

MC-102 — Aula 06

Comandos Repetitivos

Instituto de Computação – Unicamp

20 de Agosto de 2019

O comando **for**

- Estrutura:

```
for variável in lista:  
    comando(s)
```

- Para cada elemento da lista, em ordem de ocorrência, é atribuído este elemento à variável e então é executado o(s) comando(s).

O comando **for**

- O programa abaixo usa o laço **for** para imprimir números de uma lista.

```
a = [1, 21, 31]
for i in a:
    print(i)
```

A função `range`

- É comum fazermos um laço **for** iterar sobre valores numéricos.
- Em Python o comando **`range(n)`** gera uma “quase” lista¹ com os valores de 0 até $n - 1$.
- O programa abaixo imprime os números de 0 até 9.

```
range(10)
```

```
list(range(10))
```

```
for i in range(10):  
    print(i)
```

¹é um iterator na verdade, mas funciona como uma lista para nossos propósitos aqui 

A função `range`

- Podemos especificar um intervalo de valores na função `range`:
 - ▶ `range(i, f)`: gera-se números de i até $f - 1$.
- O programa abaixo imprime os números de 5 até 9.

```
for i in range(5,10):  
    print(i)
```

A função `range`

- Podemos especificar um passo a ser considerado no intervalo de valores da função `range`.
 - ▶ `range(i, f, p)`: gera-se valores a partir de i com incremento de p até $f - 1$.
- O programa abaixo imprime os números pares entre 0 e menores que 13.

```
for i in range(0,13,2):  
    print(i)
```

O comando **for**

Imprimindo os n primeiros números inteiros:

```
n = int(input("Digite um número:"))
for i in range(1,n+1):
    print(i)
```

Leitura

Leia 5 dados inteiros e coloque-os numa lista

```
lista = []  
for i in range(5):  
    dado = int(input('Entre com o dado:'))  
    lista.append(dado)
```


Faça uma função **leia5** que le 5 dados inteiros e retorna um lista com esses dados

```
def leia5():  
  
    lista = []  
    for i in range(5):  
        dado = int(input('Entre com o dado: '))  
        lista.append(dado)  
    return lista
```

Leitura

Faça uma função **leian** que le n dados inteiros e retorna um lista com esses dados

```
def leian(n):  
  
    lista = []  
    for i in range(n):  
        dado = int(input('Entre com o dado: '))  
        lista.append(dado)  
    return lista
```

Leitura

Faça uma função **soma** recebe uma lista de numeros e retorna a soma deles

```
def soma(lista):  
  
    s = 0.0  
    for x in lista:  
        s = s + x  
    return s
```

Leitura

Faça um programa que le n , depois le n dados e imprime a média deles.

- use o **soma** e **leian** acima
- faça uma solução que não cria a lista de dados

a soma deles

Leitura

Faça um programa que le n , depois le n dados e imprime a média deles.

- use o **soma** e **leian** acima

a soma deles

```
n = int(input('Quantidade de dados: '))  
  
l = leian(n)  
s = soma(l)  
print(s/len(l))
```

Leitura

Faça um programa que le n , depois le n dados e imprime a média deles.

- faça uma solução que não cria a lista de dados a soma deles

```
n = int(input('Quantidade de dados: '))
s=0.0
for i in range(n)
    x = int(input('dado: '))
    s = s + x
print(s/n)
```

Exercício

Faca uma função que dado uma lista retorna o maior elemento da lista

```
def maior(l):  
    ma = 0  
    for x in l:  
        if x>ma:  
            ma = x  
    return ma
```

Não esta 100% OK - e se os números da lista forem negativos?

Exercício

Faça uma função que dado uma lista retorna o índice do maior elemento da lista

```
def maior(l):  
    ma = 0  
    n = len(l)  
    for i in range(n):  
        if l[i]>ma:  
            ma = l[i]  
            pos = i  
    return pos
```


Leitura ate o fim

E se em vez de dar o n antes, o usuário entrar com uma linha vazia para indicar que ele terminou de entrar com os dados.

Neste caso temos 3 novos problemas:

- o que é lido quando o usuário não entra nada?
- como fazer um loop que a gente não sabe quantas vezes vai rodar (antes de começar o loop)
- como interromper um loop

O comando break

O comando **break** faz com que a execução de um laço seja terminada, passando a execução para o próximo comando depois do final do laço.

```
for i in range (1,11):  
    if(i >= 5):  
        break  
    print(i)  
print("Terminou o laço")
```

O que será impresso?

Laços e o comando break

O comando **break** faz com que a execução de um laço seja terminada, passando a execução para o próximo comando depois do final do laço.

```
for i in range (1,11):
    if(i >= 5):
        break
    print(i)
print("Terminou o laço")
```

O que será impresso?

Comando **while**

- Estrutura:

```
while condição :  
    comando(s)
```

- Enquanto a condição for verdadeira (True), ele executa o(s) comando(s).

Comando **while**

- Passo 1: Testa condição. Se condição for verdadeira vai para Passo 2.
- Passo 2.1: Executa comandos;
- Passo 2.2: Volta para o Passo 1.

O **while** só testa a condição no começo do loop!

O comando while

A forma mais comum de usar o **while** é

```
while True:
    comandos
    if tem que terminar:
        break
    mais comandos
```

Isso cria um loop que fica rodando (condição do while é sempre True!) até que no meio do loop você descobre que tem que terminar!

Laços e o comando **break**

```
i=1
while True:
    if(i > 10):
        break
    print(i)
    i = i+1
```

O que será impresso?

Leitura ate o fim

E se em vez de dar o n antes, o usuário entrar com uma liha vazia para indicar que ele terminou de entrar com os dados.

Neste caso temos 3 novos problemas:

- o que é lido quando o usuário não entra nada? - o string vazio
- como fazer um loop que a gente não sabe quantas vezes vai rodar (antes de começar o loop) - while True
- como interromper um loop - break

Leitura ate o fim

```
def leiafim ():  
    lista = []  
    while True:  
        x = input('Entre com os dados: ')  
        if x == '':  
            break  
        dado = int(x)  
        lista.append(dado)  
    return lista
```

Exercício

- Faça um programa que lê um número n e imprima os valores entre 2 e n , que são divisores de n .

Exercício

- Faça uma função que recebe 2 listas. Se elas não tiverem o mesmo tamanho, a função deve retornar 0 senão ela retorna o produto escalar das duas listas

Exercício

- Faça um programa que le 2 listas, uma de nomes (leitura até o usuário entrar com uma lista vazia) e a segunda lista de notas (o menor número de notas que o número de pessoas!), e imprima o nome das pessoas que tiveram nota < 5 . **

Exercício

- Faça um programa que le 2 listas, uma de nomes (leitura até o usuário entrar com uma lista vazia) e a segunda lista de notas (o menor número de notas que o número de pessoas!), e imprima o nome da pessoa com maior nota. **

Exercício

- Faça uma função que recebe uma lista e retorna True se a lista esta em ordem crescente

Exercício

- Faça uma função que recebe uma lista e retorna True se a lista for palindromica - a lista é igual da esquerda para a direita e o reverso

2,3,2,1 é palindromica

- **'asdffdsa'** é um string (que é mais ou menos uma lista) palindromico

3,4,3,2 não é palindromico

Exercício

- Faça uma função que recebe um inteiro e retorna True ele for primo.

Números de Fibonacci

- A série de Fibonacci é: 1, 1, 2, 3, 5, 8, 13, ...
- Ou seja o n -ésimo termo é a soma dos dois anteriores

$$F(n) = F(n - 1) + F(n - 2)$$

onde $F(1) = 1$ e $F(2) = 1$.

Problema

Fazer um programa que imprime os primeiros n números da série de fibonacci.