



**Data Science
Academy**

www.datascienceacademy.com.br

Introdução à Lógica de Programação

Você fez o exercício que eu pedi ao final da Aula 2? Não? Sério? (o Admin vai gostar dessa...kkk).

O primeiro passo para se aprimorar tecnicamente é conhecer a si mesmo e as suas próprias deficiências e gaps de conhecimento. Por isso a autoanálise e reflexão são fundamentais e um verdadeiro exercício de humildade, em perceber que você tem muito a melhorar, como qualquer ser humano. Reflita sobre as áreas nas quais você precisa melhorar, crie um plano de ação e faça acontecer.

Agora que você já compreende o que é um algoritmo, vamos estudar vários deles, começando dos mais simples e chegando nos mais complexos, inclusive em Machine Learning. Essa é a melhor abordagem para aprender programação de computadores.

Imagine que eu tenha 2 números diferentes e queira encontrar o maior deles. Como seria o pseudocódigo nesse caso? Que tal isso:

- 1- Leia 2 números, representados pelas variáveis a e b.
- 2- Se a for maior que b, retorne a como maior.
- 3- Senão, retorne b.

Ufa! Essa foi fácil! Agora em Python. Vou criar uma função chamada `maximo2` para facilitar os testes (você já sabe que os traços devem ser substituídos por espaços ou tabs):

```
# Função
def maximo2(a, b):
    ----if a > b:
    -----return a
    ----else:
    -----return b
```

```
# Executando a função
maximo2(8, 9)
maximo2(10, 9)
```

Observe que a função `maximo2` faz exatamente o que descrevemos no pseudocódigo.

Perfeito! Vamos aumentar o grau de diversão. Que tal encontrar o maior valor entre 3 números dados? Aqui vai o pseudocódigo:

- 1- Leia 3 números, representados pelas variáveis a, b e c.
- 2- Se a for maior que b e a for maior que c, retorna a.
- 3- Senão, se b for maior que a e b for maior que c, retorna b.
- 4- Senão, retorna c

Mais fácil do que eu imaginava! Agora em Python. Vou criar uma função chamada maximo3_v1 para facilitar os testes:

```
# Função
def maximo3_v1(a, b, c):
    ----if a > b and a > c:
    -----print(a)
    ----elif b > a and b > c:
    -----print(b)
    ----else:
    -----print(c)
```

```
# Executando a função
maximo3_v1(8, 9, 10)
maximo3_v1(10, 9, 8)
maximo3_v1(8, 10, 9)
```

Mais uma vez, observe que a função maximo3_v1 faz exatamente o que descrevemos no pseudocódigo. Portanto, se conseguimos definir as operações que devem ser realizadas para resolver o problema, temos mais 50% do trabalho feito, restando apenas traduzir para uma linguagem de programação!

Mas ainda tá fácil, não é? E se eu deixar o código Python assim:

```
# Função
def maximo3_v2(a, b, c):
    ----if a < b:
    -----return maximo2(b, c)
    ----else:
    -----return maximo2(a, c)
```

```
# Executando a função
maximo3_v2(8, 9, 10)
maximo3_v2(10, 9, 8)
maximo3_v2(8, 10, 9)
```

Travou o cérebro? Ótimo, o objetivo era esse mesmo. Fazer você usar a lógica.

Todos os milhões de programadores Python no planeta Terra tem acesso exatamente ao mesmo conjunto de instruções da linguagem. Então por que tantos programas diferentes? A forma como se emprega a linguagem é que faz a diferença! Se nós podemos chamar uma função de dentro da outra, por que não fazê-lo? Por que não agrupar a sintaxe da linguagem de maneira diferente, visando resolver o problema da melhor forma possível? É onde entra o componente criatividade!

No exemplo acima, as funções `maximo3_v1` e `maximo3_v2` fazem exatamente a mesma coisa, estamos apenas reorganizando as instruções da linguagem!

Não é para você ficar apenas executando o código Python. Eu e você já sabemos que funciona. Também não é para você copiar/colar esta aula e dizer que depois estuda. É para você colocar sua mente para processar o que estamos fazendo. Faça mentalmente o exercício de percorrer o caminho dos números até a saída do código Python. Aí sim você estará exercitando programação de computadores.

Na Aula 4 criaremos o pseudocódigo para os Loops. E ainda estudaremos Pesquisa Binária, Recursão, Ordenação, Tabelas Hash, Algoritmos de Grafos, Algoritmos Gulosos, Programação Dinâmica e alguns algoritmos de Machine Learning!

Compreendendo esses algoritmos seu conhecimento em programação dará um verdadeiro salto!

#aula03