

# FUNDAMENTOS DE PYTHON

## CAPÍTULO-05 - INTERAÇÕES

1

### → ATUALIZANDO VARIÁVEIS

```
>>> x = x + 1
```

NameError: name 'x' is not defined

```
>>> x = 0
```

```
>>> x = x + 1
```

```
x += 1
```

```
x -= 1
```

(INCREMENTO)

(DECREMENTO)

### → COMANDO WHILE

→ ITERAÇÃO: muito comum

→ PYTHON IMPLEMENTA VÁRIOS RECURSOS PARA TORRAR A ITERAÇÃO MAIS FÁCIL.

```
n = 0
while n > 0:
    print(n)
    n = n - 1
print('Fim')
```

WHILE  
LOOP INDEFINIDO

→ repete ações até que uma determinada condição se torne FALSA.

→ Variável que controla o Loop:  
VARIÁVEL DE INTERAÇÃO

### → LOOPS INFINITOS E BREAK

loop infinito {  
    n = 10  
    while true:  
        print(n, end=' ')  
        n = n - 1  
        print('feito')

loop controlável {

```
while true:  
    line = input('>')  
    if line == 'feito':  
        break  
    print(line)  
    print('feito')
```

→ FINALIZANDO INTERAÇÕES COM "CONTINUE"

while True:

line = input('>')

if line[0] == '#':

continue

if line == 'feito':

break

print(line)

print('feito')

finaliza a  
INTERAÇÃO  
corrente e  
volta para o  
início do  
WHILE

→ LOOP DEFINIDO USANDO "FOR"

→ Conjunto de coisas:

- lista de palavras

- linhas em um arquivo

- lista de números

FOR

LOOP DEFINIDO

→ repete as ações através  
de um conjunto de itens  
conhecidos.

→ A INTERAÇÃO acontece enquanto  
houver itens no conjunto

amigos = ['joe', 'george', 'sibele']

for friend in amigos:

print('Feliz ano novo', friend)

print('feito')

friend  
- variável de  
interação do  
LOOP



## → PADRÃO DE LOOP

- FOR
- WHILE

→ Inicializam UMA ou MAIS variáveis antes do início do LOOP.

→ Executam alguma operação de computação com cada item no CORPO DO LOOP, possivelmente modificando as variáveis no corpo do LOOP.

→ Verificando o resultado das variáveis quando o LOOP se completa.

## → CONTAR E SOMAR

→ CONTAR número de itens numa lista:

```
count = 0
```

```
for item in [3, 41, 12, 9, 74, 15]
```

```
    count = count + 1
```

```
print('Total de itens: ', count)
```

→ SOMAR itens de uma lista:

```
total = 0
```

```
for item in [3, 41, 12, 9, 74, 15]
```

```
    total = total + item
```

```
print('Soma: ', total)
```

→ MÁXIMO E MÍNIMO

→ Achar o maior valor da lista

```
largest = None
print('Antes:', largest)
for item in [3, 41, 12, 9, 74, 15]:
    if largest is None or item > largest:
        largest = item
print('Loop:', item, largest)
print('Valor máximo:', largest)
```

→ Achar o menor valor da lista:

```
smallest = None
print('Antes:', smallest)
for item in [3, 41, 12, 9, 74, 15]:
    if smallest is None or item < smallest:
        smallest = item
print('LOOP:', item, smallest)
print('Smallest:', smallest)
```

→ PODE-SE UTILIZAR FUNÇÕES

```
def min(valor):
    { bloco de programa
      de achar mínimo
```

```
def max(valor):
    { bloco de programa
      de achar máximo
```