**CASTING E INPUT**

Il casting funziona come in tutti i linguaggi di programmazione. Per capire bene il funzionamento partiamo direttamente da un esercizio.

Immagine che contiene testo, lettera

Descrizione generata automaticamente

Soluzione:

hours = float(input("How many hours did you work last month? "))  
hourly\_rate = float(input("What is your hourly rate? "))  
  
  
print("Last month, you earned",hours\*hourly\_rate,"dollars")

**PROBLEMI CON I FLOAT**

Si ricordi che tutto ciò che scriviamo in Python è scritto in linguaggio di alto livello. In realtà, il PC “ragiona” in linguaggio di basso livello: tutti i caratteri scritti in linguaggio di alto livello diventeranno, a livello macchina, solamente 0 e 1. I float, scritti in linguaggio di alto livello, a basso livello vengono approssimati. Questo significa che, un numero 2.5 scritto in linguaggio di alto livello, non sarà utilizzato direttamente dal pc per fare i calcoli. Verrà prima infatti trasformato in linguaggio macchina, ossia diventerà una sequenza di numeri 0 e 1. Nel processo di trasformazione qualcosa si perderà 🡺 Il computer lavorerà con una sua approssimazione. Vediamo un esempio.

Codice Python:

num1 = 4.3  
num2 = 1.4  
  
print(num1, num2)  
  
print(num1+num2)

Console:

Immagine che contiene testo

Descrizione generata automaticamente

La somma 4.3 + 1.4 dovrebbe dare 5.7. Tuttavia, passando dal linguaggio ad alto livello al linguaggio macchina qualcosa si perde 🡺 La loro somma sarà molto simile a 5.7, ma non uguale.

**CALCOLI TRA ESPONENZIALI**

Python esegue i calcoli da sinistra a destra. Supponiamo di avere a che fare con vari esponenziali. Quale sarà il risultato?

Codice Python:

num1 = 2  
num2 = 3  
num3 = 4  
  
  
ris = num1\*\*num2\*\*num3   
  
print(ris)

In questo caso, Python effettua il calcolo partendo da destra, e non da sinistra. Vediamo quindi cosa accade.

Per prima cosa considera num3, e lo eleva per num2. Abbiamo quindi = 3 \* 3 \* 3 \* 3 = 81.

Andando ancora verso sinistra avremo num1: num1 verrà quindi elevato ad 81:

Immagine che contiene tavolo

Descrizione generata automaticamente

Vediamo se Python ci da lo stesso risultato:

Console:

Immagine che contiene testo

Descrizione generata automaticamente

In sintesi, si supponga di avere i seguenti numeri:

a,b,c,d,e.

Si parte da destra. Si calcoli . Chiamiamo x il risultato di questo esponente. Si va verso sinistra: troviamo c. Eleveremo c per il risultato precedente: . Chiamiamo y questo risultato. Andando avanti:

= k

* Risultato =

**PRECEDENZA OPERATORI**

Gli operatori aritmetici hanno una precedenza. Per ricordarla si deve ricordare l’acronico PEMDAS (Parenthesis, Exponent,Multiplication,Division,Addition,Subtraction)

**Immagine che contiene tavolo

Descrizione generata automaticamente**

**PARAMETRI END E SEP**

Studiamo ora i parametri sep ed end che si passano al richiamo della fiunzione print.

Consideriamo le due seguenti strighe:

Codice Python:

str1 = "Ciao, mi chiamo Orsobindo e ho 12 anni"  
  
str2 = "Vivo a roma e qui c'è un tempo di schifo"  
  
print(str1, "\n",str2)

Console:

Immagine che contiene testo

Descrizione generata automaticamente

Ricordiamo che, nella funzione print, la virgola crea uno spazio!

Supponiamo ora di voler inserire un punto alla fine delle nostre frasi. Come facciamo? Si utilizza il parametro end.

Codice Python:

str1 = "Ciao, mi chiamo Orsobindo e ho 12 anni"  
  
str2 = "Vivo a roma e qui c'è un tempo di schifo"  
  
print(str1, "\n",str2, end=".")

Console:

Immagine che contiene testo

Descrizione generata automaticamente

Il punto è stato aggiiunto.

Vediamo ora il parametro sep. Questo parametro fa si che, alla sinistra delle stringhe inserite come argomento della funzione print, venga inserito un testo a scelta dell’utente

Vediamo un esempio.

Codice Python:

nome = input("Inserisci il tuo nome: ")  
cognome = input("Inserisci il tuo cognome: ")  
eta = int(input("Inserisci la tua eta: "))  
  
print("Mi chiamo",nome,cognome,"ed ho", eta,"anni.","Sono proprio bello!",sep="YYY")

Console:

Immagine che contiene testo

Descrizione generata automaticamente

Quindi, tra una stringa e l’altra, viene inserito il testo usato come separatore.

Si noti che, di solito, tra un valore e l’altro all’interno di print viene stampato uno spazio. Questo significa che, di default, il sep equivale allo spazio!!

Mostriamolo.

Codice Python:

# DEFINIZIONE STRINGHE

str1 = "Pera"

str2 = "Mela"

str3 = "Banana"

# Funziona print() senza sep

print(str1,str2,str3)

print()

# Funzione print() con sep vuoto

print(str1,str2,str3,sep = "")

print()

# Funzione print() con sep spazio

print(str1,str2,str3,sep = " ")

print()

# Funzione print() con sep asterisco

print(str1,str2,str3,sep = "\*")

print()

Console:

Immagine che contiene testo

Descrizione generata automaticamente

Quindi, utilizzare la funzione print senza il parametro sep, equivale ad utilizzare la funzione print con la funzione sep=” “ !!!!

NB. Prima della prima stringa non viene aggiunto nulla.

Ovviamente è possibile combinare i due parametri visti finora, ossia sep ed end.

Modifichiamo quindi il nostro ultimo esempio aggiungendo il parametro end:

Codice Python:

nome = input("Inserisci il tuo nome: ")  
cognome = input("Inserisci il tuo cognome: ")  
eta = int(input("Inserisci la tua eta: "))  
  
print("Mi chiamo",nome,cognome,"ed ho", eta,"anni.","Sono proprio bello!",sep="YYY",end="Ciao a tuttii")

Console:

Immagine che contiene testo

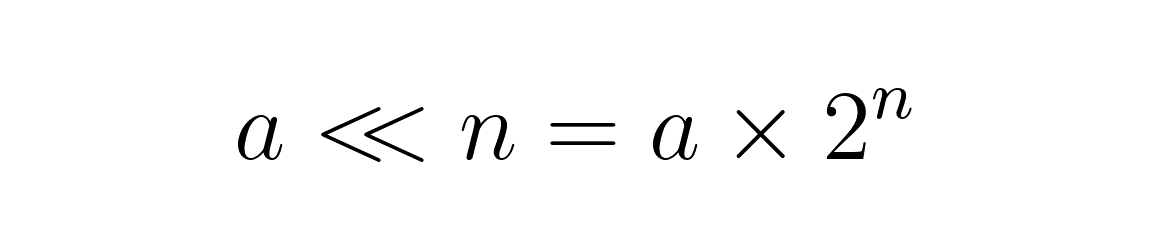
Descrizione generata automaticamente

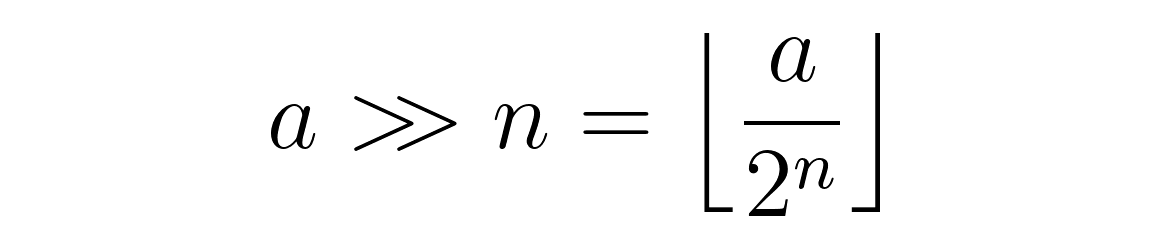
Si noti che la posizione di sep ed end all’interno del print è molto importante. Esse vanno inserite come ultimi parametri

Se si trovano in posizioni diverse, otterremo un errore in console.

**OPERAZIONI CON I BIT**

**SHIFT OPERATIONS:**





Il risultato di una shift operation sarà sempre e solo il numero intero, I decimali non vengono presi in considerazione.

**LOGICAL OPERATIONS.**

Anche gli operatori logici danno come risultato 0 o 1. Per ricordare i risultati, bisogna tener presente le seguenti tabelle:

**TABELLA OPERATORI LOGICI**

**Immagine che contiene tavolo

Descrizione generata automaticamente**

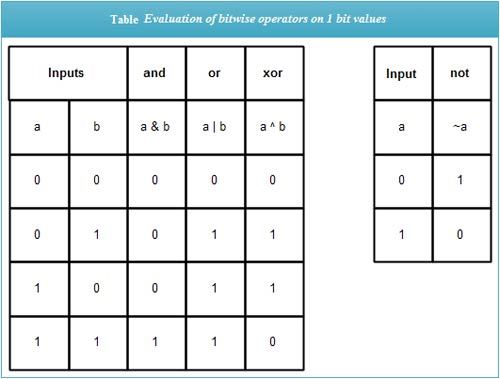
Ovviamente a e b possono assumere valori 0 oppure 1.

Compound operator:

Immagine che contiene tavolo

Descrizione generata automaticamente

Nella seguente tabella vengono riportati tutti i casi per i quali abbiamo and, or, xor etc



**TRASFORMAZIONE DA BINARIO A DECIMALE E DA DECIMALE A BINARIO**