

Design de Software

Aula de Laços: for

Objetivos de aprendizado

- Uso de laços for
 - Iterando sobre uma lista
 - Iterando sobre um range()

Aquecendo os motores

1. Faça uma **função** que percorre uma lista e retorna a soma dos elementos.

```
def soma_elementos(lista):  
    soma = 0  
    i = 0  
    while i < len(lista):  
        soma += lista[i]  
        i += 1  
  
    return soma
```

Tempo: 5 minutos

repetições: for

for percorre automaticamente elementos sobre listas, strings, etc

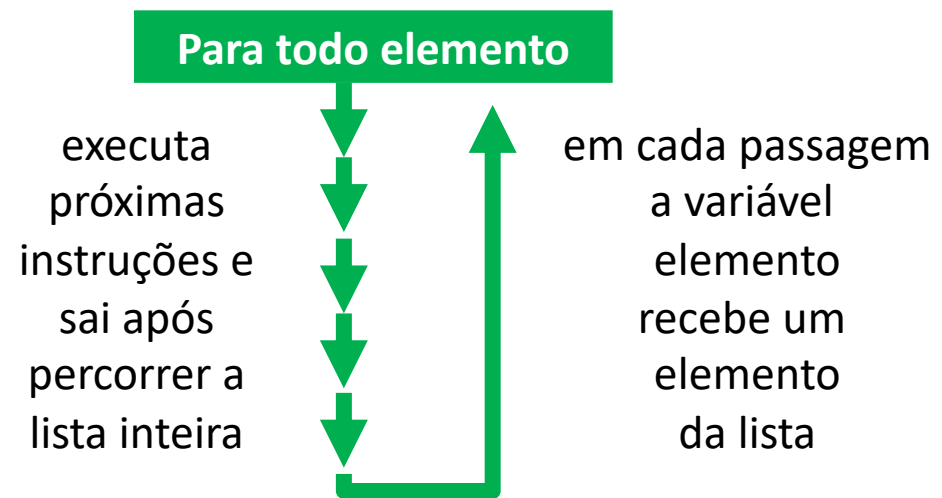
for elemento in lista:

operação 1

operação 2

...


operação n



O for é “parecido” com o while, mas NÃO possui uma condição de parada

for

Podemos ler o código abaixo:



```
for e in minha_lista:  
    print(e)
```

como:

```
Para cada elemento e em minha_lista:  
    imprima o valor do elemento e
```

for

```
lista = ['a', 1, 3.14, ['x', 'y', 'z']]

i = 0
while i < len(lista):
    e = lista[i]
    print(e)
    i += 1
```



```
lista = ['a', 1, 3.14, ['x', 'y', 'z']]

for e in lista:
    print(e)
```

for

```
lista = ['a', 1, 3.14, ['x', 'y', 'z']]
```

```
for e in lista:  
    print(e)
```

memória

lista

a	1	3.14	x	y	z
---	---	------	---	---	---

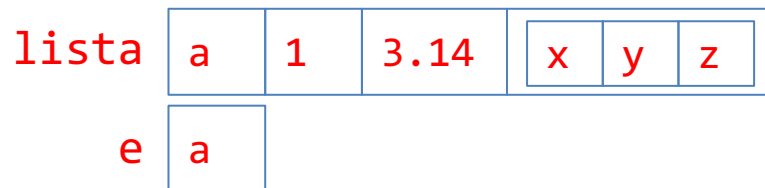
console

for

Como `lista` possui elementos, ele atribui o primeiro elemento em `e`

```
lista = ['a', 1, 3.14, ['x', 'y', 'z']]  
  
for e in lista:  
    print(e)
```

memória




console

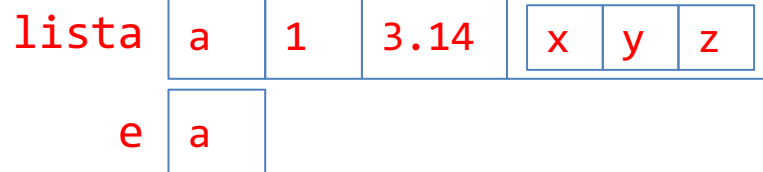
for

```
lista = ['a', 1, 3.14, ['x', 'y', 'z']]
```

```
for e in lista:  
    print(e)
```



memória



console

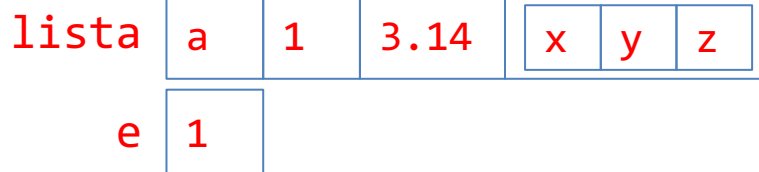
a

for

Agora **e** vai receber o próximo elemento da lista, e assim prosseguirá até terminar a lista

```
lista = ['a', 1, 3.14, ['x', 'y', 'z']]  
  
for e in lista:  
    print(e)
```

memória



console

a

For (final)

Quando acabar de percorrer a lista, sai do **for**

```
lista = ['a', 1, 3.14, ['x', 'y', 'z']]
```

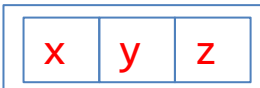
```
for e in lista:  
    print(e)
```

memória

lista



e



console

```
a  
1  
3.14  
['x', 'y', 'z']
```

For com strings

```
palavra = 'Insper'

for e in palavra:
    print(e)
```

console

```
I
n
s
p
e
r
```

For com enumerate

```
palavra = 'Insper'

for i,e in enumerate(palavra):
    print("letra[{0}] = {1}".format(i,e))
```

console

```
letra[0] = I
letra[1] = n
letra[2] = s
letra[3] = p
letra[4] = e
letra[5] = r
```

For com range()

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

De 0 até **ANTES** daqui

console

0
1
2
3
4

```
for i in range(1, 5):  
    print(i)
```

Desde aqui...

console

1
2
3
4

```
for i in range(0, 5, 2):  
    print(i)
```

Pula de 2 em 2

console

0
2
4

Exercícios - for

1. Faça um programa que conta (imprime na tela) de 0 a 10 usando *for* e *range*.
2. Faça um programa que peça 10 valores para os usuários e os coloque em uma lista. Em seguida:
 - a) Imprima a soma dos elementos
 - b) Imprima o maior elemento da listaUse repetição do tipo *for*.

Soluções

```
1) for i in range(11):  
    print(i)
```

```
2) valores = []  
    for i in range(10):  
        x = float(input("Digite um valor: "))  
        valores.append(x)
```

```
soma = 0.0
```

```
maior = valores[0]
```

```
for val in valores:
```

```
    soma += val
```

```
    if val > maior:  
        maior = val
```

O velho truque do “maior”!

```
print("Soma = {0}, maior valor = {1}".format(soma, maior))
```


Insper

www.insper.edu.br