

---

*Corso*

*Python Completo Certificato*

---

## CORSO PYTHON BASE

## MODULO 1

Dopo un excursus teorico sulla programmazione, scriviamo il nostro primo programma con la funzione print

- Cosa sono i programmi
- Cos'è un linguaggio
- Linguaggio macchina e naturale
- L' interpretazione
- Gli errori di interpretazione
- Interpretazione e compilazione
- Il linguaggio python
- Scaricare ed installare python
- Ambiente di sviluppo idle
- Il nostro primo programma
- Accedere ai codici sorgente
- Errori di sintassi
- Errori di runtime
- Le funzioni python
- La funzione print
- Gli argomenti di funzione
- Introduzione alle stringhe
- Funzione print su più righe
- Carattere speciale \n
- Funzione print a più argomenti
- Argomenti posizionali e keyword

- Funzione print - argomento end
- Funzione print - argomento sep
- Takeaways: funzione print

## MODULO 2

Impariamo a rappresentare ed acquisire diversi tipi di dati, ad usarli per effettuare dei calcoli e a conservarli in memoria

- I literals
- I tipi di dati python
- Rappresentare gli interi
- Base decimale, ottale, esadecimale
- I numeri float
- La notazione esponenziale
- Le stringhe e i caratteri speciali
- I booleani
- Espressioni logiche
- Takeaways : literals e tipi di dati
- Gli operatori aritmetici
- Le espressioni aritmetiche
- Operatori binari e unari
- Le priorità degli operatori
- Parentesi tonde nelle espressioni
- Takeaways : operatori e espressioni. Le variabili
- Convenzioni di nome
- Uso e tipi delle variabili
- Modificare le variabili

- Espressioni e variabili
- Gli operatori shortcut
- Takeaways : le variabili
- I commenti
- Scrivere codice comprensibile
- Uso corretto dei commenti
- Takeaways : i commenti
- La funzione input
- Salvare gli input nelle variabili
- Argomento funzione input
- Funzione input - tipi di dati
- Funzione di conversione int
- Convertire i valori della input
- Funzione di conversione float
- Operatori per le stringhe
- Operatore di unione +
- Operatore di replicazione \*
- Funzione di conversione str
- Takeaways: input e conversioni

## MODULO 3

Scopriamo come eseguire una porzione di codice solo se è rispettata una condizione o a ripeterla all'infinito!

- Operatori di comparazione
- Operatore eguaglianza ==
- Operatore disequaglianza !=

- Operatori > <
- Operatori non-strict <= >=
- La dichiarazione if
- L' indentazione
- Condizioni della dichiarazione if
- Dichiarazione else
- Dichiarazione elif
- Dichiarazioni elif multiple
- Dichiarazioni if-else annidate
- If-else a riga singola
- Operatori logici and or
- Espressioni logiche
- Operatore logico not
- Priorità operatori logici
- Takeaways : if e espressioni logiche
- Scrivere codice in modo ripetitivo
- Principio dry
- Il ciclo while
- Indentazione nel ciclo while
- Condizioni del ciclo while
- Cicli while infiniti
- Comando break
- Comando continue
- Dichiarazione else
- Ciclo while con valori numerici

- Ciclo while e comparazioni
- Il ciclo for
- La funzione range
- Ciclo while e ciclo for
- Argomenti funzione range
- Comando break e continue nel
- Ciclo for
- Takeaways : cicli python
- Espressioni logiche complesse
- Priorità nelle espressioni
- Complesse
- Tabelle verità and, or e not
- Operatore di comparazione
- Inverso
- Leggi di 'de morgan'
- Operatori logici e stringhe
- Operatori logici e numeri
- Operatori bitwise
- Tabelle verità operatori bitwise
- Operatori shortcut bitwise
- Operatori shifting
- Takeaways : operatori
- Le liste
- Il ciclo for per le liste
- Accedere ai valori delle liste

- Indici e range delle liste
- Operatore in
- Tipi di dati nelle liste
- Modificare le liste
- Metodi insert, append, extend
- Metodi remove, pop, clear
- Keyword del
- Funzioni len, list, sort
- Assegnare valori o riferimenti
- Metodo copy
- Unire due o più liste
- Operatore di join +
- La list comprehension
- List comprehension e
- Dichiarazione if
- Matrici bidimensionali
- Accedere a righe e colonne
- Matrici bidimensionali e ciclo for
- Modificare le matrici 2d
- Popolare dinamicamente le
- Matrici bidimensionali
- Popolare matrici 2d con list
- Comprehension
- Matrici 3d
- Takeaways parte 1 : le liste -

- Dalla definizione fino alla
- Rimozione dei valori
- Takeaways parte 2 : le liste -
- Dall'utilizzo del ciclo for fino
- Alle liste multidimensionali

## MODULO 4

Definiamo le nostre funzioni per eseguire procedure personalizzate e impariamo a gestire tipi di dati complessi

- Le funzioni python
- Le funzioni utente
- I parametri di funzione
- Più parametri per una funzione
- Parametri keyword
- Parametri di numero arbitrario
- Funzioni che ritornano un valore
- Il comando return
- Dichiarazione e corretto
- Utilizzo della funzione
- Lo scope o visibilità delle
- Variabili
- Approfondimenti comando return
- Il valore none
- Liste e funzioni
- Passare valore o riferimento
- Funzioni ricorsive



- Caratteristiche e problematiche
- Takeaways : le funzioni
- Le tuple
- Differenze rispetto alle liste
- Unpacking delle tuple
- I dizionari
- Tipi di dati per i dizionari
- Accedere ai valori dei dizionari
- Comando in per i dizionari
- Il ciclo for e i dizionari
- Sfogliare chiavi, valori e coppie
- Chiave-valore
- Modificare i dizionari
- Aggiungere coppie
- Il metodo update
- Rimuovere coppie
- Il metodo popitem
- Funzione len e sorted
- Takeaways: tuple e dizionari

## CORSO PYTHON INTERMEDIO CERTIFICATO

### MODULO 1

- Cosa sono i moduli
- Come importare un modulo

- Concetto di name space
- Import con Alisa Multipli
- Funzioni DIR
- Il Modulo Math
- Pseudorandomicità
- Il Modulo Random
- Dal File Sorgente all'Hardware
- Il modulo Platform
- I Packages
- Come creare un modulo
- La variabile "name"
- Variabili e funzioni nei moduli
- Definizioni pubbliche e private
- Dove Python cerca i moduli
- La variabile path
- Come creare un package
- Il file init.py
- Importare moduli da packages
- La repository PYPI
- Lo strumento PIP

## MODULO 2

- Codifica dei Caratteri in Python
- Codifica Ascii
- Unicode e UTF-8

- Similitudini fra stringhe e liste -1
- Funzioni ORD e CHR
- Similitudini fra stringhe e liste -2
- Funzioni Min e Max
- Metodi stringa -1
- Funzione List
- Metodi stringa -2 (capitalize, center, endswith, find)
- Metodi stringa -3 (isalnum, isalpha, isdigit, islower, isupper, isspace, join, lower, upper)
- Metodi stringa -4 (lstrip, rstrip, replace, rfind, split, strip, startswith, swapcase, title)
- Stringhe e comparazione
- Funzione sorted e metodo sort
- Le eccezioni
- Come evitare un'eccezione
- Come gestire un'eccezione
- Dichiarazione try except
- Try except – dichiarazione else
- Try con except multiple
- Gerarchia delle eccezioni
- Eccezioni all'interno di funzioni
- La dichiarazione Raise
- La Dichiarazione Assert

## MODULO 3

- Le classi e gli oggetti
- La funzione type
- Proprietà e metodi

- Programmazione procedurale e ad oggetti
- Il metodo costruttore init
- Proprietà pubbliche e private
- Metodi e parametri
- Metodi pubblici e privati
- Superclasse e sottoclasse
- Override di metodo
- Accedere ai metodi della superclasse
- Proprietà di istanza
- Proprietà speciale dict
- Accedere a proprietà private
- Proprietà di classe
- La funzione Hasattr
- Approfondimento sui metodi
- Proprietà speciali (name, module, bases)
- Ereditarietà multipla
- Metodo speciale str
- Ereditarietà a più livelli
- Funzioni issubclass e isinstance
- Keyword is
- Accesso a proprietà fra superclassi e sottoclassi
- Ordini di accesso su classi ereditate
- Polimorfismo
- Metodo Astratto
- Composizione di oggetti

- Comparazione fra ereditarietà e composizione
- Le classi di eccezioni
- La dichiarazione finally
- L'oggetto eccezione nella dichiarazione except
- Creare delle eccezioni

## MODULO 4

- Cosa sono i generatori
- Creare un generatore mediante una classe
- La dichiarazione Yield
- Creare un generatore mediante una funzione
- La list comprehension
- Espressione condizionale
- Similitudine fra list comprehension e generatori
- Creare un generatore mediante list comprehension
- Le funzioni anonime lambda
- La funzione map
- Le funzioni filter
- La closure
- I file
- La Funzione Open
- Metodi di lettura dei file
- Metodi di scrittura dei file
- Mode della funzione open
- I file binari
- La funzione bytearray

- Lettura e scrittura su file binari
- Gli stream stdin, stdout, stderr
- Eccezioni nell'accesso ai file
- Errno e strerror
- Il modulo OS e i suoi metodi
- Il modulo datetime
- La classe date di datetime
- La classe time di datetime
- Il modulo time
- La classe datetime
- Il Metodo strftime
- Operazioni fra date
- La classe timedelta
- Il metodo calendar
- La funzione calendar
- La funzione month
- La classe calendardizionali

## CORSO DI PYTHON AVANZATO

### MODULO 1 - OBJECT ORIENTED PROGRAMMING

- Basi di OOP: classi, oggetti, attributi, metodi, instance vs class variables, superclasses vs subclasses. Costruttori specifici e di default.
- Magic methods: eq, abs, str, getattr, getitem
- iteratori e generatori.

- Metodi statici e di classe, decoratori @classmethod e @staticmethod.
- Classi astratte e metodi astratti, override di metodi astratti.
- Attribute encapsulation: Metodi getter, setter e deleter.
- Ereditarietà, ereditarietà multipla, polimorfismo.
- Subclassing di classi built-in.
- Metaclassi: la metaclassa type e la funzione type(), gli attributi speciali \_\_name\_\_, \_\_class\_\_, \_\_bases\_\_ e \_\_dict\_\_.

## MODULO 2 - ADVANCED PROGRAMMING CONCEPTS

- Funzioni: packing e unpacking di argomenti posizionali e keyword (\*args e \*\*kwargs), first class objects, decoratori di funzioni e di classi, decoratori con argomenti.
- Gestione degli errori e delle eccezioni: eccezioni come oggetti.
- Shallow e deep copy: la funzione id, metodi copy e deepcopy.
- Serialization of Python objects: modulo pickle e oggetti serializzati in singolo byte stream, funzioni dumps() e loads()

## MODULO 3 - CODING CONVENTIONS

- Filosofia del Python: indentazione, continuation lines, maximum line length, line breaks, blank lines, l'istruzione import this, importare moduli, whitespaces, commenti, docstrings, naming conventions, type hints.

## MODULO 4 - GUI PROGRAMMING

- Il modulo tkinter e gli elementi grafici: finestra, titolo, bottoni, icone, label.

- Event-driven programming, Tk(), mainloop() e title
- aggiungere widgets alla finestra: bottoni, label, frames, location e coordinate, size, il metodo place()
- Canvas e suoi metodi, widget Entry, Radiobutton e Button, grid e place managers, binding events con bind().
- Colorare widget e color modes (RGB e HEX).
- Clickable e non clickable widgets

## MODULO 5 - NETWORK PROGRAMMING

- Protocollo REST, network sockets, dominio, indirizzo, porta, protocollo e servizio.
- Comunicazione connection-oriented e connectionless, client e server.
- Il modulo socket per importare e creare socket, connettere socket a server http e chiudere la connessione, inviare richieste ai server con il metodo send(), ricevere risposte con il metodo recv(), gestione delle eccezioni e tipi di eccezione.
- JSON: sintassi, tipi di dati (numerici, strings, booleani, null), dati strutturati (array e oggetti). Serializzare dati Python e deserializzare JSON (i metodi dumps() e load() ).
- XML: sintassi, struttura, processare file xml.
- Il modulo request: metodi http (GET, POST, PUT, DELETE), CRUD, aggiungere e aggiornare dati, data fetching e rimuovere dati da server, response status codes.
- Database programming in Python: il modulo sqlite, creare e chiudere connessioni con i metodi connect e close, creare tabelle, inserire, leggere, aggiornare e eliminare dati, metodi del cursore (execute, executemany, fetchone, fetchall), creare statement SQL di base (SELECT, INSERT, INTO, UPDATE, DELETE). Parsare documenti XML, cercare dati in documenti XML usando i metodi find e findall, costruire documenti XML usando la classe Element e la funzione SubElement.
- Leggere e scrivere da/su CSV con reader, writer, DictReader, DictWriter.



- Logging nelle applicazioni e differenti livelli di logging, usare gli attributi di LogRecord per creare formati di log.
- Creare e parsare configuration files usando l'oggetto ConfigParser.
- File .ini e interpolare valori in file .ini
- Object Oriented Programming
- Advanced Programming Concepts
- Coding Conventions
- Gui Programming
- Network Programming

## CORSO LIBRERIE PYTHON CORSO PRATICO SU RISORSE BUILT-IN ED ESTERNE IN AMBITO PYTHON

### OS:

- il modulo os per lavorare con file e cartelle
- Gestione e creazione di file e cartelle;

### ESPRESSIONI REGOLARI:

- il modulo re per lavorare e gestire le espressioni regolari;

### GENERAZIONE DI NUMERI CASUALI:

- il modulo random per la generazione di numeri casuali;

### TESTING IN PYTHON:

- test del codice con il modulo unittest;

## DATABASE PROGRAMMING IN PYTHON:

- il modulo sqlite, creare e chiudere connessioni con i metodi connect e close,
- creare tabelle, inserire, leggere, aggiornare e eliminare dati,
- metodi del cursore (execute, executemany, fetchone, fetchall),
- creare statement SQL di base (SELECT, INSERT, INTO, UPDATE, DELETE);

## JSON, CSV E XML:

- i moduli json, csv e xml per lavorare e gestire con file json, csv e xml;
- principali differenze tra questi formati e usi comuni di tali moduli;

## NUMPY:

- il modulo NumPy per gestire e lavorare con array numerici;
- Operazioni tra scalari ed array e broadcasting;
- Principali usi del modulo NumPy;

## PANDAS:

- il modulo Pandas per il Data Analysis;
- Series e DataFrame per gestire dati tabellari;
- Leggere e scrivere su csv e xlsx;
- Gestire dati corrotti e filtrare dati;
- Principali usi del modulo Pandas;

## MATPLOTLIB:

- il modulo Matplotlib per la visualizzazione dei dati;
- Interfacce implicite ed interfacce esplicite;
- Plot ed elementi di una figura;
- Plot interpolati, scatter plot, istogrammi e grafici a torta;
- Principali usi del modulo Matplotlib;

## DATETIME:

- il modulo datetime per gestire e lavorare con date e intervalli di date ed ore;

- Convertire stringhe in date e formattare date in stringhe;

#### LOGGING:

- il modulo logging per realizzare un logger per la reportistica;
- Creazione di logger personalizzati e reindirizzamento di flussi di dati su file log;

#### GUI PROGRAMMING:

- Il modulo tkinter e gli elementi grafici: finestra, titolo, bottoni, icone, label, event-driven programming, Tk(),
- mainloop() e title,
- aggiungere widgets alla finestra (bottoni, label, frames, location e coordinate, size, il metodo place() ),
- Canvas e suoi metodi,
- widget Entry,
- Radiobutton e Button, grid e place managers, binding events con bind();
- Colorare widget e color modes (RGB e HEX);
- Clickable e non clickable widgets