

Seção 3 - Condicionais e loops

Palavras chave: *Control flow* ou *Control structures*

Essa aula é inspirada em: <https://docs.python.org/3/tutorial/controlflow.html>
(<https://docs.python.org/3/tutorial/controlflow.html>)

3.1 Condicional *if*

Construções

- **if** condição:
- **elif** condição:
- **else**:

Operações booleanas

Além dos comparadores, também podemos fazer operações booleanas. Elas são úteis em estruturas condicionais. Duas delas são: **or** e **and**.

Acesse: <https://docs.python.org/3/library/stdtypes.html#> (<https://docs.python.org/3/library/stdtypes.html>).

In [32]:

```
x = 5

if x > 0:
    print('positivo')
elif x < 0:
    print('negativo')
else:
    print('nenhuma alternativa')
```

positivo

In [33]:

```
x = 'Matemática'

if x == 'Matemática':
    print(x)
```

Matemática

In [38]:

```
x = 5

if type(x) is not str:
    print(x)
```

5

In [42]:

```
x = -10

if x > 4 or x < 0:
    print(x)
```

-10

In []:

3.2 Loop *for*

- **for** variável **in** o_que_será_iterado:

Há uma série de tipos que podem ser utilizado como iteradores, como:

- range --> seus elementos são inteiros
- list --> seus elementos são os que estão contidos na lista
- enumerate --> seus elementos são tuplas que contem inteiros e valores

In [51]:

```
lista = ['a',2,'b',4,5,5,'w']

for i in range(len(lista)):
    print(i)
```

0
1
2
3
4
5
6

In [56]:

```
lista = ['a',2,'b',4,5,5,'w']  
  
for i in enumerate(lista):  
    print(i[1])
```

```
a  
2  
b  
4  
5  
5  
w
```

3.3 Loop *while*

- **while** condição:

Enquanto a condição for verdadeira, o programa executará os comandos

Uma noção sobre tipos booleanos é interessante para executar esse comando

In [59]:

```
x = 10  
  
while x > 5:  
    print(x)  
    #x = x - 1  
    x += 1  
    if x > 20:  
        break
```

```
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20
```

3.4 Exercício 1 -

Crie um condicional que, dada uma variável, faça as operações:

- Se for número positivo, retorna "Número positivo"
- Se for número negativo, retorna "Número negativo"
- Se for zero, retorna "zero"
- Se não for número, retorna "Não é um número"

In []:

3.5 Exercício 2 -

Faça um loop **for** para criar uma lista que intercale os valores das listas a seguir:

- valores = [1,2,3,4,5]
- letras = ['a','b','c','d','e']

Dica: você pode utilizar o método `.append()` que adiciona um elemento ao final da lista.

In []:

3.6 Exercício 3 -

Dado um número inteiro, faça um operador **while** que calcule o fatorial desse número. Dicas: utilize as operações `*=` e também `-=`

In []:

3.7 Desafio 1 - Números primos

Crie uma lista com os cinquenta primeiros números primos.

Dica: pode ser utilizado o comando `break` e `continue`

In []:

3.8 Desafio 2 - Combinação

Crie uma lista de todas as palavras formadas com as letras: ARARA

Dica: o problema se trata de uma permutação com elementos repetidos.

$$P_5^{3,2} = \frac{5!}{3!2!} = 10$$

Anotações conforme: IEZZI, Gelson *et al.* Matemática: 1a série. São Paulo: Atual, 1981.

Para escrever fórmulas no Markdown: <http://luvxuan.top/posts/Markdown-math/>
(<http://luvxuan.top/posts/Markdown-math/>)