

# Processamento da Informação

---

Fabrício Olivetti de França

02 de Fevereiro de 2019



## 1. Laços de Repetição II

## Laços de Repetição II

---

Um outro tipo de laço de repetição é o *for*. Seu funcionamento é usado em conjunto com a função **range** da seguinte forma:

```
def contar_ate(n):  
    for i in range(1, n+1):  
        print(i)
```

Ao aplicar a função para o argumento `contar_ate(5)`, teremos como saída:

1  
2  
3  
4  
5

A função `range(x, y)` retorna uma lista de valores sequenciais entre  $x$  e  $y - 1$ , ou seja, o intervalo aberto  $[x, y[$ .

Também podemos utilizar a função **range** com três argumentos: **range(x, y, z)** que gera a sequência de valores no intervalo  $[x, y[$ , porém incrementando o valor de **z** em **z**:

```
def conta_pares(n):  
    for i in range(2, n+1, 2):  
        print(i)
```

A saída seria para `conta_pares(10)` seria:

2

4

6

8

10



**Fizz Buzz** é um teste lógico muito utilizado para o aprendizado de iterações e condicionais. Defina uma função `imprime_fizz` que recebe um número inteiro `n` e imprima os valores de 1 a `n` de tal forma que todo valor múltiplo de 3 seja substituído por **Fizz**, todo múltiplo de 5 por **Buzz** e múltiplos de 3 e 5 por **FizzBuzz**. Defina a função `fizzbuzz` para fazer a impressão de um único número.

Defina uma função **somatoria** que recebe um inteiro  $n$  e faça a soma dos números de 1 até  $n$  (inclusive).

Defina uma função **produtoria** que recebe um inteiro  $n$  e faça o produto dos números de 1 até  $n$  (inclusive).

Defina a função `fatorial` que calcula:

$$n! = n \cdot (n - 1) \dots 1$$

Defina a função `fibonacci` que calcula o número de fibonacci  $n$ :

$$F(0) = 0$$

$$F(1) = 1$$

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

Defina a função **primo** que determinar se um número **n** é primo utilizando o comando **for** (ao invés de **while**).