



**Data Science
Academy**

www.datascienceacademy.com.br

Introdução à Lógica de Programação

Vamos para a Aula 2 de Introdução à Lógica de Programação.

Um algoritmo é uma sequência de instruções para resolver um problema computacional bem formulado, especificado em termos de entrada e saída. Um algoritmo usa a entrada para gerar a saída.

Para resolver um problema computacional, você precisa executar as instruções especificadas pelo algoritmo.

Você usa algoritmos todos os dias, o dia inteiro. Quando acorda, você define os passos necessários para ir até o trabalho ou estudar:

- 1- Acordar
- 2- Tomar banho
- 3- Escovar os dentes
- 4- Colocar a roupa
- 5- Tomar café
- 6- Obter dados sobre trânsito ou condições de tempo
- 7- Sair
- 8- etc, etc, etc

Pois bem, os computadores precisam fazer o mesmo. Eles precisam de uma sequência lógica de passos para resolver um problema.

Como os computadores não entendem a linguagem humana, os algoritmos devem ser reformulados em uma linguagem de programação (como Python, Java, C ++, Perl, Ruby, Go ou dezenas de outras) para fornecer instruções específicas ao computador.

Uma boa estratégia ao iniciar o desenvolvimento de um programa é primeiro escrever o pseudocódigo que representa os passos que você pretende executar e só então converter seu pseudocódigo em uma linguagem de programação.

Para os iniciantes essa abordagem pode ser muito benéfica, até que a lógica por trás da programação esteja clara. Os desenvolvedores mais experientes em geral não usam mais o pseudocódigo e já começam direto na linguagem de programação desejada!

Mas aqui nós iremos praticar pseudocódigo com diferentes abordagens. Nas próximas aulas eu mostrarei o pseudocódigo de diversas operações básicas em programação, como condicionais e loops, mas agora gostaria de voltar à Sequência de Fibonacci e pensar sobre o problema de forma diferente.

Programação, em sua essência, é raciocínio lógico!

Na Aula 1, nós queríamos imprimir na tela o valor da Sequência de Fibonacci em uma posição específica. Agora nós queremos imprimir todos os valores da Sequência de Fibonacci até uma posição específica.

Poderíamos escrever o pseudocódigo da seguinte forma (não leia apenas o que está nas linhas a seguir. Leia e visualize o problema, ou melhor ainda, crie um fluxo em um pedaço de papel. Isso ajudará a destravar seu cérebro para o pensamento lógico).

Obs: Vou abreviar a Sequência de Fibonacci para SF.

Problema: Imprimir todos os valores da Sequência de Fibonacci até uma posição específica dada.

Pseudocódigo:

- 1- Defina a variável n como sendo a posição desejada na SF.
- 2- Defina a variável n1 para o valor da primeira posição na SF.
- 3- Defina a variável n2 para o valor da segunda posição na SF.
- 4- Defina e inicialize a variável count para servir como contador, a fim de saber quando chegamos ao fim da SF até o valor de n.
- 5- Se o valor de n for menor que zero, imprima mensagem na tela indicando que n não pode ser negativo, pois não temos posição negativa na SF.
- 6- Senão, se n for igual a 1, imprime n1.
- 7- Senão, repetimos os passos abaixo enquanto count for menor que n:
 - 7.1. Imprime n1 na tela.
 - 7.2. Soma n1 com n2 para obter o próximo valor.
 - 7.3. A variável n1 recebe o valor da variável n2.
 - 7.4. A variável n2 recebe o valor calculado em 7.2.
 - 7.5. Incrementa o contador.

Veja que o pseudocódigo ficou um pouco mais complexo do que fizemos na Aula 1 (pois o problema agora é mais complexo), mas com pseudocódigo podemos raciocinar sobre a lógica a ser usada. ISSO É PROGRAMAÇÃO. E é exatamente isso que você deve praticar sempre que se deparar com um problema em Ciência de Dados (teremos problemas bem divertidos pela frente).

Quando estiver satisfeito com seu pseudocódigo, traduza as ações para sua linguagem de programação preferida (em nosso caso, Python). Aqui está o código em Python correspondente ao pseudocódigo anterior (não usaremos função nesse exemplo e recomendo que você salve o código em um arquivo texto com extensão .py e depois execute via terminal usando

python nome_arquivo.py.

Os traços representam a indentação e devem ser substituídos por espaços ou tabs):

Programa para exibir a Sequência de Fibonacci até o enésimo termo, onde n é fornecido pelo usuário.

1. Defina a variável n como sendo a posição desejada na SF.
n = 10

2. Defina a variável n1 para o valor da primeira posição na SF.
n1 = 0

3. Defina a variável n2 para o valor da segunda posição na SF.
n2 = 1

4. Defina e inicialize a variável count para servir como contador, a fim de saber quando chegamos ao fim da SF até o valor de n.
count = 0

5. Se o valor de n for menor que zero, imprima mensagem na tela indicando que n não pode ser negativo, pois não temos posição negativa na SF.
if n < 0:
----print("A posição não pode ser negativa")

6. Senão, se n for igual a 1, imprime n1.
elif n == 1:
----print("Sequência de Fibonacci até ", n, ":")
----print(n1)

7. Senão, repetimos os passos abaixo enquanto count for menor que n:
else:
----print("Sequência de Fibonacci até", n, ":")
----while count < n:

-----# 7.1. Imprime n1 na tela.
-----print(n1, end = ' , ')

-----# 7.2. Soma n1 com n2 para obter o próximo valor.

```
-----nth = n1 + n2
```

```
-----# 7.3. A variável n1 recebe o valor da variável n2.
```

```
-----n1 = n2
```

```
-----# 7.4. A variável n2 recebe o valor calculado em 7.2
```

```
-----n2 = nth
```

```
-----# 7.5. Incrementa o contador.
```

```
-----count += 1
```

Ao executar o programa acima, a saída será:

Sequência de Fibonacci até 10:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34

Altere o valor de n e teste seu programa com diferentes valores.

Observe, mais uma vez, que usamos instruções básicas em Python. Nosso maior trabalho foi definir a sequência de operações, ou seja, a lógica do problema.

Agora eu gostaria que você fizesse um exercício. Não precisa compartilhar aqui, isso é apenas para sua autoanálise e reflexão.

- Dos itens acima, em qual você sentiu mais dificuldade?

Dependendo da sua resposta:

1- Se a dificuldade foi em Python, fazer o curso gratuito de Python Fundamentos para aprender as instruções das linguagens.

2- Senão, se a dificuldade foi no pseudocódigo, praticar mais a definição de problemas. Pseudocódigo é apenas língua portuguesa para especificar uma série de operações.

3- Senão, se a dificuldade foi em formular os passos necessários para gerar o pseudocódigo, pesquise e estude sobre a Sequência de Fibonacci, para compreender mais sobre a área de negócio na qual você quer resolver o problema (imprimir todos os valores até uma posição dada).

Ops, acabei usando pseudocódigo para resolver problema de pseudocódigo. 😂

Talvez meu código interno esteja em loop! 🤖

Até a próxima aula! #aula02