

# Linguagem de Programação



## 01: LÓGICA E ALGORITMOS

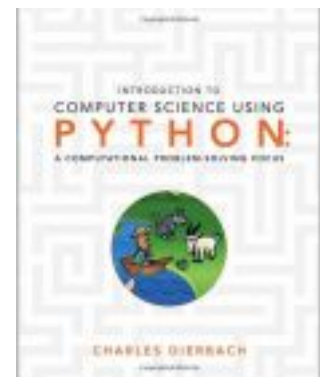
Nossos **objetivos** nesta aula são:

- Compreender a origem e o significado do termo **algoritmo**
- Aplicar este conceito para **resolver** alguns problemas simples
- entender a **relação** de **algoritmos** com **programas de computadores**
- implementar algoritmos em **Python**



A referência para esta aula são as sessões 1.1 (O que é Ciência da Computação?), 1.2 (Algoritmos) e 1.5 (O processo para resolver um problema computacional) do **Capítulo 1** do livro:

DIERBACH, C. *Introduction to Computer Science Using Python: A Computational Problem Solving Focus*. 1st Edition, New York: Wiley, 2012.



## MOTIVAÇÃO

A **tecnologia computacional** tem mudado e está continuamente mudando a vida das pessoas. Essencialmente todos os aspectos da vida têm sido impactados pela computação.

- Áreas relacionadas com computação estão emergindo em praticamente em todas as áreas de conhecimento: Biologia, Química, Medicina, Jornalismo, Arquitetura, Música, Fotografia, Matemática, Ciências Sociais, Manufatura auxiliada por computador, inteligência computacional, têxtil, economia.
- O campo de **ciência de computação** é fundamentalmente resolver problemas computacionais e linguagens de programação e computadores são ferramentas, porém não transmite a sua extensão e diversidade, como as suas várias áreas de estudo:
- Engenharia de Software (incluindo o projeto e implementação de sistemas de software), Gerenciamento de Banco de Dados, Redes de Computadores, Computação Gráfica, Segurança da Informação, Arquitetura de computadores, Interação Computador e Homem, Robótica, Inteligência Artificial, dentre outras.
- A definição de ciência da computação como resolução de problemas computacionais inicia com a definição de algoritmo.



# Linguagem de Programação



## 01: LÓGICA E ALGORITMOS

### ALGORITMO

O termo **algoritmo** pode ser visto desde o **século IX**. Foi nesta época que o cientista, astrônomo e matemático persa *Abū 'Abd Allāh Muhammad ibn Mūsā al-Khwārizmī* usou pela primeira vez o termo para indicar **regras** de operações aritméticas utilizando algarismos indoarábicos.

No século XII, Adelardo de Bath traduziu o termo para o latim **Algorithmi**. De lá para cá, o termo evoluiu bastante incluindo todos os **procedimentos definidos para resolver problemas** ou **realizar tarefas**.

A formalização da noção de algoritmo ocorreu em 1936 com os trabalhos de **Alan Turing** e **Alonzo Church**, que desenvolveram independentemente os modelos de **Máquinas de Turing** e **Cálculo Lambda**.

Do ponto de vista computacional, um **algoritmo** pode ser visto como um **conjunto de regras e procedimentos lógicos perfeitamente definidos que levam à solução de um problema em um número finito de passos**.

*Donald Knuth*, um dos pesquisadores mais respeitados em algoritmos, indica uma lista de cinco propriedades que são **requisitos para algoritmos**:

- **Finitude**: um algoritmo deve sempre terminar após um número finito de etapas (ou passos).
- **Definição**: cada passo de um algoritmo deve ser definido com precisão. As ações a serem executadas deverão ser especificadas rigorosamente e sem ambiguidades.
- **Entrada**: valores que são dados ao algoritmo antes que ele inicie.
- **Saída**: os valores resultantes das ações do algoritmo a partir de uma determinada entrada.
- **Eficácia**: todas as operações a serem realizadas pelo algoritmo devem ser suficientemente básicas para poderem, em princípio, ser feitas com precisão e em um período de tempo finito por um homem usando papel e lápis.



Embora os requisitos de Knuth sejam intuitivos, falta-lhes **rigor formal**. A formalidade pode ser conseguida com o uso de **lógica**. Assim, vamos exigir que um algoritmo seja **uma sequência lógica de passos com começo, meio e fim**.

Comumente, esta lógica é conhecida como **lógica de programação** e isto ocupará grande parte de nossa disciplina.

# Linguagem de Programação



## 01: LÓGICA E ALGORITMOS

De um ponto de vista mais geral, podemos dizer que **Lógica** é uma parte da **Filosofia** que trata das **formas do pensamento em geral (dedução, indução, hipótese, inferência, dentre outros)** e das **operações intelectuais que visam à determinação do que é verdadeiro ou falso**. Dentre outras coisas, a Lógica pode produzir algoritmos para estas formas e operações.

A fim de resolver um problema **computacionalmente**, duas coisas são necessárias: uma **representação** que capta todos os aspectos relevantes do problema, e um **algoritmo** que resolve o problema pelo uso da representação.



Vamos considerar o problema conhecido como o homem, o repolho, o bode e o lobo:

- Um homem vive no lado leste de um rio. Ele deseja trazer um repolho, um bode e um lobo a uma vila, no lado oeste do rio, para vender. Contudo, seu barco é grande o suficiente somente para ele e, ou o repolho, ou o bode, ou o lobo. Além disso, o homem não pode deixar o bode sozinho com o repolho porque o bode vai comer o repolho, e ele não pode deixar o lobo sozinho com o bode porque o lobo vai comer o bode. Como o homem pode resolver seu problema?

Há uma forma algorítmica fácil para resolver este problema com simplicidade, tentando todas as combinações possíveis de itens que podem ser levados e trazidos de uma margem à outra do rio. Tentar todas as soluções possíveis para um determinado problema é referido como “abordagem de força bruta”.

O que seria uma representação adequada para este problema? Uma vez que apenas os aspectos relevantes do problema precisam ser representados, todos os detalhes irrelevantes podem ser omitidos. A representação que deixa de fora detalhes do que está sendo representado é uma forma de abstração.

O uso de abstração é prevalente em computação. Neste caso, a cor do barco é relevante? A largura do rio? O nome do homem? Não, a única informação relevante é onde cada item está a cada passo. A localização coletiva de cada item, neste caso, refere-se ao estado do problema. Assim, o estado inicial do problema pode ser representado como se segue:

**homem repolho bode lobo**

0 0 0 0

# Linguagem de Programação



## 01: LÓGICA E ALGORITMOS

Nesta representação, o número **0** indica que cada objeto correspondente está no lado leste do Rio. Se o homem fosse atravessar o bode com ele, por exemplo, então a representação de o novo estado de problema seria:

**homem repolho bode lobo**

1 0 1 0

Em que o número **1** indica que o objeto correspondente está no lado oeste do rio. Neste caso, o homem e o bode. (A localização do repolho e do lobo são deixadas inalteradas). Uma solução para este problema é uma sequência de passos que converte o estado inicial:

0 0 0 0

Em que todos os objetos estão no lado leste do rio, para o estado final:

1 1 1 1

Em que todos os objetos estão no lado oeste do rio. Cada etapa corresponde ao homem atravessando o rio com um objeto em particular (ou o homem atravessa sozinho).

### Para praticar

- 1) A partir do estado inicial, com todos os elementos na margem leste do rio, escreva uma sequência de passos que leve todos os elementos para a margem oeste do rio, levando em consideração tudo o que foi descrito anteriormente.
- 2) Escreva uma sequência de passos para obter dois números, calcular e apresentar a soma dos dois números.

## O PROCESSO PARA RESOLVER PROBLEMAS COMPUTACIONAIS

A resolução de um problema computacional não envolve simplesmente o ato de programar um computador. Ele é um **processo**, sendo que a programação é apenas um dos passos. Antes de escrever um programa é preciso entender e pensar a solução do problema, depois desenvolver o projeto, escrever e testar a solução encontrada.

Uma vez que o algoritmo tenha sido definido, podemos **simulá-lo** com ferramentas como Scratch, VisuAlg ou Raptor, representá-los em **Fluxograma** ou **implementá-lo** diretamente em alguma **linguagem de programação** (Python, C, C++, Java, Pascal, PHP, dentre outras).

# Linguagem de Programação



## 01: LÓGICA E ALGORITMOS

Para a maior parte dos problemas, a linguagem de programação é o que menos importa. Se a lógica do algoritmo for bem feita, podemos implementá-lo na maioria das linguagens de programação conhecidas. Nossa disciplina será focada na linguagem **Python**.

### Uma pitada de Python...

A linguagem Python foi escolhida por ser uma linguagem muito versátil, usada não só no desenvolvimento Web, mas em muitos outros tipos de aplicações. Embora simples, é também uma linguagem poderosa, podendo ser usada para administrar sistemas e desenvolver grandes projetos. É uma linguagem clara e objetiva, pois vai direto ao ponto, sem rodeios. A página oficial do Python é <https://www.python.org/>, e na página da APyB - Associação Python Brasil (<https://python.org.br/empresas/>) é possível ver uma lista de estados, cidades e empresas que usam Python no Brasil. Python é uma linguagem que conta com uma comunidade muito forte e ativa em todo o mundo, então o leitor deste material estará aprendendo a programar em uma linguagem que poderá utilizar na prática e em qualquer lugar, estando longe de ficar apenas na teoria.

O nome Python é uma homenagem ao grupo humorístico inglês **Monty Python** criadores e intérpretes da série cômica Monty Python's Flying Circus (sucesso na década de 1970).

Apesar de sua sintaxe simples e clara, Python é uma linguagem multiparadigma que oferece muitos recursos disponíveis também em linguagens mais complexas como Java e C++, como por exemplo: programação orientada a objetos, recursos avançados de manipulação de arquivos, textos, listas, e outras estruturas de dados. Por ser uma linguagem interpretada, podemos ainda executar o mesmo programa sem modificações em várias plataformas de hardware e sistemas operacionais.

Python é software livre, ou seja, pode ser utilizado gratuitamente, graças ao trabalho da Python Foundation (<http://www.python.org/>) e de inúmeros colaboradores. Python pode ser utilizada em praticamente qualquer arquitetura de computadores ou sistema operacional, como Linux, Microsoft Windows ou Mac OS X.

Vamos começar com um primeiro exemplo:

```
>>> print("Olá mundo!")  
Olá mundo!
```

Temos uma linha de código onde a função *print* é usada para exibir uma mensagem de texto na tela. Podemos usá-la também para exibir o valor de uma variável:

5

# Linguagem de Programação



## 01: LÓGICA E ALGORITMOS

```
>>> a = 10
>>> print(a)
10
```

Neste caso, a variável *a* recebe o valor 10 e em seguida o conteúdo de *a* é exibido na tela. Podemos também pedir uma informação ao usuário. Para isso usamos a função *input*:

```
>>> nome = input()
|
```

Ao usarmos a função *input*, o Python ficará esperando que seja digitado algo e só continuará a execução após apertarmos o botão “Enter”. Neste caso guardando o dado digitado na variável *nome*.

### Para praticar

- 1) Escreva um programa em Python que calcula e apresenta a soma de dois números.
- 2) Escreva um programa em Python que calcula e apresenta o dobro, o triplo e o quadrado de um número dado.

### LISTA DE EXERCÍCIOS

- 1) Faça um programa que peça as 4 melhores notas de atividades contínuas, calcule e mostre a média.
- 2) Faça um programa que converta metros para milímetros.
- 3) Faça um programa que peça o raio de um círculo, calcule e mostre sua área.
- 4) Faça um programa que peça o valor do lado de um quadrado, calcule a área e em seguida exiba o dobro deste valor para o usuário.

- 5) Faça um programa que pergunte a diária do aluguel de um carro e quantos dias irá durar a locação, calcule o valor final do aluguel e exiba o resultado na tela.