

1 Ghicește numărul

Acum ești pregătit să scrii primul program adevărat în Python! Let's start!

Jocul e următorul: Calculatorul (mai exact programul scris de noi) va alege absolut la întâmplare un număr iar utilizatorul va trebui să îl ghicească. Vom dezvolta apoi programul astfel încât să alegem intervalul din care poate fi numărul, să îl ajutăm pe utilizator spunându-i de fiecare dată dacă a ales un număr prea mare sau prea mic, apoi îi vom ține și un scor al încercărilor.

```
import random
SecretNum = random.randrange(1,7)
print(" GHICESTE NUMARUL !")

Raspuns = input("Care e raspunsul tau? ")
Raspuns = int(Raspuns)

while (Raspuns != SecretNum):
    print("Mai incearca ! ")
    Raspuns = input("Care e raspunsul tau? ")
    Raspuns = int(Raspuns)
print ("Felicitari, l-ai ghicit!")
```

Observă prima linie **import random**.

Aici vom importa o bibliotecă (library) Python. Sunt multe biblioteci Python care ne vor ajuta. Ele sunt seturi de programe gata scrise, grupate după utilitate

Aceasta (random) se ocupă de generarea numerelor aleatoare. Vom apela o metodă a ei, numită `randrange()` pentru a obține numărul secret.

O bibliotecă se folosește întotdeauna așa: **library.metoda()**

Observă că am folosit iar funcția `int()`. Știi de ce?

Provocare: Cum se puteau scrie rândurile 5 și 6 într-unul singur?

2 Dezvoltarea no.1- Ajutorul

Acesta a fost doar scheletul programului nostru. Hai să adăugăm testarea numărului (dacă e mai mare sau mai mic decât numărul secret). Modifică de la rândul 5 în jos:

```
Raspuns = int(input("Care e raspunsul tau? "))

while (Raspuns != SecretNum):
    if (Raspuns < SecretNum):
        print(" Prea mic! Incearca din nou!")
    elif (Raspuns > SecretNum):
        print(" Prea mare! Incearca din nou!")
    Raspuns = int(input("Care e raspunsul tau? "))
print ("Felicitari, l-ai ghicit!")
```

Fii atent la **indentari** (alinieră rândurilor în structurile IF și WHILE)

3 Dezvoltarea no.2 - Scor

Să adăugăm și un scor al numărului de încercări. Adaugă înainte de structura WHILE

scor = 0, iar în interiorul lui WHILE, chiar la început adaugă **scor = scor + 1**.

Modifică astfel și comanda print de la final:

```
print ("Felicitari, l-ai ghicit din numai " +str(scor) + " incercari!" )
```

Provocare: Modifică programul astfel încât utilizatorul să nu aibă decât 4 încercări. După a patra încercare nereușită, programul să se termine cu mesajul „Sorry! Numar de incercari epuizat!”



4 Dezvoltarea no.3a - Error prone

Dar dacă utilizatorul introduce o literă în loc de un număr? Hai să protejăm programul împotriva acestei erori. Înlocuiește fiecare linie în care ai un **input** cu următorul cod:

```
Raspuns = input("Care e raspunsul tau? ")
while not Raspuns.isdigit():
    print("Te rog sa introduci numai cifre!")
    Raspuns = input("Care e raspunsul tau? ")
Raspuns=int(Raspuns)
```

5 Dezvoltarea no.3b - Error prone cu TRY / EXCEPT

Dar și mai elegant s-ar face tratarea erorilor cu o secvență TRY... EXCEPT

```
try:
    Raspuns = int(input("Care e raspunsul tau? "))
except ValueError:
    print("Te rog sa introduci numai cifre!")
```

Se încearcă executarea secvenței TRY, dacă se iveste o eroare, atunci ea va fi „tratată” automat în secvența EXCEPT, iar utilizatorul nu mai primește o eroare.

6 Dezvoltarea no.3c - Error prone cu TRY / EXCEPT

Dar noi insistăm, și dorim ca utilizatorul să introducă obligatoriu un număr (deci numai cifre). Vom folosi din nou o buclă WHILE. Atâta timp cât True (adică întotdeauna) se încearcă execuția secvenței TRY cu acel input. Dar, există totuși posibilitatea să iasă din bucla aceasta. Dacă se reușește un input corect, adică nu intră pe ramura de eroare (cu Except) atunci se va ieși cu **break** din această buclă WHILE.

```
while True:
    try:
        Raspuns = int(input("Care e raspunsul tau? "))
        break
    except ValueError:
        print("Te rog sa introduci numai cifre!")
```

Încearcă să rulezi acum programul cu toate modificările făcute. Codul de mai sus trebuie folosit în loc de primul input (rândul 1 din Dezvoltarea no.1), dar la următorul input (rândul 8 din Dezvoltarea no.1) e suficient să folosești o secvență try simplă, fără while (Dezvoltarea no.3a), deoarece ne aflăm deja într-o structură repetitivă.

