

1 Variabile (din nou...)

Hai să folosim și variabile de tip numeric și să trecem, în sfârșit, la calcule serioase:

```
aa = 67
bb = 89
suma = aa + bb
print(suma)
```

Pare OK! Dar prea banal. Ar trebui să afișăm un mesaj mai clar.



Modifică ultimul rând astfel:

```
print(" Suma numerelor", aa , "si" , bb , "este " , suma)
```

Sau așa:



```
print("Suma numerelor " + str(aa) + " si " + str(bb) + " este chiar " + str(suma))
```

Observă că am fost nevoiți să folosim **funcția str()** care transformă un număr într-un string. Numai așa am avut voie să le concatenăm (să le lipim cu semnul +)

Putem folosi funcția print și așa:

```
print("Suma numerelor %s si %s este: %s: " %(aa,bb,suma) )
```

Semnul % indică includerea în mijlocul textului (cu o anumită formatare) a variabilelor enumerate la capătul liniei după semnul %.

Și acum să calculăm și media aritmetică:

```
media = ( aa + bb)/2
print("Media numerelor " + str(aa) + " si " + str(bb) + " este " + str(media))
```

În acest caz simplu al nostru, nici nu era nevoie de o variabilă în plus pentru medie sau pentru sumă, Python știe să facă direct calculele în interiorul **funcției print()**.
Așfel, puteam scrie direct:

```
media = ( aa + bb)/2
print("Media numerelor " + str(aa) + " si " + str(bb) + " este " + str((aa + bb)/2) )
```

Depinde cât de clar vrem să fie codul nostru sau câte linii de cod vrem să economisim.
Putem scrie și așa, și așa!

```
a = 6.78
a = round(a)
print(a)
```

e echivalent cu:

```
a = 6.78
print( round(a))
```

Testează ambele variante! Ce face **funcția round()** ?

```
a = 10/3
print(round(a,2))
```

Aici funcția `round(a,2)` ne va returna rezultatul rotunjit la două zecimale.

2 Mai multe despre Strings

Strings au foarte multe metode, unele dintre ne vor fi folositoare când avem de prelucrat texte. Hai sa explorăm câteva din ele și să vede ce se poate face cu un șir de caractere.

Scrie în fereastra Python Shell :

```
>>>text = "Python is my favorite language"
```

Această propoziție este văzută de Python ca un string.

```
>>> len(text)
```

Rezultatul e 30, pentru ca acest string are 30 de caractere (inclusiv spațiile)

```
>>> text.index("n")
```

Rezultatul e 5, adică prima apariție a caracterului "n" are index-ul 5

Știi de ce e 5 și nu 6? **Pentru că indexul unui string începe întotdeauna de la 0.**

Deci

```
P y t h o n   i s   m y   . . . .
0 1 2 3 4 5 6 7 8 9 10 11 . . . .
```

```
>>> text.count("n")
```

Rezultatul e 2, câte caractere "n" se află în acest string.

```
>>> text[3:6]
```

Aceasta va afișa "hon", adică o "felie" din string, și anume de la caracterul 3 la 5 (marginea superioară nu se ia) (ține minte: indexul unui string începe întotdeauna de la 0)

```
>>> text[3:]
```

Aceasta va afișa "hon is my favorite language" pentru că dacă oțim numărul de după : se ia de la 3 până la capăt

```
>>> text[:3]
```

Aceasta va afișa "Pyt", pentru că am omis numărul de dinainte de : se ia de la început până la indexul 2.

```
>>> text[-3:]
```

Aceasta va afișa "age", pentru că am folosit un număr negativ, deci se va începe numărarea invers, de la coadă.

```
>>> text[:-3]
```

Aceasta va afișa "Python is my favorite langu"

```
>>> text.startswith("Py")
```

Aceasta va afișa True, pentru că stringul începe cu "Py"

```
>>> text.split(" ")
```

Aceasta va afișa o listă
['Python', 'is', 'my', 'favorite', 'language']
obținută prin separarea sub-stringurilor cu caracterul spațiu

