

# Python Dev Course

Email di Giacomo Usai: [giacomo.usai.iic97@gmail.com](mailto:giacomo.usai.iic97@gmail.com)

Tutorial su come installare Python:

<https://www.youtube.com/watch?v=Kn1HF3oD19c>

Sito ufficiale Python dove è possibile scaricare l'ultima versione sul proprio pc:

<https://www.python.org/downloads/>

Sito ufficiale JetBrains dove scaricare la versione Community di PyCharm per poter sviluppare il proprio codice:

<https://www.jetbrains.com/pycharm/download/#section=windows>

**Nota.** PyCharm per funzionare deve avere una versione di Python installata sul PC.

Risorse utili dove esercitarsi: i seguenti siti in inglese sono molto validi e contengono diverse categorie a seconda degli argomenti che caratterizzano il linguaggio. In ognuno sono presenti esercizi con le soluzioni.

<https://www.w3schools.com/python/>

<https://www.geeksforgeeks.org/python-programming-language/>

**Consiglio di scaricare questo libro molto utile, anche un pò divertente, per tutti gli studenti appassionati di informatica.**

<https://drive.google.com/file/d/1tMllvuRregMCwwYb-c25DIhJ7FxsPaM8/view>

Vengono proposti quesiti di varia natura e analizzate le interessanti soluzioni che si possono raggiungere grazie all'informatica.

Per qualsiasi approfondimento rimango a disposizione!

## Lezione 1

---

1. `print("Hello, World!")`
2. indentations, cioè “*indentazione*” del codice usando il tasto TAB della tastiera
3. Comment: su una riga (#) o multiriga (""")
4. Variabili
  - a. string, int
  - b. caso “-”
  - c. multi assegnazione su una riga
5. keyword **type()**
  - a. `x = 5`
  - b. `x = "Hello World"`
  - c. `x = 20.5`

d. `x = ["apple", "banana", "cherry"]`

e. `x = True`

f. `print(type(x))`

6. Definire una funzione:

a. `def my_function(x):`

Nella Prima lezione abbiamo introdotto il linguaggio Python. Come installarlo sul proprio PC e come scaricare l'ambiente di sviluppo per poter scrivere del codice (PyCharm).

Abbiamo visto la funzione ***print*** che consente di stampare a video il contenuto "di qualcosa": una stringa, un numero o sostanzialmente qualsiasi informazione siamo interessati, basta appunto specificare cosa vogliamo che ***print*** stampi passandogli questa informazione all'interno delle parentesi tonde.

Questo meccanismo di mettere all'interno delle parentesi tonde il nome di una variabile è molto utile e verrà usato parecchio.

Infine abbiamo brevemente visto come definire una ***funzione***: cioè una procedura in grado di prendere informazioni in input, elaborarle, e fornire un risultato in output. Questo risultato dipende da come e cosa vogliamo che la funzione restituisca. Un esempio è la funzione concatenazione di due stringhe: quindi una funzione che prende in input 2 stringhe e restituisce la loro unione (es. input="Ci", "ao"; avremo come output="Ciao", cioè l'unione della prima stringa con la seconda).

In Python scriveremo:

```
def concatenazionestringhe(x, y):  
    print(x + y)
```

Come possiamo notare, l'operatore "+" ci consente di fare la concatenazione (cioè unire) due stringhe.

Nel caso in cui invece di due stringhe vengono forniti due numeri interi (es. 2 e 3), verrà eseguita la loro somma. Quindi lo stesso

operatore "+" si comporta in modo diverso a seconda dell'input che Noi diamo: nel caso di due numeri farà la somma, nel caso di due stringhe farà la concatenazione/unione delle due stringhe.

## Compito 1

- Collegarsi al sito [https://www.w3schools.com/python/exercise.asp?filename=exercise\\_syntax1](https://www.w3schools.com/python/exercise.asp?filename=exercise_syntax1) e svolgere esercizi su *Syntax, Comments, Variables, Data Types* e *Numbers*.
- scrivere una funzione che prende come argomento 3 numeri e stampa la loro somma a video.
- scrivere una funzione che prende come argomento 3 stringhe e stampa le due stringhe concatenate tra loro.

## Lezione 2

7. ripasso lezione precedente e compiti

8. Definire una **funzione**:

- a. a cosa serve? quando è conveniente crearne una?
- b. funzioni primitive di Python

9. Python *Keywords*: cioè le parole speciali. Abbiamo visto **def** (per creare una funzione), **IF**, **True**... e ne vedremo molte altre. (Qui trovi l'elenco completo [https://www.w3schools.com/python/python\\_ref\\_keywords.asp](https://www.w3schools.com/python/python_ref_keywords.asp) )

10. *Namespaces* and *Scope* in Python

- a. posso chiamare due variabili con lo stesso nome?
- b. posso chiamare una variabile con lo stesso nome di una keyword? Cioè creare una variabile che si chiama ad esempio *True*?

11. cos'è il **MAIN()**? Perché è importante?

12. controllo: **IF**

- a. [https://www.w3schools.com/python/python\\_conditions.asp](https://www.w3schools.com/python/python_conditions.asp)

13. cicli: **FOR**

- a. [https://www.w3schools.com/python/python\\_for\\_loops.asp](https://www.w3schools.com/python/python_for_loops.asp)

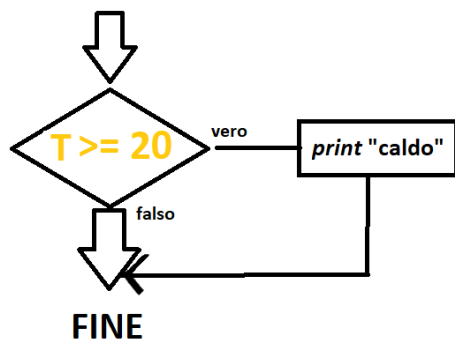
14. **input** da console utente

Riassunto Seconda lezione:

il **main()** è una parte presente in tutti i programmi: le parentesi dopo il main cosa indicano? Parentesi tonde, indicano una *funzione*.

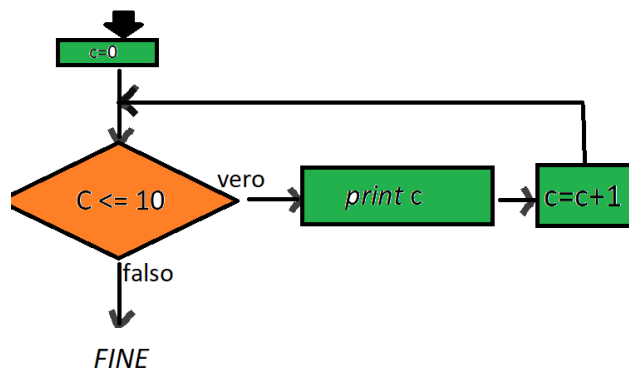
Un programma può contenere una o più funzioni, una delle quali deve essere **main**. Perché è obbligatorio avere una funzione **main**? Perché ogni programma comincia eseguendo la funzione **main**, è pertanto il nostro punto di partenza, un riferimento universale.

Principio di funzionamento del controllo **IF**: l'immagine seguente mostra sia il flow chart (diagramma di flusso) sia relativo codice python.



```
if (T >= 20) :
    print("caldo")
```

Principio di funzionamento del comando **FOR**: l'immagine mostra sia il flow chart (diagramma di flusso) sia relativo codice python.



```
for c in range(10):
    print(c)
```

Eseguendo questo codice python, si ottiene una stampa in sequenza dei numeri da 0 a 9, come mostrato in figura:

```

Run: main2 x
C:\Users\Giacco\PycharmProjects\PyDevMensa\venv\Scripts\python.exe C:/Users/Giacco/PycharmProjects/
0
1
2
3
4
5
6
7
8
9
Process finished with exit code 0
  
```

## Compito 2

- Abbiamo visto il ciclo FOR per stampare una sequenza di numeri da 0 a 9.
  - Sapresti modificarlo in modo che stampi partendo da 5?
  - Sapresti modificarlo in modo che stampi da 1 a 100?
- In matematica, il fattoriale di un numero è il prodotto di tutti i suoi predecessori fino ad 1: quindi ad esempio il fattoriale di 5 si indica con 5! e sarà  $5! = 5*4*3*2*1 = 120$ .

Completare il seguente codice python, sostituendo i “???”, in modo che esegua il fattoriale di 5:

```
def funzione_fattoriale():  
    fattoriale = 1  
    for c in range(2, ???):  
        print(c)  
        fattoriale = c * ???  
    print(fattoriale)
```

# Lezione 3

---

15. ripasso lezione precedente e compiti
16. implementazione della funzione fattoriale di un numero preso da tastiera
  - a. usando la funzione `input()`
17. *Serie di Fibonacci* e ciclo **while**:
18. **Sequenze** (<https://www.geeksforgeeks.org/python-lists/?ref=lbp>)
  - a. esempio: `A = [23, 302, 9]`
  - b. cosa sono?
  - c. come si accede ad un elemento?
19. Funzione **len()** per conoscere la lunghezza della sequenza
20. Funzione **append()** per inserire in coda alla sequenza
21. Funzione **remove()** per rimuovere un elemento (uno o più) dalla sequenza

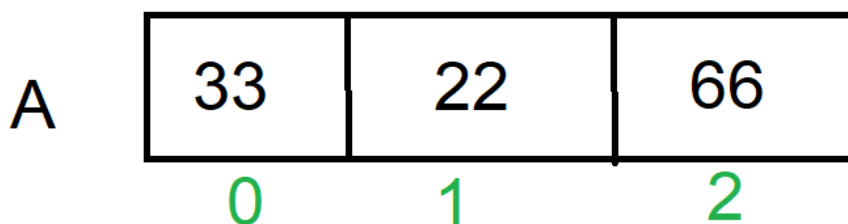
Riassunto Terza lezione:

Si noti che l'indice iniziale è zero (0) e poi così via 1,2!!

Quindi se definisco una sequenza di 3 numeri:

```
A = [33, 22, 66]
```

La seguente immagine aiuta a comprendere la logica con cui si accede ad una Sequenza in Python:



`A[0] = 33`

`A[1] = 22`

`A[2] = 66`

La seguente tabella riassume le principali operazioni che possiamo effettuare su una sequenza (in questo caso chiamata "s").

Operation	Result	Notes
<code>s[i] = x</code>	item <i>i</i> of <i>s</i> is replaced by <i>x</i>	
<code>s[i:j] = t</code>	slice of <i>s</i> from <i>i</i> to <i>j</i> is replaced by the contents of the iterable <i>t</i>	
<code>del s[i:j]</code>	same as <code>s[i:j] = []</code>	
<code>s[i:j:k] = t</code>	the elements of <code>s[i:j:k]</code> are replaced by those of <i>t</i>	(1)
<code>del s[i:j:k]</code>	removes the elements of <code>s[i:j:k]</code> from the list	
<code>s.append(x)</code>	appends <i>x</i> to the end of the sequence (same as <code>s[len(s):len(s)] = [x]</code> )	
<code>s.clear()</code>	removes all items from <i>s</i> (same as <code>del s[:]</code> )	(5)
<code>s.copy()</code>	creates a shallow copy of <i>s</i> (same as <code>s[:]</code> )	(5)
<code>s.extend(t)</code> or <code>s += t</code>	extends <i>s</i> with the contents of <i>t</i> (for the most part the same as <code>s[len(s):len(s)] = t</code> )	
<code>s *= n</code>	updates <i>s</i> with its contents repeated <i>n</i> times	(6)
<code>s.insert(i, x)</code>	inserts <i>x</i> into <i>s</i> at the index given by <i>i</i> (same as <code>s[i:i] = [x]</code> )	
<code>s.pop()</code> or <code>s.pop(i)</code>	retrieves the item at <i>i</i> and also removes it from <i>s</i>	(2)

Inoltre abbiamo visto come calcolare la Serie di Fibonacci e il ciclo **while**. Il seguente codice è quello implementato nell'esempio pratico e come possiamo vedere riceve un numero *n* che determina quanto andrà avanti in ciclo while: cioè fintanto che la nostra variabile *a* è minore di *n*. Fintanto che la condizione è vera il programma eseguirà le operazioni all'interno del ciclo while, infine quando la condizione non sarà più vera, quindi avremo *a* ≥ *n*, esco dal ciclo **while** e stampo il risultato.



```
def fib(n):
    a, b = 0, 1
    while a < n:
        print(a, end=' ')
        a, b = b, a+b
    print()
fib(1000)
```

## Compito 3

- Definire una funzione che somma tutti gli elementi della seguente sequenza:
    - A = [10, 5, 2, 3, 50, 20, 10]
  - Esercitarsi sul sito:
    - [https://www.w3schools.com/python/exercise.asp?filename=exercise\\_lists1](https://www.w3schools.com/python/exercise.asp?filename=exercise_lists1)
    - sono presenti anche le soluzioni!
  - Difficile:** Definire una funzione che riceve in input 3 numeri inseriti da tastiera dall'utente e li salva all'interno di una sequenza.
    - aiuto 1: per ricevere il numero da tastiera usare il codice visto a lezione  
`a = int(input('inserisci numero: '))`
    - aiuto 2: per salvare il numero (a) all'interno della sequenza bisogna crearla prima di richiedere l'input, e usare la funzione "append()" che inserisce in fondo alla sequenza.
    - aiuto 3: è utile usare un ciclo **for** per ripetere 3 volte la stessa procedura: cioè chiedo il numero in input all'utente e poi lo salvo nella sequenza, ripeto per 3 volte.
- Prendi spunto dal seguente codice, è un ciclo **for** che stampa per 3 volte la lettera "s":

```
for i in range(3):
    print("s")
```

## Lezione 4

23. funzione **append**

24. funzione **pop**

25. funzione **remove**

26. format

- a. per formattare una stringa in output (console): di fatto sostituisce le graffe con la variabili che passiamo alla funzione format:

```
a=3223  
print("{0} = {1}".format("ciao", a))
```

risultato: stampa "ciao = 3223"

Riassunto Quarta lezione:

Con il seguente codice che definisce una sequenza, controllo se il numero 3 è all'interno...usando la keyword **in**.

```
myList = [1,2,3,4,5]  
if 3 in myList:  
    print("3 is present")
```

Dopo aver visto come definire una sequenza, in questo caso chiamata A (lettera maiuscola!), abbiamo introdotto le funzioni:

- **pop**
  - Remove and return item at index (default last)
  - Rimuove e ritorna l'elemento all'indice specificato (di default rimuove dalla coda/fondo della lista)
- **append**
  - Append object to the end of the list.
  - Aggiunge elemento in coda alla lista.
- **remove**
  - Remove first occurrence of value.  
Raises ValueError if the value is not present.  
Rimuove prima occorrenza del valore specificato. Genera errore se il valore non è presente all'interno della

sequenza (e quindi non trova nessun elemento specificato da rimuovere).

```
A = [10, 5, 2, 3, 50, 20, 10]
```

```
A.pop()
```

```
A.append(30)
```

```
print(A)
```

## Compito 4

- Buon **Natale!!!**

# Function to draw a Christmas tree with a given height

```
def draw_tree(height):
```

```
    # Loop through each row of the tree
```

```
    for i in range(1, height + 1):
```

```
        # Print the spaces before the asterisks on each row
```

```
        for j in range(height - i):
```

```
            print(" ", end="")
```

```
        # Print the asterisks on each row
```

```
        for j in range(2 * i - 1):
```

```
            print("*", end="")
```

```
        # Move to the next line
```

```
        print()
```

# Call the function to draw a tree with a height of 5

```
draw_tree(5)
```