzate

- 8.3. Comprensione di liste e dizionari in Python
 - 8.3.1. Sintassi ed esempi
 - 8.3.2. Efficienza e leggibilità
 - 8.3.3. Applicazioni pratiche
- 8.4. Funzioni sui dati in Python
 - 8.4.1. Creazione di funzioni
 - 8.4.2. Campo di applicazione e spazio dei nomi
 - 8.4.3. Funzioni anonime e *Lambda*
- 8.5. Argomenti delle funzioni e valori di ritorno in Python
 - 8.5.1. Argomenti posizionali e nominati
 - 8.5.2. Valori di ritorno multipli
 - 8.5.3. Argomenti variabili e parole chiave
- 8.6. Funzioni *Lambda* e funzioni di ordine superiore in Python
 - 8.6.1. Uso delle funzioni *Lambda*
 - 8.6.2. Funzioni *Map*, *Filter* y *Reduce*
 - 8.6.3. Applicazioni nell'elaborazione dei dati
- 8.7. Gestione dei file in Python
 - 8.7.1. Lettura e scrittura di file
 - 8.7.2. Gestione di file binari e di testo
 - 8.7.3. Migliori pratiche e gestione delle eccezioni
- 8.8. Lettura e scrittura di file di testo e binari in Python
 - 8.8.1. Formati e codifica dei file
 - 8.8.2. Gestione di file di grandi dimensioni
 - 8.8.3. Serializzazione e deserializzazione (*JSON*, *pickle*)
- 8.9. Contesti e operazioni sui file
 - 8.9.1. Utilizzo del gestore di contesti (*with*)
 - 8.9.2. Tecniche di elaborazione dei file
 - 8.9.3. Sicurezza e gestione degli errori
- 8.10. Librerie di modellazione Python
 - 8.10.1. *Scikit-learn*
 - 8.10.2. *TensorFlow*
 - 8.10.3. *PyTorch*

Modulo 9. Gestione dei dati in Python con NumPy e Pandas

- 9.1. Creazione e manipolazione Arrays in NumPy
 - 9.1.1. NumPy
 - 9.1.2. Operazioni di base con *Array*
 - 9.1.3. Manipolazione e trasformazione di *Array*
- 9.2. Operazioni di vettorializzazione con *Array*
 - 9.2.1. Vettorizzazione
 - 9.2.2. Funzioni universali (*ufunc*)
 - 9.2.3. Efficienza e prestazioni
- 9.3. Indicizzazione e segmentazione in NumPy
 - 9.3.1. Accesso agli elementi e *Slicing*
 - 9.3.2. Indicizzazione avanzata e booleana
 - 9.3.3. Riordino e selezione
- 9.4. Serie Pandas e *DataFrames*
 - 9.4.1. Pandas
 - 9.4.2. Strutture dati in Pandas
 - 9.4.3. Manipolazione dei *DataFrames*
- 9.5. Indicizzazione e selezione in Pandas
 - 9.5.1. Accesso ai dati in serie e *DataFrames*
 - 9.5.2. Metodi di selezione e filtraggio
 - 9.5.3. Uso di *loc* e *iloc*
- 9.6. Operazioni con Pandas
 - 9.6.1. Operazioni aritmetiche e allineamento
 - 9.6.2. Funzioni di aggregazione e statistiche
 - 9.6.3. Trasformazioni e applicazione di funzioni
- 9.7. Gestione di dati incompleti in Pandas
 - 9.7.1. Rilevamento e gestione dei valori nulli
 - 9.7.2. Riempimento e cancellazione di dati incompleti
 - 9.7.3. Strategie per la gestione dei dati incompleti
- 9.8. Funzioni e applicazioni in Pandas
 - 9.8.1. Concatenazione e unione di dati
 - 9.8.2. Raggruppamento e aggregazione (*groupby*)
 - 9.8.3. *Pivot Tables* y *Crosstabs*

- 9.9. Visualizzazione con Matplotlib
 - 9.9.1. Matplotlib
 - 9.9.2. Creazione e personalizzazione dei grafici
 - 9.9.3. Integrazione con Pandas
- 9.10. Personalizzazione dei grafici in Matplotlib
 - 9.10.1. Stili e configurazioni
 - 9.10.2. Grafici avanzati (*scatter, bar, ecc.*)
 - 9.10.3. Creare visualizzazioni complesse

Modulo 10. Tecniche avanzate e applicazioni pratiche in NumPy e Pandas

- 10.1. Caricamento dei dati da diverse fonti
 - 10.1.1. Importazione da CSV, Excel e database
 - 10.1.2. Lettura di dati da API e web
 - 10.1.3. Strategie per la gestione dei big data
- 10.2. Archiviazione di dati in Python
 - 10.2.1. Esportazione in diversi formati
 - 10.2.2. Efficienza di memorizzazione
 - 10.2.3. Sicurezza e privacy dei dati
- 10.3. Strategie di pulizia dei dati in Python
 - 10.3.1. Identificare e correggere le incongruenze
 - 10.3.2. Normalizzazione e trasformazione dei dati
 - 10.3.3. Automazione dei processi di pulizia
- 10.4. Trasformazione avanzata dei dati in Pandas
 - 10.4.1. Tecniche di manipolazione e trasformazione
 - 10.4.2. Combinazione e ristrutturazione di *DataFrames*
 - 10.4.3. Uso di espressioni regolari in Pandas
- 10.5. Combinazione di *DataFrames* in Pandas
 - 10.5.1. *Merge, Join* e concatenazione
 - 10.5.2. Gestione dei conflitti e chiavi
 - 10.5.3. Strategie di combinazione efficienti
- 10.6. Trasformazione avanzata pivot dei dati in Pandas
 - 10.6.1. *Pivot* e *Melt*
 - 10.6.2. Tecniche di *Reshape* e trasposizione
 - 10.6.3. Applicazioni nell' analisi dei dati
- 10.7. Serie temporali in Pandas
 - 10.7.1. Gestione dei tempi e delle scadenze
 - 10.7.2. *Resampling* e *Window Functions*
 - 10.7.3. Analisi delle tendenze e della stagionalità
- 10.8. Uso avanzato dell'indice in Pandas
 - 10.8.1. Indici multilivello e gerarchici
 - 10.8.2. Selezione e manipolazione avanzata
 - 10.8.3. Ottimizzazione delle query
- 10.9. Strategie di ottimizzazione di prestazioni
 - 10.9.1. Miglioramento della velocità e dell'efficienza
 - 10.9.2. Uso di Cython e Numba
 - 10.9.3. Parallelizzazione ed elaborazione distribuita
- 10.10. Progetti pratici di manipolazione dei dati
 - 10.10.1. Sviluppo di esempi reali
 - 10.10.2. Integrazione di tecniche Python
 - 10.10.3. Strategie per risolvere problemi di dati complessi

“

Questo programma innovativo si adatterà a te per tenerti aggiornato sulle ultime tendenze e tecnologie, assicurandoti di essere all'avanguardia nell'innovazione dello sviluppo del Software”

06

Metodologia

Questo programma ti offre un modo differente di imparare. La nostra metodologia si sviluppa in una modalità di apprendimento ciclico: *il Relearning*.

Questo sistema di insegnamento viene applicato nelle più prestigiose facoltà di medicina del mondo ed è considerato uno dei più efficaci da importanti pubblicazioni come il *New England Journal of Medicine*.



66

Scopri il Relearning, un sistema che abbandona l'apprendimento lineare convenzionale, per guidarti attraverso dei sistemi di insegnamento ciclici: una modalità di apprendimento che ha dimostrato la sua enorme efficacia, soprattutto nelle materie che richiedono la memorizzazione”

Caso di Studio per contestualizzare tutti i contenuti

Il nostro programma offre un metodo rivoluzionario per sviluppare le abilità e le conoscenze. Il nostro obiettivo è quello di rafforzare le competenze in un contesto mutevole, competitivo e altamente esigente.

“

Con TECH potrai sperimentare un modo di imparare che sta scuotendo le fondamenta delle università tradizionali in tutto il mondo"



Avrai accesso a un sistema di apprendimento basato sulla ripetizione, con un insegnamento naturale e progressivo durante tutto il programma.



Imparerai, attraverso attività collaborative e casi reali, la risoluzione di situazioni complesse in ambienti aziendali reali.

Un metodo di apprendimento innovativo e differente

Questo programma di TECH consiste in un insegnamento intensivo, creato ex novo, che propone le sfide e le decisioni più impegnative in questo campo, sia a livello nazionale che internazionale. Grazie a questa metodologia, la crescita personale e professionale viene potenziata, effettuando un passo decisivo verso il successo. Il metodo casistico, la tecnica che sta alla base di questi contenuti, garantisce il rispetto della realtà economica, sociale e professionale più attuali.

“

Il nostro programma ti prepara ad affrontare nuove sfide in ambienti incerti e a raggiungere il successo nella tua carriera”

Il Metodo Casistico è stato il sistema di apprendimento più usato nelle migliori Scuole di Informatica del mondo da quando esistono. Sviluppato nel 1912 affinché gli studenti di Diritto non imparassero la legge solo sulla base del contenuto teorico, il metodo casistico consisteva nel presentare loro situazioni reali e complesse per prendere decisioni informate e giudizi di valore su come risolverle. Nel 1924 fu stabilito come metodo di insegnamento standard ad Harvard.

Cosa dovrebbe fare un professionista per affrontare una determinata situazione?

Questa è la domanda con cui ti confrontiamo nel metodo dei casi, un metodo di apprendimento orientato all'azione. Durante il corso, gli studenti si confronteranno con diversi casi di vita reale. Dovranno integrare tutte le loro conoscenze, effettuare ricerche, argomentare e difendere le proprie idee e decisioni.

Metodologia Relearning

TECH coniuga efficacemente la metodologia del Caso di Studio con un sistema di apprendimento 100% online basato sulla ripetizione, che combina diversi elementi didattici in ogni lezione.

Potenziando il Caso di Studio con il miglior metodo di insegnamento 100% online: il Relearning.

Nel 2019 abbiamo ottenuto i migliori risultati di apprendimento di tutte le università online del mondo.

In TECH imparerai con una metodologia all'avanguardia progettata per formare i manager del futuro. Questo metodo, all'avanguardia della pedagogia mondiale, si chiama Relearning.

La nostra università è l'unica autorizzata a utilizzare questo metodo di successo. Nel 2019, siamo riusciti a migliorare il livello di soddisfazione generale dei nostri studenti (qualità dell'insegnamento, qualità dei materiali, struttura del corso, obiettivi...) rispetto agli indicatori della migliore università online.





Nel nostro programma, l'apprendimento non è un processo lineare, ma avviene in una spirale (impariamo, disimpariamo, dimentichiamo e re-impariamo). Pertanto, combiniamo ciascuno di questi elementi in modo concentrico. Questa metodologia ha formato più di 650.000 laureati con un successo senza precedenti in campi diversi come la biochimica, la genetica, la chirurgia, il diritto internazionale, le competenze manageriali, le scienze sportive, la filosofia, il diritto, l'ingegneria, il giornalismo, la storia, i mercati e gli strumenti finanziari. Tutto questo in un ambiente molto esigente, con un corpo di studenti universitari con un alto profilo socio-economico e un'età media di 43,5 anni.

Il Relearning ti permetterà di apprendere con meno sforzo e più performance, impegnandoti maggiormente nella tua specializzazione, sviluppando uno spirito critico, difendendo gli argomenti e contrastando le opinioni: un'equazione diretta al successo.

Dalle ultime evidenze scientifiche nel campo delle neuroscienze, non solo sappiamo come organizzare le informazioni, le idee, le immagini e i ricordi, ma sappiamo che il luogo e il contesto in cui abbiamo imparato qualcosa è fondamentale per la nostra capacità di ricordarlo e immagazzinarlo nell'ippocampo, per conservarlo nella nostra memoria a lungo termine.

In questo modo, e in quello che si chiama Neurocognitive Context-dependent E-learning, i diversi elementi del nostro programma sono collegati al contesto in cui il partecipante sviluppa la sua pratica professionale.

Questo programma offre i migliori materiali didattici, preparati appositamente per i professionisti:



Materiale di studio

Tutti i contenuti didattici sono creati appositamente per il corso dagli specialisti che lo impartiranno, per fare in modo che lo sviluppo didattico sia davvero specifico e concreto.

Questi contenuti sono poi applicati al formato audiovisivo che supporterà la modalità di lavoro online di TECH. Tutto questo, con le ultime tecniche che offrono componenti di alta qualità in ognuno dei materiali che vengono messi a disposizione dello studente.



Master class

Esistono evidenze scientifiche sull'utilità dell'osservazione di esperti terzi.

Imparare da un esperto rafforza la conoscenza e la memoria, costruisce la fiducia nelle nostre future decisioni difficili.



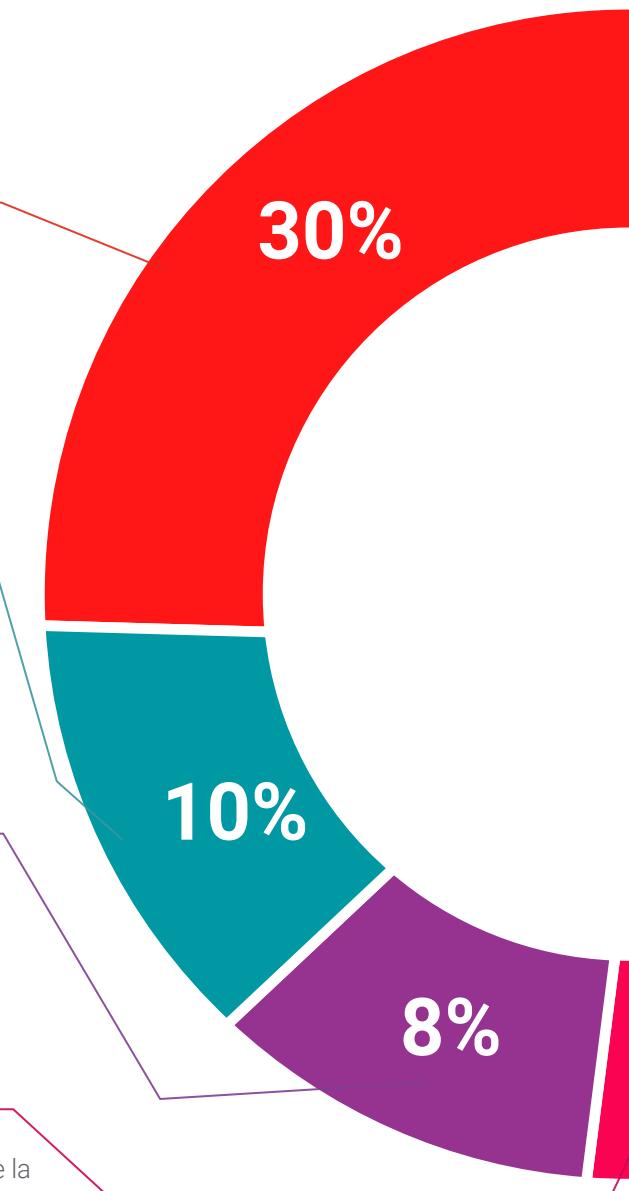
Pratiche di competenze e competenze

Svolgerai attività per sviluppare competenze e capacità specifiche in ogni area tematica. Pratiche e dinamiche per acquisire e sviluppare le competenze e le abilità che uno specialista deve sviluppare nel quadro della globalizzazione in cui viviamo.



Letture complementari

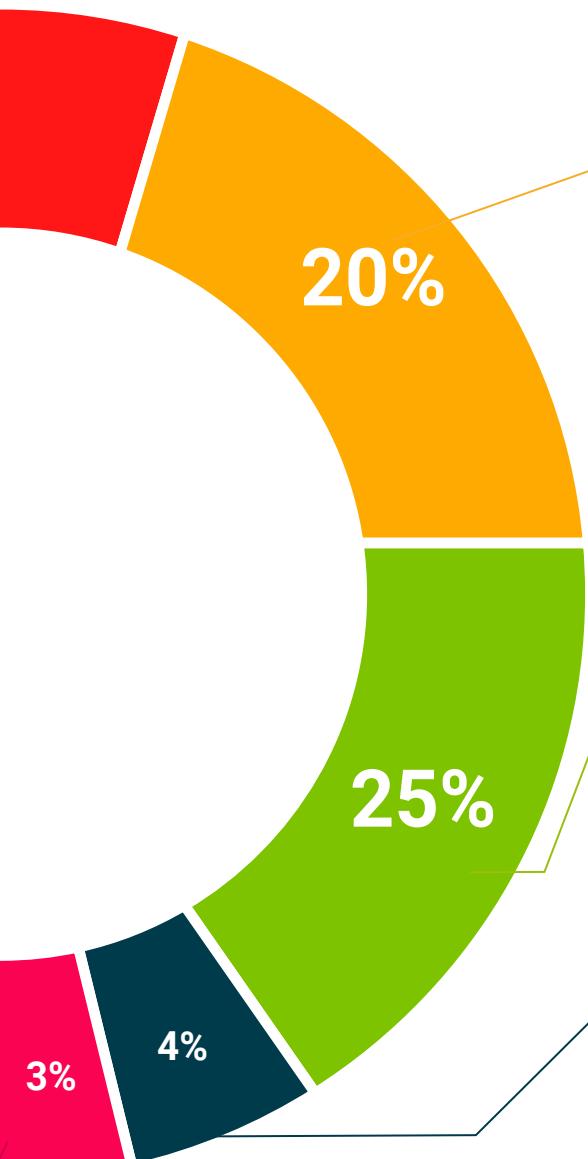
Articoli recenti, documenti di consenso e linee guida internazionali, tra gli altri. Nella biblioteca virtuale di TECH potrai accedere a tutto il materiale necessario per completare la tua specializzazione.



30%

10%

8%



Casi di Studio
Completerai una selezione dei migliori casi di studio scelti appositamente per questo corso. Casi presentati, analizzati e monitorati dai migliori specialisti del panorama internazionale.



Riepiloghi interattivi
Il team di TECH presenta i contenuti in modo accattivante e dinamico in pillole multimediali che includono audio, video, immagini, diagrammi e mappe concettuali per consolidare la conoscenza.



Questo esclusivo sistema di specializzazione per la presentazione di contenuti multimediali è stato premiato da Microsoft come "Caso di successo in Europa".

Testing & Retesting

Valutiamo e rivalutiamo periodicamente le tue conoscenze durante tutto il programma con attività ed esercizi di valutazione e autovalutazione, affinché tu possa verificare come raggiungi progressivamente i tuoi obiettivi.



07

Titolo

Questo programma ti consentirà di ottenere il titolo di studio di Master in Sviluppo in Python rilasciato da TECH Global University, la più grande università digitale del mondo.



66

*Porta a termine questo programma e
ricevi il tuo titolo universitario senza
sostamenti o fastidiose formalità"*

Questo programma ti consentirà di ottenere il titolo di studio di **Master in Sviluppo in Python** rilasciato da **TECH Global University**, la più grande università digitale del mondo.

TECH Global University è un'Università Ufficiale Europea riconosciuta pubblicamente dal Governo di Andorra ([bollettino ufficiale](#)). Andorra fa parte dello Spazio Europeo dell'Istruzione Superiore (EHEA) dal 2003. L'EHEA è un'iniziativa promossa dall'Unione Europea che mira a organizzare il quadro formativo internazionale e ad armonizzare i sistemi di istruzione superiore dei Paesi membri di questo spazio. Il progetto promuove valori comuni, l'implementazione di strumenti congiunti e il rafforzamento dei meccanismi di garanzia della qualità per migliorare la collaborazione e la mobilità tra studenti, ricercatori e accademici.



Questo titolo privato di **TECH Global University** è un programma europeo di formazione continua e aggiornamento professionale che garantisce l'acquisizione di competenze nella propria area di conoscenza, conferendo allo studente che supera il programma un elevato valore curriculare.

Titolo: Master in Sviluppo in Python

Modalità: online

Durata: 12 mesi

Accreditamento: 60 ECTS



futuro
salute fiducia persone
educazione informazione tutor
garanzia accreditamento insegnamento
istituzioni tecnologia apprendimento
comunità impegno
attenzione personalizzata innovazione
conoscenza presente qualità
formazione online
sviluppo istituzioni
classe virtuale lingue



Master
Sviluppo in Python

- » Modalità: online
- » Durata: 12 mesi
- » Titolo: TECH Global University
- » Accreditamento: 60 ECTS
- » Orario: a tua scelta
- » Esami: online

Master

Sviluppo in Python