

Python - Programmazione a oggetti

Le principali funzionalità per la progettazione, lo sviluppo e la manutenzione delle applicazioni

 A distanza



5 giorni (35 Ore)

Open : 2.795,00 € +IVA

WebCode: IT.38

Packaged in azienda : 6.450,00 € +IVA +10% di
Project Management (Quota riferita ad un gruppo
di 10 pax max)

Customized : Su richiesta

Provenendo dal mondo Open Source, Python è un linguaggio rinomato e di **facile accesso**. Offre agli sviluppatori una serie di strumenti e funzionalità per facilitare i loro compiti (pile di moduli, nessuna compilazione richiesta, debugger integrato, test shell, ricca documentazione...). Ma prima di poter beneficiare di tutti questi vantaggi, i nuovi sviluppatori Python dovranno appropriarsi delle **nozioni essenziali** di programmazione a oggetti e apprendere la **sintassi** del linguaggio. Seguendo questa formazione di 5 giorni, i partecipanti copriranno tutti gli argomenti che consentiranno loro di progettare, sviluppare, distribuire e mantenere applicazioni Python.

Questa formazione **prepara** alla **certificazione TOSA Python**, che gli interessati potranno conseguire autonomamente presso enti terzi.

A chi è rivolto



Per chi

- Sviluppatori
- Ingegneri
- Project manager vicini allo sviluppo



Prerequisiti

Avere una conoscenza di base della programmazione
(idealmente in linguaggio a oggetti)

Programma

1 - Sintassi del linguaggio Python

- Identificatori e riferimenti
- Convenzioni di codifica e regole di denominazione
- Blocchi, commenti
- I tipi di dati disponibili
- Variabili, visualizzazione formattata, ambito locale e globale
- Manipolazione di tipi numerici, manipolazione di stringhe di caratteri
- Gestione di array dinamici (lista), array statici (tupla) e dizionari
- L'uso dei file
- La struttura condizionale if/elif/else
- Operatori logici e operatori di confronto
- I cicli di iterazione while e for
- Interruzione delle iterazioni di interruzione/continua
- La funzione gamma
- Funzioni di scrittura e documentazione
- Espressioni lambda
- generatori
- Strutturare il codice in moduli

2 - Approccio orientato agli oggetti

- Principi del paradigma oggettuale
- La definizione di un oggetto (stato, comportamento, identità)
- La nozione di classe, attributi e metodi
- Incapsulamento dei dati
- Comunicazione tra oggetti
- Ereditarietà, trasmissione delle caratteristiche di una classe
- La nozione di polimorfismo
- Associazione tra classi
- Interfacce
- Presentazione di UML
- Diagrammi di classi, sequenze, attività...
- Concetto di modello di progettazione

3 - Programmazione ad oggetti in Python

- Caratteristiche del modello a oggetti Python
- Scrivere classi e istanziarle
- Costruttori e Distruttori
- Protezione dell'accesso di attributi e metodi
- La necessità del parametro Self
- Ereditarietà singola, ereditarietà multipla, polimorfismo
- Le nozioni di visibilità
- Metodi speciali
- Introspezione
- L'implementazione delle interfacce
- Best practice e modelli di progettazione comuni
- Utilizzo del meccanismo di eccezione per la gestione degli errori

4 - Utilizzo di STDLIB

- Argomenti passati sulla riga di comando
- Utilizzo del motore delle espressioni regolari Python con il modulo "re", caratteri speciali, cardinalità
- Manipolazione del file system
- Presentazione di alcuni importanti moduli della libreria standard: modulo "sys", "os", "os.path"
- Pacchetto e installazione di una libreria Python
- Accesso a database relazionali, funzionamento delle API DB

5 - Strumenti di controllo qualità

- Strumenti di analisi del codice statico (Pylint, Pychecker)
- Analisi dei report di analisi (tipi di messaggi, avvisi, errori)
- Estrazione automatica della documentazione
- Il debugger di Python (analisi passo-passo e post-mortem)
- Sviluppo guidato dai test
- Moduli di unit test Python (Unittest...)
- Automazione dei test, aggregazione dei test
- Test di copertura del codice, profilazione

6 - Creazione HMI TKINTER

- I principi della programmazione delle interfacce grafiche
- Presentazione della libreria TkInter
- I principali contenitori
- Presentazione dei widget disponibili (Button, Radiobutton, Entry, Label, Listbox, Canvas, Menu, Scrollbar, Text...)
- Il gestore delle finestre
- Il posizionamento dei componenti, i diversi layout
- La gestione degli eventi, l'oggetto evento
- App multi-finestra

7 - Interfaccia Python/C

- Panoramica del modulo Ctypes
- Caricamento di una libreria C
- Chiamata di una funzione
- Riscrittura di una funzione Python in C con l'API Python/C
- Creazione di moduli C per Python con Pyrex
- L'interprete Python in C
- Utilizzando il code profiler



Obiettivi del corso

- Conoscere la sintassi del linguaggio Python
- Acquisire i principi della programmazione a oggetti
- Implementare le funzionalità del modulo Python e le best practice associate
- Saper progettare interfacce grafiche
- Comprendere l'uso di strumenti per testare e valutare un programma Python



Esercitazioni

- Una formazione molto pratica: i partecipanti saranno portati a svolgere numerosi **workshop** che permetteranno loro di acquisire una prima esperienza pratica dello sviluppo in Python
- Metodi, **best practice** e suggerimenti per ottenere il massimo dalla programmazione in Python e sviluppare così interfacce affidabili ed efficienti



Date 2026



Ultimi posti



Edizione garantita

dal 16 mar al 24 mar

- dal 16 mar al 18 mar
- dal 23 mar al 24 mar

dal 21 set al 29 set

- dal 21 set al 23 set
- dal 28 set al 29 set