

Computação 1 - Python

Aula 8 - Teórica: Estrutura de Repetição : for

Estrutura de Repetição *while*

Estrutura que permite a repetição de um conjunto de comandos. Até o momento vimos o *while*:

```
while condição:  
    conjunto de comandos
```

- Com *while* podemos implementar qualquer algoritmo que envolva repetição.
- **DICA:** o *while* é mais recomendado quando não se sabe ao certo quantas vezes a repetição será feita, pois a **condição** é um teste booleano qualquer e não necessariamente uma contagem.

Estrutura de Repetição *while*

Lembre: Faça uma função que gere números aleatórios entre 1 e 10 e calcule a soma destes números até que seja gerado o número 5.

```
from random import randint
# sem entrada → int

def somaAleatoria():
    soma = 0
    numero = randint(1,10)
    while numero != 5:
        soma = soma + numero
        numero = randint(1,10)
    return soma
```

O número de repetições dos comandos associados ao laço *while* depende de quando sair o número 5. Podem ser 2 vezes ou 1000 vezes!

Estrutura de Repetição *while*

Faça uma função que some 10 números gerados aleatoriamente no intervalo de 1 a 5.

Como seria essa função com *while*?

Estrutura de Repetição *while*

Faça uma função que some 10 números gerados aleatoriamente no intervalo de 1 a 5.

Como seria essa função com *while*?

```
from random import randint
# Função que soma 10 números gerados aleatoriamente
# no intervalo de 1 a 5
# sem entrada → int
def soma10():
    contador = 0
    soma = 0
    while contador < 10:
        numero = randint(1,5)
        soma = soma + numero
        contador = contador + 1
    return soma
```

O número de repetições será 10 em qualquer execução do programa, independente dos números aleatórios gerados.

Estrutura de Repetição *for*

Faça uma função que some 10 números gerados aleatoriamente no intervalo de 1 a 5.

```
from random import randint
# Função que soma 10 números gerados aleatoriamente
# no intervalo de 1 a 5 - usando for
# sem entrada → int
def soma10usandofor():
    soma = 0
    for contador in range(10):
        numero = randint(1,5)
        soma = soma + numero
    return soma
```

```
for var in range(n):
    comandos
```

OU

```
for var in [0,..., n-1]:
    comandos
```

No programa acima, a variável *contador* vai assumir os valores 0, 1, 2, 3, 4, 5, 6, 7, 8 e 9.

Estrutura de Repetição

IMPORTANTE: diferença de uso entre *while* e *for*:

- **While:** decisão sobre repetir ou não baseia-se em teste booleano. Risco de loop infinito. :-)
- **For:** Contagem automática do número de repetições.

Estrutura de Repetição *for*

- A função `range(...)` pode ter 1, 2 ou 3 argumentos:
 - `range(numero)`: faz com que a variável do **for** assuma valores de 0 a `numero-1`
`for x in range(10) → x recebe 0,1,2,...,9`
 - `range(inf,sup)`: faz com que a variável do **for** assuma valores de `inf` a `sup-1`
`for x in range(3,8) → x recebe 3,4,5,6,7`
 - `range(inc, sup, inc)`: faz com que a variável do **for** assuma valores de `inf` a `sup-1` com incremento de `inc`
`for x in range(3,8,2) → x recebe 3,5,7`

Estrutura de Repetição *for*

Faça um programa que determina a soma de todos os números pares desde 100 até 200. (Usando **for** ao invés de **while**)

Estrutura de Repetição *for*

Faça um programa que determina a soma de todos os números pares desde 100 até 200. (Usando **for** ao invés de **while**)

```
# Função que soma todos os números pares
# de 100 até 200
# sem entrada -> int
def somaPares():
    Soma = 0
    for Par in range(100,202,2) :
        Soma = Soma + Par
    return Soma
```

Estrutura de Repetição *for*

Resolva usando *for*:

1. Faça uma função que calcule o valor de $N!$, onde N é passado como parâmetro. (Sem usar o factorial do módulo *math*).
2. Faça uma função que calcule e retorne o valor de

$$S = \frac{1}{1} + \frac{3}{2} + \frac{5}{3} + \frac{7}{4} + \dots + \frac{99}{50}$$

3. Faça uma função que calcule e retorne o valor de

$$S = \frac{1}{1} - \frac{2}{4} + \frac{3}{9} - \frac{4}{16} + \dots - \frac{10}{100}$$

Computação 1 - Python

Aula 8 - Teórica: Estrutura de Repetição : for