



Python programming



programmazione
in Python per il Liceo
delle Scienze Applicate

Giulio Angiani
I.I.S. "Blaise Pascal" - Reggio Emilia

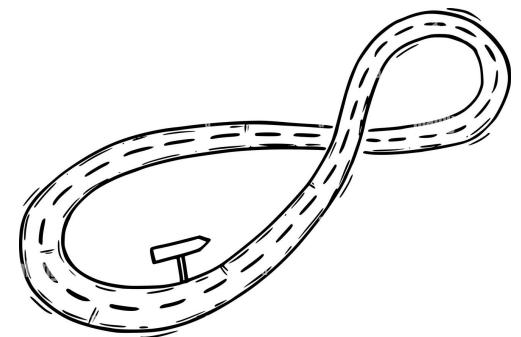


Programmazione in Python

Programmazione in Python

RECAP: Nella scorsa lezione abbiamo visto

- Cos'è un ciclo
- Cicli infiniti
- Uscita dai cicli infiniti
- Esercizi sui cicli infiniti con **break** di uscita



Python - Ripetizione di operazioni

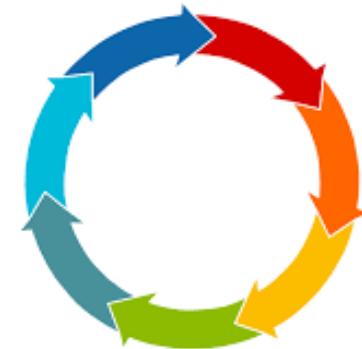
CICLI E RIPETIZIONI

Parliamo di **Cicli** quando dobbiamo scrivere un programma che esegue **tante volte** la stessa operazione oppure **operazioni simili**

Tre tipologie di **CICLI** con alcune sottotipologie

Cicli **infiniti**

- eseguiremo **per sempre** le operazioni simili
- uscita dal ciclo : MAI



Cicli a **numerosità predefinita**

- Sappiamo da subito che eseguiremo le operazioni simili **N volte**
- uscita dal ciclo : dopo il numero N scelto

Cicli a **numerosità non predefinita**

- **NON** sappiamo da subito **quante volte** eseguiremo le operazioni simili
- uscita dal ciclo : quando si verifica una certa condizione all'interno del ciclo

Python - Ripetizione di operazioni

Secondo tipo: Cicli a **numerosità predefinita**

- Il numero di **iterazioni** è già nota prima di iniziare
- Non c'è bisogno di **break** di uscita o di altre condizioni

In **Scratch** usavamo il blocco indicato qui a destra... 

In **Python** diventa:

```
for i in range(10):  
    # qui le istruzioni che voglio eseguire per 10 volte  
    print("Ciao ragazzi e ragazze!!!!")
```



PYTHON

```
Ciao ragazzi e ragazze!!!!  
Ciao ragazzi e ragazze!!!!
```

OUTPUT



Python - Ripetizione di operazioni

Secondo tipo: Cicli a **numerosità predefinita**

Spiegazione

L'istruzione **for nomevariabile in range(-numerosità)** di python funziona così:

- definisce al volo una variabile che ha valore 0
- dopo ogni iterazione la incrementa di 1
- quando il valore di questa variabile vale come numerosità esce dal ciclo

Soltamente si usa la variabile **i** come **contatore** del ciclo e la funzione **range** per indicare la numerosità della ripetizione.

Possiamo però usare **qualsiasi** variabile (che non sia in uso per qualche altro motivo)

Esempi:

- l'istruzione **for i in range(5)**: crea un ciclo da 5 iterazioni con il contatore **i**
- l'istruzione **for j in range(8)**: crea un ciclo da 8 iterazioni con il contatore **j**
- l'istruzione **for conta in range(10)**: crea un ciclo da 10 iterazioni con il contatore **conta**



Python - Ripetizione di operazioni (2)

Secondo tipo: Cicli a **numerosità predefinita**

Altri esempi ed uso avanzato di **range**

la funzione **range** prende sempre **almeno** un valore intero (che indica la numerosità N della ripetizione) perché di **default** comincia a contare da 0 e arriva fino ad N-1

Ad esempio:

- l'istruzione **for i in range(5):** crea un ciclo da 5 iterazioni con il contatore **i**
 - comincia a contare da 0 fino a 4 (0, 1, 2, 3, 4 \Rightarrow 5 iterazioni)

```
for i in range(5):
    print("il contatore vale", i)
```

PYTHON

0
1
2
3
4

OUTPUT



Python - Ripetizione di operazioni (3)

Secondo tipo: Cicli a **numerosità predefinita**

Altri esempi ed uso avanzato di **range**

range può avere altri parametri per definire **da dove** iniziare a contare e fino a **dove arrivare** (sempre -1)

- di default il passo di incremento è **1** \Rightarrow il contatore aumenta di 1 dopo ogni iterazione

Ad esempio:

- l'istruzione **for i in range(6, 10):** crea un ciclo da 4 iterazioni con il contatore **i**
 - comincia a contare da 6 fino a 9 (6, 7, 8, 9 \Rightarrow 4 iterazioni)

```
for i in range(6, 10):  
    print("il contatore vale", i)
```

PYTHON

6
7
8
9

OUTPUT



Python - Ripetizione di operazioni (4)

Secondo tipo: Cicli a **numerosità predefinita**

Altri esempi ed uso avanzato di **range**

range può avere altri parametri per definire **da dove** iniziare a contare, fino a **dove arrivare** (sempre -1) ed il passo di **incremento**

- se aggiungo il terzo parametro a range definisco anche il passo di incremento

Ad esempio:

- l'istruzione **for cont in range(10, 30, 5):** crea un ciclo da 4 iterazioni con il contatore **cont**
 - comincia a contare da 10 fino a 29 con passo 5 (10, 15, 20, 25 \Rightarrow 4 iterazioni)

```
for cont in range(10, 30, 5):
    print("il contatore vale", cont)
```

PYTHON

10
15
20
25

OUTPUT





Ripetizioni in Python

Esempi ed esercizi



Python - Ripetizione di operazioni

Esempi ed esercizi

Problema:

- Stampare tutti i numeri interi positivi minori di 10 (0 lo consideriamo positivo)
- Algoritmo: Devo partire da 0 e arrivare fino a 9 compreso
 - è esattamente quello che fa `range...`
 - partire da 0 e arrivare ad un numero in meno del termine....

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

0
1
2
3
4
5
6
7
8
9



PYTHON

OUTPUT

Python - Ripetizione di operazioni

Esempi ed esercizi

Problema:

- Stampare tutti i numeri pari compresi tra 100 e 110 compreso
- Algoritmo:
 - i numeri pari sono uno ogni 2...
 - parto da 100 e arrivare fino a 110 compreso (quindi range(111))

```
# metodo 1
# controllo tutti i numeri e vedo se sono divisibili per 2
for N in range(100, 111):
    if N%2 == 0:
        print(N)
```

PYTHON

```
100
102
104
106
108
110
```

OUTPUT



Python - Ripetizione di operazioni

Esempi ed esercizi

Problema:

- Stampare tutti i numeri pari compresi tra 100 e 110 compreso
- Algoritmo:
 - i numeri pari sono uno ogni 2...
 - parto da 100 e arrivare fino a 110 compreso (quindi range(111))

```
# metodo 2 : più bello
# controllo tutti i numeri e vedo se sono divisibili per 2
for N in range(100, 111, 2):
    # salto di 2 in 2....
    print(N)
```

PYTHON

```
100
102
104
106
108
110
```

OUTPUT





Giulio Angiani
I.I.S. "Blaise Pascal" - Reggio Emilia