

# PENSA UN NUMERO...

Imparare a programmare è un po' come imparare il gioco del calcio.

Teoricamente, si potrebbe studiare ogni singola mossa sulla carta e scendere in campo soltanto quando si è in grado di padroneggiarle tutto, ma ciò, oltre ad essere scarsamente divertente, sarebbe anche molto faticoso e lento.

Lo stesso si può dire della programmazione: si possono leggere e rileggere un'infinità di testi, ma il metodo migliore è proprio quello di 'entrare in campo'.

Da dove iniziare?

## INDOVINARE UN NUMERO

Il più semplice gioco per computer è quello nel quale la macchina 'pensa' un numero a caso e il giocatore deve indovinarlo.

### LA FUNZIONE `random.randint()`

Tutti i computer sono dotati di un generatore casuale di numeri, utile in questo genere di giochi.

In PYTHON vi si accede tramite la funzione '`randint()`' della libreria '`random`'. I numeri prodotti sono compresi in un range di valori specificati nei parametri.

Verifichiamolo con un semplice programma che chiameremo indovina (il `.py` finale è una 'estensione' del file necessario per identificare che questo è un programma in python)

```
$ edit indovina.py
```

```
import random

x = random.randint(1,10)
print(x)
```

Una volta avviato il nostro programma fornirà ogni volta un numero diverso. Tutti i numeri prodotti saranno compresi tra 1 e 10 perchè questo gli abbiamo specificato nelle parentesi; quei due numeri si chiamano "parametri" della funzione '`randint`' del modulo '`random`'

```
$ python3 indovina.py
6
$ python3 indovina.py
2
$ python3 indovina.py
7
```

'python3' è il comando necessario per avviare il programma.

## USO DELLE VARIABILI

Nel breve programma appena presentato, non solo abbiamo generato un numero, ma gli abbiamo anche assegnato un nome: 'x'; Da questo momento in poi, se adoperiamo di nuovo 'x', il computer sa di dover fornire il numero 'scelto a caso'.

Questa 'x', che serve al computer per identificare un particolare numero e usarlo in operazioni aritmetiche oppure in un confronto, si chiama 'variabile'.

### La funzione 'input'

Avendo generato il numero casuale, il passo successivo è quello di far sì che il computer accetti un nostro tentativo di indovinarlo. A questo scopo ci serviamo della funzione 'input'.

La sola parola 'input' non basta: occorre specificare anche un nome di variabile, nella quale il computer depositerà il valore immesso.

Liberi di scegliere una lettera, o anche una combinazione di lettere qualsiasi (purché non 'x', già impegnata), usiamo 'g'.

Dal momento che ci aspettiamo che il giocatore inserisca un numero, allora dovremo istruire il computer per dirgli che deve aspettarsi un numero; questo lo si fa con la funzione 'int'.

E' anche utile mettere un messaggio di cortesia che dica al giocatore di inserire un numero.

Il comando completo è:

```
g = int(input("Inserire un numero "))
```

### I confronti

Adesso però è necessario che il computer confronti il valore da lui generato con quello da noi immesso. Per far ciò impieghiamo un comando particolare di Python:

```
if x == g:
    print("Risposta esatta")
```

Il comando 'if' (se) è molto utile e frequentemente impiegato nei programmi.

Notare il doppio '=='; il singolo '=' che abbiamo utilizzato prima serve ad 'assegnare' un valore ad una variabile.

Il doppio '==' serve per effettuare un confronto tra variabili.

Oltre ad if, il Python mette a disposizione un comando aggiuntivo 'else' (altrimenti) che fornisce un'alternativa qualora la condizione NON sia soddisfatta:

```
if x == g:
    print("Risposta esatta")
else:
    print("Risposta sbagliata")
```

## IMPOSTAZIONE DEL PROGRAMMA

Adesso immettiamo il programma per intero.

Nel farlo approfittiamo per cambiare i nomi alle variabili mettendo nomi che rendano più l'idea di cosa andranno a contenere, quindi 'x' che contiene il numero da indovinare diventa 'caso' e 'g' che indica il numero inserito dal giocatore diventa 'num'

```
$ edit indovina.py
import random

caso = random.randint(1,10)

num = int(input("Inserire un numero "))
if num == caso:
    print("Risposta esatta")
else:
    print("Risposta sbagliata")
```

Avviando il programma con

```
$ python indovina.py
```

il gioco funziona ma si esaurisce in una sola 'mandata'.

## I CICLI

Per fare in modo che il giocatore possa riprovare ad indovinare è necessario aggiungere un 'ciclo'.

Questo si fa con il comando 'while' ('mentre') che fa in modo che tutti i comandi che vengono dopo siano ripetuti tante volte fino a che si verifica una certa condizione. Per ora come condizione inseriamo 'True' (che significa 'vero'), cioè 'ripeti per sempre'.

Questo lo mettiamo nella riga tra "caso = ..." e "num = ...".

```
caso = random.randint(1,10)
while True:
    num = int(input("Inserire un numero "))
```

Importante: dopo le condizioni, quali while e if, che terminano con il simbolo ':', python vuole che i comandi successivi siano "indentati" cioè spostati leggermente verso destra mettendoci un po' di spazi davanti in modo tale da

identificare un 'blocco' di comandi che dovrà eseguire all'interno di quella condizione.

## La fine del ciclo

In questo modo il giocatore che non ha indovinato può riprovare ad indovinare, tuttavia anche se ha indovinato gli viene chiesto di riprovare. Quindi dobbiamo modificare il nostro programma per assicurarci di entrare nel ciclo esclusivamente se non ha ancora indovinato.

Allora quella condizione 'True' che prima significava 'sempre' ora dovrebbe essere 'solo se ha indovinato'.

Per fare questo possiamo aggiungere una nuova variabile che chiameremo 'indovinato' che all'inizio del gioco sarà falso ('False') perchè non ha ancora giocato, ma se indovina dovrà cambiare e diventare 'True' dentro la prima condizione dell'if, e poi confrontata nella riga del while.

Il codice completo finale diventa quindi

```
import random

caso = random.randint(1,10)
print ("Ciao. Il computer ha scelto un numero tra 1 e 10. Riesci ad indovinarlo?")

indovinato = False

while indovinato == False:
    num = int(input("Inserire un numero "))
    if num == caso:
        print("Risposta esatta")
        indovinato = True
    else:
        print("Risposta sbagliata, riprova")

print ("Arrivederci")
```

# IMPARIAMO LE MOLTIPLICAZIONI

La funzione `randint()` si rivela utile in molteplici occasioni.

Supponiamo di voler insegnare ad un bambino la tabellina del 9, potremmo scrivere

```
num = int(input("Quanto fa 1 per 9?"))
if num == 9:
    print("Corretto")
num = int(input("Quanto fa 2 per 9?"))
if num == 18:
    print("Corretto")
...
```

ma sarebbe decisamente chilometrico e non prevederebbe neanche il caso di una risposta errata!

Un metodo migliore, applicabile molto più in generale, è basato sull'impiego della funzione `randint()` con la quale, oltretutto, è possibile evitare una precisa sequenza nelle domande, ottenendo uno strumento didattico di maggior efficacia.

Nella stesura di un programma è sempre bene partire da un nucleo centrale, lasciando gli abbellimenti ad un secondo tempo.

Proviamo il seguente programma.

```
$ edit tabelline.py
```

```
import random

mol = random.randint(0,12)
# mol sta per moltiplicando
print(f"Quanto fa {mol} per 9?")
# f"" è una sintassi che consente di stampare a schermo
# anche il contenuto di una variabile se questa si inserisce tra {}
num = int(input())
if num == mol*9:
    print("Corretto")
```

```
$ python3 tabelline.py
```

Le righe che cominciano per '#' è possibile inserirle in un codice, ed è buona norma, per inserire commenti che aiuteranno più avanti a ricordarsi a cosa servivano i pezzi di codice scritti.

La funzione `randint()` viene utilizzata in modo simile al gioco precedente; solo che scegliamo in un range più ampio.

Il numero generato viene assegnato alla variabile 'mol'.

Poi si chiede al giocatore quanto dovrebbe essere il risultato della moltiplicazione e con `input` lo si assegna a 'num'. Questa viene poi confrontata

con il prodotto reale della moltiplicazione 'mol' per 9 (nei computer il 'per' si indica con un asterisco '\*', il 'diviso' con una barra '/')

Per fare in modo che il gioco continui all'infinito con ogni volta un numero diverso possiamo inserire il tutto (dalla generazione del numero in giù) dentro un ciclo (ricordandosi di indentare tutto ciò che deve rimanere dentro il ciclo).

```
import random
while True:
    mol = random.randint(0,12)
    # mol sta per moltiplicando
    print(f"Quanto fa {mol} per 9?")
    # f"" è una sintassi che consente di stampare a schermo anche il contenuto
    # di una variabile se questa si inserisce tra {}
    num = int(input())
    if num == mol*9:
        print("Corretto")
```

Una volta lanciato il programma con

```
$ python3 tabelline.py
```

questo non finirà mai e si dovrà premere i tasti CTRL+C (oppure premere STOP sull'editor) che significa 'ferma il programma'

Ora possiamo inserire qualche abbellimento, come per esempio chiedere il nome del giocatore.

Questo possiamo sempre farlo con input(), ma visto che ora non dovrà essere inserito un numero, dobbiamo omettere l'inserimento di int() prima di input().

```
nome = input("Ciao, come ti chiami? ")
print(f"Ciao {nome}, ho alcune domande per te.")
while True:
    ...
```

In caso di errore possiamo magari dire quale è il risultato corretto.

```
if num == mol*9:
    print("Corretto")
else:
    print(f"Sbagliato. La risposta corretta è {mol*9}")
```

E poi magari possiamo chiedere se vuole continuare a giocare o piuttosto smettere in modo da evitare di dover uscire con CTRL+C.

per questo dobbiamo aggiungere un'altra input() che assegneremo alla variabile 'risp' ('risposta') in cui l'utente metterà 'S' o 'N' (Si o No). Dovremo quindi controllare cosa ha risposto e se è 'n' allora eseguire un nuovo comando 'break' che consente di uscire dal ciclo while senza dover mettere una condizione al posto di True.

```
    risp = input("Vuoi continuare? (S/N) ")
    if risp == "n" or risp == "N":
        break
    else:
        print ("Ecco la prossima:")
```

la 'or' significa 'o' e serve perchè il giocatore potrebbe aver inserito una N minuscola o una N maiuscola.

ecco il programma completo:

```
$ edit tabelline.py
```

```
import random

while True:
    # genero il moltiplicando causale
    mol = random.randint(0,12)
    # chiedo la risposta al giocatore e la metto in num
    print(f"Quanto fa {mol} per 9?")
    num = int(input())
    # verifico se ha indovinato
    if num == mol*9:
        print("Bravo, hai indovinato!")
    else:
        print(f"Mi dispiace, hai sbagliato. La risposta corretta era {mol*9}")
    # vediamo se il giocatore vuole ancora continuare
    risp = input("Vuoi continuare? (S/N) ")
    if risp == "n" or risp == "N":
        break
    else:
        print ("Ecco la prossima moltiplicazione:")
print("Arrivederci")
```

```
$ python3 tabelline.py
```