



# Lógica de Programação Orientada a Objetos

# Bem-vindos!



## Objetivos

- ✓ Loops
- ✓ Funções

# WHILE SIMPLER

---

```
while True:  
    name = input('Enter your name: ')  
    print('Hello {}'.format(name))
```

# BREAK E CONTINUE

---

```
SECRET = 'xyzzzy'

while True:
    password = input('Please enter your password: ')
    if not password:
        continue
    elif password == SECRET:
        break

    print('Wrong password!')

print('Welcome!')
```

# LOOP FOR

---

```
# Defina que print do Python 3 seja compatível no Python 2
from __future__ import print_function

>>> for c in 'Hello World!':
        print(c, end=' ')
H e l l o   W o r l d !
```

# USANDO RANGE()

---

```
>>> for i in range(10):  
    print(i, end=',')  
0,1,2,3,4,5,6,7,8,9,  
  
>>> for i in range(1, 11, 2):  
    print(i, end=',')  
1,3,5,7,9,  
  
>>> for i in range(10, 0, -1):  
    print(i, end=',')  
10,9,8,7,6,5,4,3,2,1,
```

# DEFINIÇÃO DE INTERVALOS

---

- ✓ `range(n)` produz  $[0, 1, \dots, n-1]$
- ✓ `range(i, j)` produz  $[i, i+1, \dots, j-1]$
- ✓ `range(i, j, k)` produz  $[i, i+k, \dots, m]$

# LOOPS ANINHADOS

---

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



# ENUMERATE

---

```
>>> for i, c in enumerate('Hello World!'):
      print(i, c)
0 H
1 e
2 l
3 l
4 o
5
6 W
7 o
8 r
9 l
10 d
11 !
```

# FUNÇÕES

```
>>> def increment(x):  
    return x + 1  
>>> increment(3)  
4
```

- ✓ Dois pontos (:) indica o início de um bloco  
As linhas seguintes são endentadas
- ✓ Declaração de função não especifica o tipo de retorno
- ✓ Todas as funções retornam um valor (None se não for especificado)
- ✓ Tipos de dados parametrizados não são especificados

# EXERCÍCIOS

---

## ✓ Loops:

<https://www.hackerrank.com/challenges/python-loops/problem>

## ✓ Funções:

<https://www.hackerrank.com/challenges/write-a-function/problem>