**Subprograme în Python**

Definirea unei funcții în Python se face prin:

**def nume\_funcție(eventualii parametri)**

**cod**

**.....**

**return eventualele\_valori**

Particularități / asemănări / deosebiri cu C++

- NU apare tipul de date întors, prin urmare din antetul funcției NU putem să ne dăm seama dacă o funcție are rezultat void sau de un anumit tip.

- Acest lucru se stabilește DOAR după faptul că return este de unul singur sau după return apar expresii

- În mod nativ NU putem avea parametri prin referință (din cei care să se schimbe în urma apelului)

- O variabilă inițializată înaintea funcției este globală, dar recunoscută doar în programul principal (adică NU și în funcție). Dacă și funcția conține variabile cu același nume, atunci acestea din funcție sunt locale (și le "umbresc" pe cele globale).

Dacă vrem ca o anumită variabilă globală să fie recunoscută într-o anumită funcție, în corpul acelei funcții scriem

"**global variabilă**"

Vezi aplicațiile comentate

Obs: Ca și în C++, **return** produce ieșirea imediată din funcție (întrerupe orice cod ar fi fost în curs de executare în aceasta.

Dacă funcția este de tip void, putem să NU mai punem return (la atingerea sfârșitului de funcție se va ieși din aceasta)