

# PYTHON FOR DATA SCIENCE

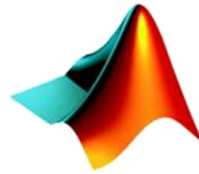
---

Prof. Nielsen Rechia

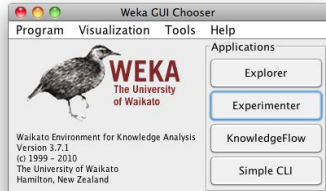
[nielsen.machado@uniritter.edu.br](mailto:nielsen.machado@uniritter.edu.br)

# Ferramentas

2



MATLAB®



# Python

3



IP[y]: IPython  
Interactive Computing



# Python

4

Linguagem de programação de propósito geral, alto nível e baseada na legibilidade

Interpretada

<https://www.python.org/about/gettingstarted/>

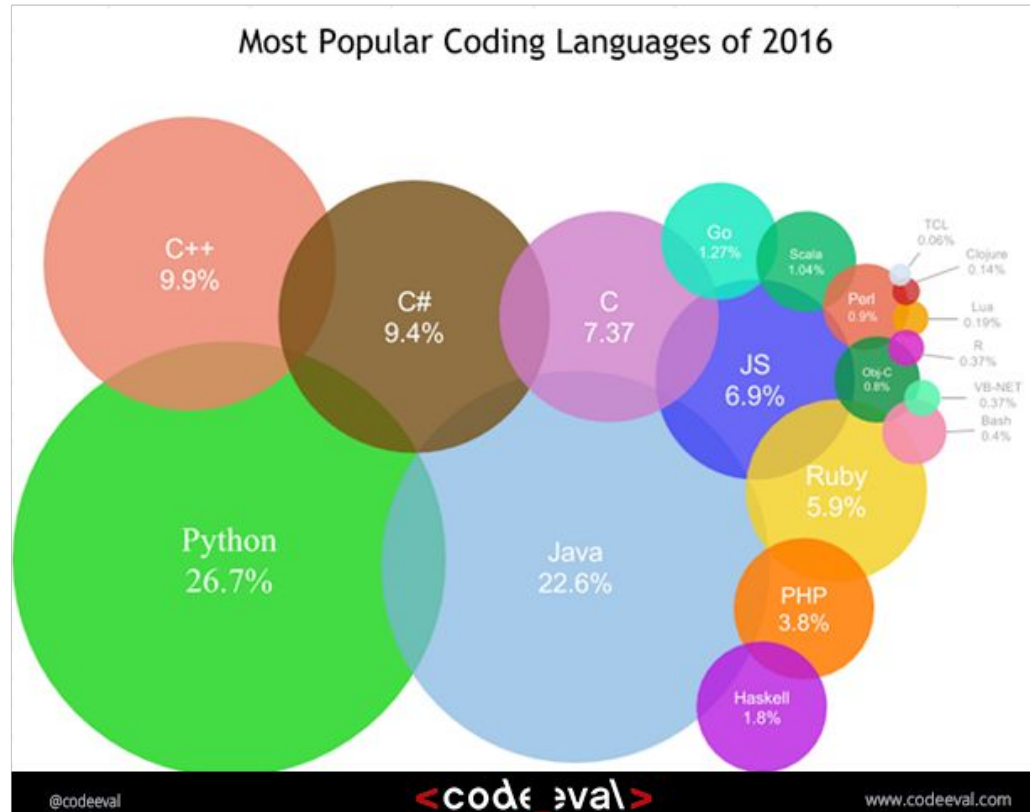
Para verificar a versão do Python:

```
python --version
```



# Python

5



# Python

6

```
def quicksort(arr):
    if len(arr) <= 1:
        return arr
    pivot = arr[len(arr) / 2]
    left = [x for x in arr if x < pivot]
    middle = [x for x in arr if x == pivot]
    right = [x for x in arr if x > pivot]
    return quicksort(left) + middle + quicksort(right)

print quicksort([3,6,8,10,1,2,1])
```

[1, 1, 2, 3, 6, 8, 10]

```
void quick_sort(int vetor[],int left,int right){
    if (right<= left) return;
    else {
        int pivot = vetor[left];    int inicio=left;
        int fim=right;
        do {
            while (left <=right && vetor[left] <= pivot) left++;
            while (vetor[right] > pivot) right--;
            if (left < right) { /* faz troca */
                int temp = vetor[left];
                vetor[left] = vetor[right];
                vetor[right] = temp;
                left++; right--;
            }
        } while (left <= right);
        /* troca pivot */
        vetor[inicio] = vetor[right];
        vetor[right] = pivot;    pivot=right;
        /* ordena sub-vetores restantes */
        if(inicio<pivot) quick_sort(vetor,inicio,pivot-1);
        if(pivot<fim)        quick_sort(vetor,pivot+1,fim);
    } }
```

# Exemplos

7

- Python e tipos de variáveis
- Operações matemáticas (int e float)
- Operadores lógicos
- Manipulação de strings
- Vetores (arrays)
- Comandos de decisão
- Laços de repetição
- Funções

Veremos usando Notebooks no Anaconda

# Exercícios

8

**Exercício #1:** Faça um função para determinar se um número é par ou ímpar.

**Exercício #2:** Faça uma função para determinar o maior número da lista:

[ -10, 10, 0, 1, 1, 7, 11, 5, 4, 3 ]

**Exercício #3:** Faça uma função para determinar a soma dos números da lista:

[ -10, 10, 0, 1, 1, 7, 11, 5, 4, 3 ]

**Exercício #4:** Faça uma função para **Fibonacci**:

**Exercício #5:** Faça uma função para a música **99 Bottles of Beer**:



# Numpy

Pacote para a linguagem Python com uma larga coleção de funções matemáticas e estatísticas

Suporta arrays e matrizes multidimensionais com alto desempenho



# Exercícios

10

**Exercício #1:** Faça uma função para determinar o maior número do array:

`[-10, 10, 0, 1, 1, 7, 11, 5, 4, 3]`

**Exercício #2:** Faça uma função para determinar a soma dos números do array:

`[-10, 10, 0, 1, 1, 7, 11, 5, 4, 3]`

**Exercício #3:** Faça uma função para determinar o desvio padrão do array:

`[-10, 10, 0, 1, 1, 7, 11, 5, 4, 3]`

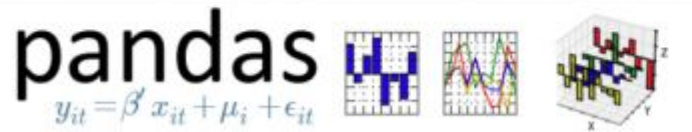
# Pandas

11

Biblioteca para manipulação e análise de estruturas de dados em Python de forma tabular.

Data Frames similares a Excel, mas cada coluna é limitada a um tipo de dado.

<https://pandas.pydata.org/pandas-docs/stable/10min.html>



# Pandas

12

**Exercício #1:** Faça o exercício proposto no jupyter

# Matplotlib

13

Biblioteca com recursos para a geração de gráficos 2D e 3D em Python.

Gráficos com alta qualidade a partir de comandos simples.



# Matplotlib

14

**Exercício #1:** Faça o exercício proposto no jupyter