

Ciência das Redes

Introdução à linguagem Python

Ricardo Luders
Thiago H Silva



- Linguagem flexível (Web, GUI, Scripting, etc.)
- Foco em legibilidade e produtividade
- Bom suporte para análise de dados

- Importante para garantir que todos absorvam melhor o conteúdo do curso.

Sintaxe

```
print("Hello world!")
```

Hello world!

A maioria das linguagens não se importa com indentação

A maioria dos humanos sim

```
print ( "Teste" )  
    print( "Introdução" )
```

Não está indentado



```
if foo:
    if bar:
        baz(foo, bar)
    else:
        qux()
```

Não precisa de `{ }` como em C ou Java


```
#Uma linha de comentário
```

```
'''Um bloco  
inteiro  
de comentário'''
```

Tipos

```
# Esta é uma string  
nome = "Thiago H Silva"  
  
# Esta também é uma string  
casa = 'Curitiba, Brasil'
```

```
# Números inteiros
```

```
ano = 2010
```

```
ano = int("2010")
```

```
# Números de ponto flutuante
```

```
pi = 3.1415
```

```
pi = float("3.1415")
```

```
#Declaração de uma lista
```

```
favoritos = []
```

```
#Adicionando
```

```
favoritos.append(42)
```

```
#Extendendo
```

```
favoritos.extend(["Python", True])
```

```
# Equivalente a
```

```
favoritos = [42, "Python", True]
```

```
numbers = [1, 2, 3, 4, 5]
```

```
len(numbers)
```

```
# 5
```

```
numbers[0]
```

```
# 1
```

```
numbers[0:2]
```

```
# [1, 2]
```

```
numbers[2:]
```

```
# [3, 4, 5]
```

```

pessoa = {}

pessoa['nome'] = 'Thiago Silva'

pessoa.update({
    'medidas': [1.80, 90],
    'sexo': 'masculino'
})
```

```
peessoa = {'nome': 'Thiago', 'sexo': 'Masculino'}  
  
print(peessoa['nome'])  
print(peessoa.keys())  
print(peessoa.values())  
print(peessoa.items())
```

Thiago

dict_keys(['nome', 'sexo'])

dict_values(['Thiago', 'Masculino'])

dict_items([('nome', 'Thiago'), ('sexo', 'Masculino')])


```
usuario = {}  
usuario.update({'login': {'thiago': {'dependente': 0}}})  
  
print(usuario)  
print(usuario['login'])  
print(usuario['login']['thiago'])  
print(usuario['login']['thiago']['dependente'])
```

```
{'login': {'thiago': {'dependente': 0}}}  
{'thiago': {'dependente': 0}}  
{'dependente': 0}  
0
```

Operadores

a	=	10	#	10
a	+=	1	#	11
a	-=	1	#	10

b	=	a	+	1	#	11
c	=	a	-	1	#	9

d	=	a	*	2	#	20
e	=	a	/	2	#	5
f	=	a	%	3	#	1
g	=	a	**	2	#	100

```
animais = "gato " + "cachorro "  
animais += "coelho"  
animais
```

```
'gato cachorro coelho'
```

Logical And
a **and** b

Logical Or
a **or** b

Conjunção (\wedge)

A	B	$A \wedge B$
V	V	V
V	F	F
F	V	F
F	F	F

Disjunção (\vee)

A	B	$A \vee B$
V	V	V
V	F	V
F	V	V
F	F	F

```
# Ordering
```

```
a > b
```

```
a >= b
```

```
a < b
```

```
a <= b
```

```
# Equality/Difference
```

```
a == b
```

```
a != b
```

Controle de fluxo

```
nota = 82
if nota >= 90:
    if nota == 100:
        print('A+')
    else:
        print('A')
elif nota >= 80:
    print("B")
elif nota >= 70:
    print("C")
else:
    print("F")
```

B


```
for x in range(3):  
    print(x)
```

0
1
2

```
frutas = ['banana', 'laranja']
```

```
for f in frutas:  
    print(f)
```

banana
laranja

```
estados = {'SP': 'Sao Paulo', 'MG': 'Minas Gerais'}  
  
for chave, valor in estados.items():  
    print(chave+' - '+valor)
```

SP - Sao Paulo

MG - Minas Gerais

```
c = 0
while c < 4:
    print(c)
    c = c + 1
```

0
1
2
3

```
c = 0
while True:
    c = c + 1
    print(c)
    if c == 4:
        break
```

1
2
3
4

Entrada e saída

```
arquivo = open('resultado.txt', 'w')  
arquivo.write('a\n')  
arquivo.write('b')  
arquivo.close()
```

```
#with fecha o arquivo automaticamente  
with open('resultado.txt', 'r') as reader:  
    for line in reader.readlines():  
        print(line)
```

a

b

Funções

```
def imprime(t):  
    print(t)  
  
def main():  
    texto = "Tenha um bom curso!"  
    imprime(texto)  
  
if __name__ == '__main__':  
    main()
```

Tenha um bom curso!


```
import math  
  
print(math.sqrt(100))
```

10.0

Exceções

```
try:  
    alguma coisa:  
except:  
    ação se erro
```

Numpy

Os arrays NumPy são mais rápidos e compactos do que as listas Python.

O NumPy usa muito menos memória para armazenar dados e fornece um mecanismo de especificação dos tipos de dados.

Isso permite que o código seja otimizado.

Um array é uma estrutura de dados central da biblioteca NumPy.

Um vetor é uma matriz com uma única dimensão (não há diferença entre os vetores de linha e coluna)

Uma matriz se refere a uma matriz com duas dimensões.

Para matrizes 3-D ou dimensões superiores, o termo tensor também é comumente usado.

“ndarray” é a abreviação de “array N-dimensional”. Um array N-dimensional é simplesmente um array com qualquer quantidade de dimensões.

```
np.array([1,2,3])
```



1
2
3

The diagram illustrates a vector addition operation. On the left, a light blue vertical rectangle labeled 'data' contains the numbers 1 and 2. In the middle is a plus sign '+'. To its right is an orange vertical rectangle labeled 'ones' containing the number 1 in both its top and bottom sections. To the right of the orange rectangle is an equals sign '='. On the far right is a gray vertical rectangle containing the number 2 in its top section and the number 3 in its bottom section. This represents the element-wise addition of the 'data' vector and the 'ones' vector.

data		ones	=	
1	+	1		2
2		1		3

$$\begin{bmatrix} 1 \\ 2 \end{bmatrix} * 1.6 = \begin{bmatrix} 1 \\ 2 \end{bmatrix} * \begin{bmatrix} 1.6 \\ 1.6 \end{bmatrix} = \begin{bmatrix} 1.6 \\ 3.2 \end{bmatrix}$$

Outros detalhes na demonstração

Jupyter + conda

Ou

Colab



Demonstração de recursos do jupyter-notebook

Mais informações nos materiais complementares

Realizar a instalação do ambiente de programação, seguindo as instruções disponíveis nos materiais complementares.

- <https://docs.python.org/3/reference/>
- Alguns dos slides foram adaptados de um material disponibilizado pelo professor Nowell Strite
- Documentação oficial da biblioteca Numpy: <https://numpy.org/>