

Python

October 6, 2019

1 Conceitos Básicos

- Definição de variáveis:

```
<nome_variavel> = <valor>
```

Exemplos:

```
string = "Olá!"
inteiro = 4
decimal = 3.14
```

- Arrays

Os arrays são criados com parentesis retos [], estando os elementos do mesmo dentro destes separados por vírgulas.

```
<nome_array> = [elem1, elem2, ..., elemN]
```

Exemplo:

```
array = [1, 2, 3, "a_dog"]
```

- Adicionar elementos

Para adicionar elementos a um array utiliza-se o método `append`, seguido do elemento que se pretende adicionar.

Exemplo:

```
array.append(5)
print(array)
[1, 2, 3, "a_dog", 5]
```

- Selecionar elementos de um array

Para se selecionar elementos de um array, utiliza-se o índice do elemento que pretendemos selecionar dentro de parentesis retos, a seguir ao nome do array.

Exemplo:

```
#Selecionar a string "a dog" do array, sendo o seu indice 3
print(array[3])
a dog
```

```
#selecionar o numero 3 do array, sendo o seu indice 2, e atribuir o mesmo
num = array[2]
print(num)
3
```

- Eliminar elementos de um array

Para eliminar elementos de um array utilizada a função `del` sendo dado como o seu parametro o nome do array seguido do indice do elemento que se pretende eliminar entre parentesis retos `[]`.

Exemplo:

```
#queremos apenas que o array contenha numeros, pelo que se irá eliminar
#a string "a dog" contido no indice 3
del(array[3])
print(array)
      [1, 2, 3, 5]
```

- Adicionar elementos num indice especifico

Para se adicionar elementos ao array, num indice especifico é utilizado o método `insert` aplicado ao array seguido do indice onde se pretende adicionar e o elemento que se aprende adicionar no mesmo.

```
<nome_array>.insert(<indice>, <elemento>)
```

Exemplo:

```
#inserir novamente a string "a dog" no array no indice 3,
#onde estava anteriormente
array.insert(3, "a_dog")
print(array)
      [1, 2, 3, "a_dog", 5]
```

- Selecionar um subconjunto do array

```
<nome_array>[indice_inicial:indice_final]
```

Exemplo:

```
array[2:4]
      [3, "a_dog"]
```

- Ciclo for

```
for <condição>:
    <desenvolvimento do que fará quando se verifica a condição>
```

Exemplo:

```
for x in range(0,3):
    print "We're on time" + x
```

Quando `x` pertence ao intervalo `[0;3[` (`range(a,b)` cria o intervalo/sequencia `[a.b[`) escreve/imprime a frase `We're on time x`, substituindo `x` pelo respetivo valor

- Ciclo while

```
for <condição>:
    <desenvolvimento do que fará quando se verifica a condição>
```

Exemplo:

```
x = 1
while x<3:
    print "We're on time" + x
    x += 1
```

Enquanto `x` for inferior a 3, imprime `We're on time x`, substituindo `x` pelo respetivo valor, incrementando de seguida 1 ao mesmo

- Condicionais if/else

```
if <condição>:
    <o que faz se condição verdade>
else:
    <o que faz se condição falsa>
```

Exemplo:

```
num = 3
if num >= 0:
    print("Positivo ou zero")
else:
    print("Negativo")
```

Se a variável num for maior ou igual que zero imprime a frase "Positivo ou zero", caso contrário imprime a frase "Negativo"

- Definição de funções:

```
def <nome> (<args>):
    <corpo da função>
    termina com return>
```

Exemplo:

```
def soma (a, b):
    return a + b
```

2 Numpy

URL Tutorial:

<https://numpy.org/devdocs/user/quickstart.html>

3 Pandas

URL Getting Started:

https://pandas.pydata.org/pandas-docs/stable/getting_started/index.html

4 Matplotlib

URL Tutorial:

<https://matplotlib.org/tutorials/introductory/usage.html#sphx-glr-tutorials-introductory-usage-py>

5 scikit-learn

URL Tutorial:

<https://scikit-learn.org/stable/tutorial/index.html>