Come si scrivono cose in python

11 January 2025

Indice dei Contenuti

1 Tipi di variabile

I tipi in python sono delle classi, ovvero degli "stampini" che definiscono come si comporta una variabile quando gli si applica una funzione oppure un metodo se è previsto che questa funzione o metodo sia compatibile con questa classe.

La funzione `type(oggetto)` ritorna il tipo della variabile `oggetto`.

1.1 Booleani

```
tipo_bool = type(True)
print(tipo_bool) # Output: <class 'bool'>
```

1.2 Numeri

1.2.1 Interi

```
# qui sopra x non è definito
x = 45 # assegnazione di x a 45, un tipo int
# adesso x è uguale a 45

x = x + 2 # riassegno x a 45 e gli sommo 2
# adesso x è 47
```

1.2.2 Decimali o floating point o virgola mobile

```
# inizializzazione di un numero con la virgola, tipo float
chiuaua = 75.3
# i numeri con la virgola si scrivono usando il punto, la virgola viene
usata per le sequenze di valori, si vedranno poi nelle collezioni
```

*il nome delle variabili è arbitrario e scelto dal programmatore ma è bene che abbiano un nome che rappresenti quello che contiene.

1.3 Caratteri e stringhe

In python non c'è differenza, le cose che si possono fare con una stringa, sono le stesse che si possono fare con i caratteri.

```
# assegnare una variabile ad una stringa
questa_stringa = "aprire e chiudere doppie virgolette"
un_altra = 'oppure i singoli quotes/apici/apostrofi'
carattere1 = "" # carattere vuoto
carattere2 = "a" # carattere lettera a
carattere3 = "A" # è diverso da carattere2

# comparazione
print(carattere2 == carattere3) # è falso a causa della case \
sensitivity, cioè a piccolo è diverso da A grande

# Accedere ad una lettera e assegnarla ad una nuova variabile
un_char = questa_stringa[0]

#"aprire e chiudere doppie virgolette"
# 0:"a" 1:"p" 2:"r" 3:"i" 4:"r" 5:"e" 6:" " e così via...

print(un_char == "a") # output: True
```

2 IF/ELIF/ELSE

La keyword `pass` in questo esempio dice di andare avanti perchè altrimenti un blocco if privo di istruzioni non sarebbe valido. Il seguente codice fa solo vedere la struttura di un blocco if/elif/else

2.1 sintassi

```
if condizione1:
  # codice che viene eseguito se condizione ritorna un valore True
  pass
# opzionale valido solo se viene messo dopo un blocco if o dopo un
altro elif
elif condizione2:
  #esegui se condizione2 è vero
  pass
# ce ne possono essere quanti se ne vuole di elif
elif condizione3:
  #esegui se condizione2 è vero
  pass
```

2.2 Codizioni

Le condizioni sono espressioni che ritornano booleani, tipi bool:

```
3 < 4 # viene valutato da python come True

x = 3
y = -1
print(x < y) # output: False
```

3 Cicli

3.1 for loop

3.2 while loop

4 Collezioni

Servono per "collezionare" più tipi di dato, dello stesso tipo oppure diverso in una **struttura dati**.

Una **struttura dati** è un modo di rappresentare dei dati in modo schematico, con un certo pattern, in modo che sia agevole accedervi dal punto di vista di scrittura del codice, quindi che sia facile da leggere, intuitivo e possibilmente conciso.

4.1 Tuple

Sono sequenze ordinate di elementi e **non** sono modificabili/mutabili.

- Si possono usare come chiavi purchè gli elementi contenuti siano a loro volta solo tipi immutabili, quindi altre stringhe, numeri o tuple. (ricorsivamente)

```
tup_example = (1,2,3,4)

# equivalente a
tup_example = 1,2,3,4

# accesso
tup_example[2] # terzo elemento, corrisponde a 3

# destrutturamento
uno, due, tre, quattro = 1,2,3,4
print(
  uno == 1 and
  due == 2 and
  tre == 3 and
  quattro == 4
) # output: True
```

4.2 Liste

4.3 Dizionari

4.4 Set (Insiemi)