

The background features a network of blue icons including a laptop, a smartphone, a globe, a Wi-Fi symbol, a hand pointing, a speech bubble, and a telephone. In the center, there is a faint silhouette of a person wearing glasses and a beard, with various symbols like a gear, a star, and a peace sign inside their head.

UNIVERSIDADE CATÓLICA DE PETRÓPOLIS
CENTRO DE ENGENHARIA E COMPUTAÇÃO
TECNÓLOGO EM REDES DE COMPUTADORES

[PCD] PRINCÍPIO DE COMUNICAÇÃO DE DADOS
|PARTE 2 | AULA 04 | Python | Estruturas de Controle|

[4]

SOCKETS | Python

Python

```
>>> class Learning:
--   def __init__(self, name, age, gender):
--       self.title = learn
--       self.subtitle = python
--       self.paragraph = everyday
--
>>> Programmer = Learning("learn", python, "everyday")
>>> print Sue
<__main__.Programmer instance at 0x32111320>
>>> print Programmer.subtitle
python
```

[4.1]

SOCKETS | Python

Estruturas de Controle

...

Estruturas de Controle:

- Desvio condicional : if
- Laços de Repetição: while e for
- Funções

Blocos:

- Os conteúdos das estruturas de controle devem ser um bloco.
- Blocos são determinados por endentação
- O estilo pep8 orienta definir em 4 espaços para endentação padrão.
- Se você estiver usando Tab para endentar, tenha certeza que seu editor converta para 4 espaços.

Exemplo de um Bloco

```
if True:  
    print('Você está certo')
```

IF simples

```
# Lembre de usar raw_input() no Python 2
x = int(input("Informe um numero :"))
if x > 0:
    msg = 'Maior que zero'
elif x == 0:
    msg = 'Igual a zero'
else:
    msg = 'Menor que zero'
print(msg)
```

Operador Ternário

```
#!/usr/bin/env python3
x = int(input("Informe um numero : "))
msg = 'Maior que zero' if x > 0 else 'Igual ou menor que zero'
print(msg)
```

While simples

```
#!/usr/bin/env python3
while True:
    nome = input('Qual o seu nome ? ')
    print('Olá {}!'.format(nome))
```

Break e Continue

```
#!/usr/bin/env python3
SECRET = 'xyzyz'
while True:
    password = input('Informe sua senha : ')
    if not password:
        continue
    elif password == SECRET:
        break
    print('Senha inválida. Tente novamente')
print('Acesso Liberado!')
```

For

```
#!/usr/bin/env python3
# Torna o print do Python 3 compatível no Python 2
from __future__ import print_function

print ("\nExemplo 1 :\n")
for c in 'Olá Mundo !':
    print(c, end=' ')
print ("\n")
```

Exemplo 1 :

O l á M u n d o !

Range - Definição de intervalos:

- `range(n)` → produz `[0, 1, ..., n-1]`
- `range(i, j)` → produz `[i, i+1, ..., j-1]`
- `range(i, j, k)` → produz `[i, i+k, ..., m]`

For – Range ()

```
#!/usr/bin/env python3
from __future__ import print_function
print ("\nExemplo 2 - Range ()\n")
for i in range(10):
    print(i, end=',')
print ("\n")
```

Exemplo 2 - Range ()
0,1,2,3,4,5,6,7,8,9,

For – Range ()

```
#!/usr/bin/env python3
from __future__ import print_function
print ("\nExemplo 3 - Range ()\n")
for i in range(1, 11, 2):
    print(i, end=', ')
print ("\n")
```

Exemplo 3 - Range ()
1,3,5,7,9,

For – Range ()

```
#!/usr/bin/env python3
from __future__ import print_function
print ("\nExemplo 4 - Range ()\n")
for i in range(10, 0, -1):
    print(i, end=',')
print ("\n")
```

Exemplo 4 - Range ()
10,9,8,7,6,5,4,3,2,1,

For – Aninhado

```
#!/usr/bin/env python3
# Torna o print do Python 3 compatível no Python 2
from __future__ import print_function

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

For – Enumerate

```
#!/usr/bin/env python3
# Torna o print do Python 3 compatível no Python 2
from __future__ import print_function

for i, c in enumerate('Olá Mundo !!!'):
    print(i, c)
```

[4.2]

SOCKETS | Python

Funções

...

Funções

```
#!/usr/bin/env python3
def incremento(x):
    return x + 1

valor=incremento(15)
print (" 15 + 1 = %d" % (valor))
```

Main function:

- O python executa seus scripts da primeira para a ultima linha.
- Não existe uma função que ele deve executar
- Mas podemos definir uma função principal ou uma “main function”.

Main Function

```
#!/usr/bin/env python3
def main():
    print("Olá Mundo !!")

if __name__ == "__main__":
    main()
```

Main function: `if __name__ == "__main__":`

- A estrutura permite determinar se o arquivo está sendo:
 - **Executado** como um **script**
 - Ou sendo **importado** como um **módulo**
- A variável “`__main__`” determina em qual contexto o arquivo está sendo executado

Main function: Boas práticas

- Sempre coloque os códigos dentro de funções ou objetos;
- Use `__name__` para controles a execução
- Crie a função `main()` com o código inicial
- Chame as outras funções a partir da `main ()`

