Introdução a Estruturas de dados

MSc Sandir R Campos

ED - Importância

- As estruturas de dados, junto com o estudo de algoritmos, fazem parte dos fundamentos da programação e muito se lê/ouve sobre a importância do estudo destes temas. Nesta disciplina, vamos abordar as estruturas de dados do início: o que são, alguns exemplos e por que são importantes.

O que são dados?

- Os dados (e seus diversos tipos) são os blocos básicos da programação. Eles representam uma unidade ou um elemento de informação que pode ser acessado através de um identificador - por exemplo, uma variável.

Variáveis

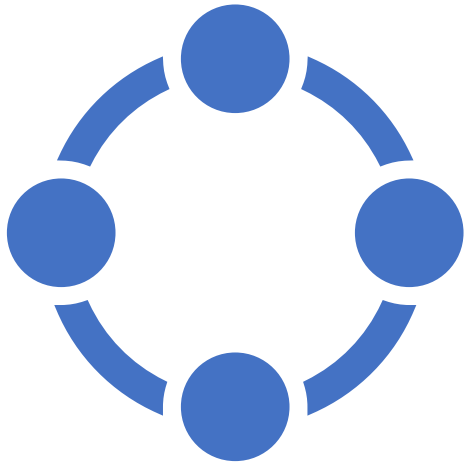
A maior parte das linguagens de programação trabalha com variações baseadas nos quatro tipos primitivos abaixo:

- INT ou número inteiro: valores numéricos inteiros (positivos ou negativos);
- FLOAT ou o chamado “ponto flutuante”: valores numéricos com casas após a vírgula (positivos ou negativos);
- BOOLEAN ou booleanos: representado apenas por dois valores, “verdadeiro” e “falso”. Também chamados de operadores lógicos;
- TEXT: sequências ou cadeias de caracteres, utilizados para manipular textos e/ou outros tipos de dados não numéricos ou booleanos, como hashes de criptografia.

O JavaScript

- tem como tipos primitivos embutidos na estrutura básica da linguagem: number, string, boolean e symbol (de “nome simbólico”, usado entre outras coisas para criar propriedades únicas em objetos).

C# - (C-Sharp)



- trabalha com uma quantidade maior de tipos primitivos, de acordo com o espaço de memória que será ocupado pela variável: Boolean, Byte, SByte, Int16, UInt16, Int32, UInt32, Int64, UInt64, IntPtr, UIntPtr, Char, Double e Single.

Linguagem C++

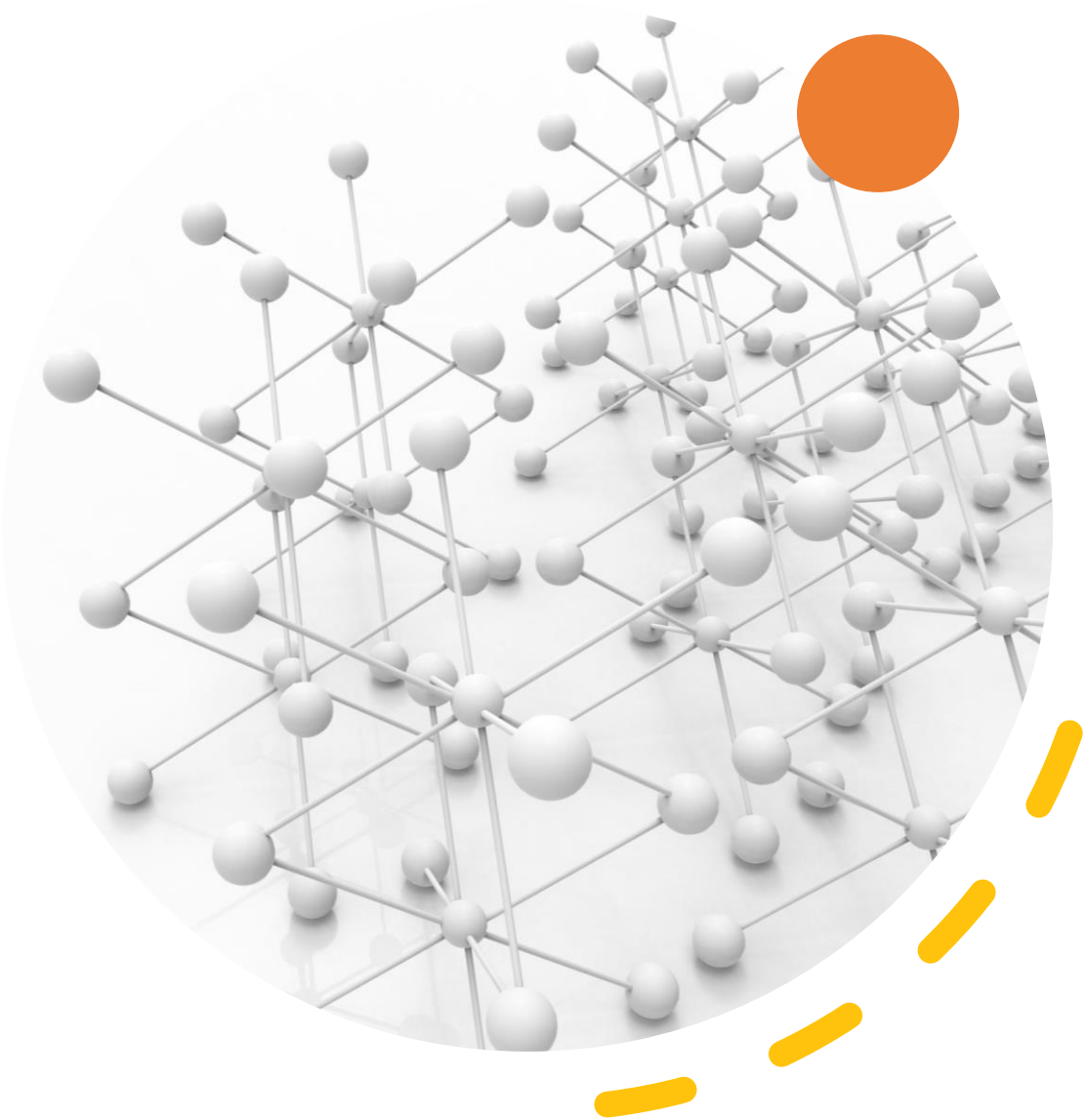
não tem um tipo próprio de dado booleano; false é representado pelo número 0 e qualquer outro algarismo representa true. Outras linguagens podem trabalhar com outras variações.

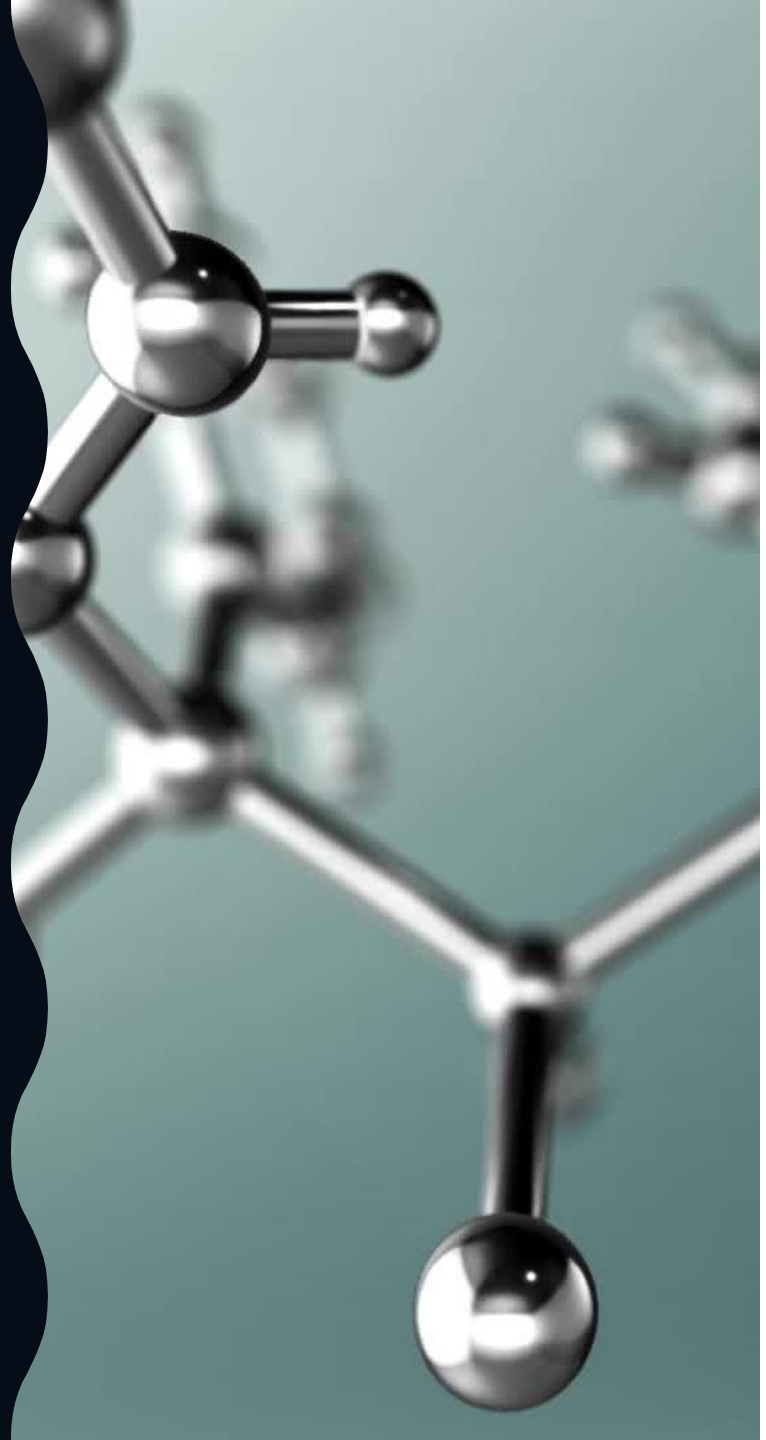
Pesquisa 1

- Quais são os tipos de dados em Python e em JAVAScript?

O que são estruturas de dados?

- Em computação, normalmente utilizamos os dados de forma conjunta.
- A forma como estes dados serão agregados e organizados depende muito de como serão utilizados e processados, levando-se em consideração, por exemplo, a eficiência para buscas, o volume dos dados trabalhados, a complexidade da implementação e a forma como os dados se relacionam.
- Estas diversas formas de organização são as chamadas estruturas de dados.



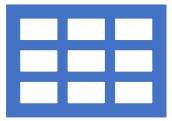


Programa

- **Podemos afirmar que um programa é composto de algoritmos e estruturas de dados, que juntos fazem com que o programa funcione como deve**

Métodos

- Cada estrutura de dados tem um conjunto de métodos próprios para realizar operações como:
  - Inserir ou excluir elementos;
  - Buscar e localizar elementos;
  - Ordenar (classificar) elementos de acordo com alguma ordem especificada.



Características das estruturas de dados

- As estruturas de dados podem ser:
  - lineares (ex. arrays) ou não lineares (ex. grafos);
  - homogêneas (todos os dados que compõe a estrutura são do mesmo tipo) ou heterogêneas (podem conter dados de vários tipos);
  - estáticas (têm tamanho/capacidade de memória fixa) ou dinâmicas (podem expandir).

Array

- Também chamado de vetor, matriz ou arranjo, o array é a mais comum das estruturas de dados e normalmente é a primeira que estudamos.
- Um array é uma lista ordenada de valores:

```
const listaNumeros = [4, 6, 2, 77, 1, 0];
const listaFrutas = ["banana", "maçã", "pera"];
```

Lista Ordenada

- Por ordenada, entenda-se aqui uma lista onde os valores sempre são acessados na mesma ordem. Ou seja, a não ser que seja utilizada alguma função ou método para alterar a ordem, o primeiro elemento do array listaNumeros sempre será 4, e o último, 0.

Criando um Array

```
var frutas = ['Maçã', 'Banana'];

console.log(frutas.length);

// 2
```

Fazer

Criar um Array com 50 elementos

Acessar um item (index) do Array

```
var primeiro = frutas[0];
// Maçã
var ultimo = frutas[frutas.length - 1];
// Banana
```

Fazer

Acessar os elementos 25 39 e 42

Iterar um Array

```
frutas.forEach(function (item, indice, array) {
console.log(item, indice);
});
// Maçã 0
// Banana 1
```

Fazer

Mostrar os elementos e índices da sua lista

Adicionar um item ao final do Array

```
var adicionar = frutas.push('Laranja');
```

```
// ['Maçã', 'Banana', 'Laranja']
```

Fazer

Adicionar 20 elementos a sua lista

Remover um item do final do Array

```
var ultimo = frutas.pop(); // remove Laranja (do final)
```

```
// ['Maçã', 'Banana'];
```

Fazer

Retirar 5 elementos de sua lista

Remover do início do Array

- var primeiro = frutas.shift(); // remove Maçã do início
- // ['Banana'];

Fazer

Retirar 3 primeiros elementos de sua lista

FIM DA AULA 1