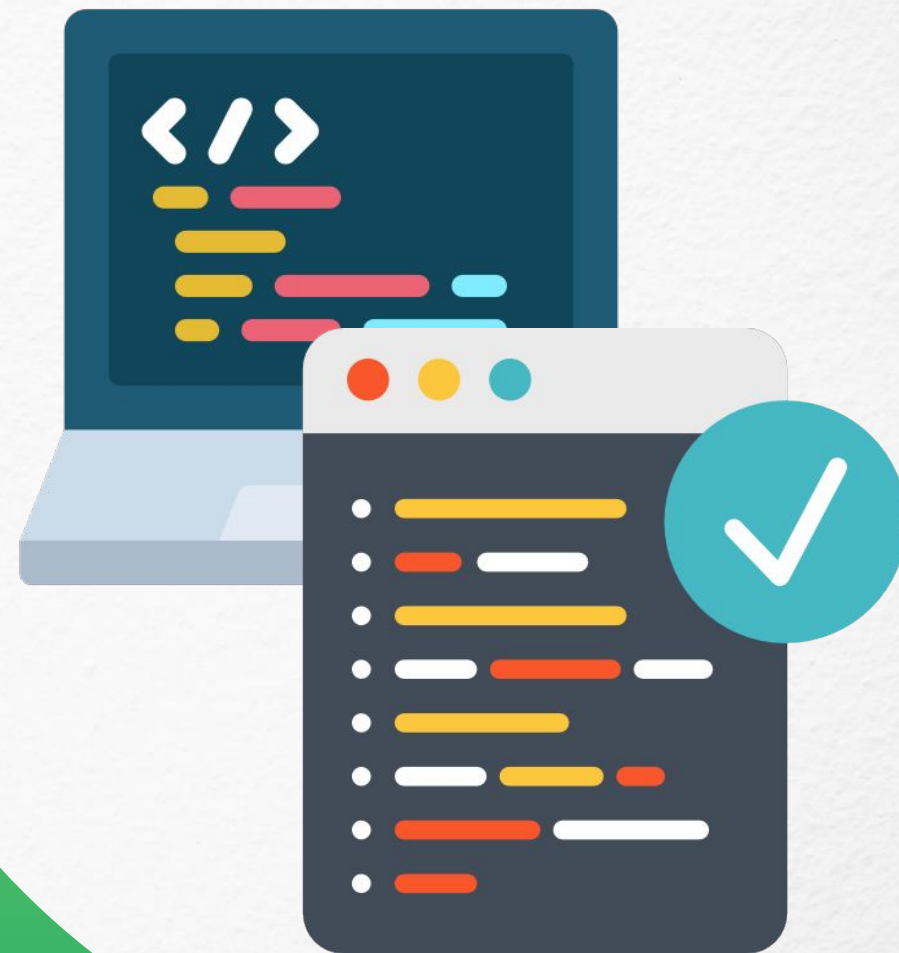




Escola Tecnológica
Desembargador Paulo Feitoza

Linguagens de Programação

Professora Luísa Braga



linguagens de programação

imagine que você precisa fazer um suéter de lã, mas você não sabe fazer isso.

você irá precisar **comunicar** uma costureira experiente como você espera que esse suéter fique e o que você deseja como resultado.

a costureira ouvirá as suas **instruções** e decidirá quais técnicas de costura ela irá utilizar para atender o seu pedido.

se você descreveu de forma que ela conseguiu entender, então o seu suéter será entregue exatamente como você imaginou.

linguagens de programação

se você fala apenas Português e procurou por uma costureira que só fala Alemão, provavelmente você não terá o resultado da forma que você queria, pois a costureira não será capaz de entender as suas instruções.

para que ela entenda, você deve passar todas as instruções para ela em Alemão, assim o suéter será entregue como o esperado.

linguagens de programação | definição

são ferramentas que permitem aos desenvolvedores escreverem **instruções** que um computador pode entender e executar.

elas são usadas para criar software, aplicativos, sites e muitos outros tipos de programas.

cada linguagem de programação tem sua própria sintaxe e conjunto de regras que os programadores seguem para escrever código.

linguagens de programação | exemplos

```
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello, World!");  
    }  
}
```

```
console.log("Hello, World!");
```

```
print("Hello, World!")
```

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {  
    cout << "Hello, World!" << endl;  
    return 0;  
}
```


linguagens de programação | usos

desenvolvimento de software: criar programas para diversas plataformas, como desktop, web, mobile e dispositivos embarcados.

análise de dados: manipular e interpretar grandes conjuntos de dados para obter insights e tomar decisões.

inteligência artificial: desenvolver sistemas que aprendem e tomam decisões autonomamente.

automação: automatizar tarefas manuais repetitivas para aumentar a produtividade.

linguagens de programação | tipos

uma das formas de definir os tipos de linguagem de programação é baseada na forma como a linguagem de alto nível é convertida em instruções para o computador, os tipos que temos são:

compiladas: o código é convertido em linguagem de máquina antes da execução. Ex: C, C++.

interpretadas: o código é lido e executado linha por linha, sem conversão prévia. Ex: Python, JavaScript.

linguagens de programação | exemplos

Python: linguagem geral, fácil de aprender, usada para diversas aplicações.

Java: linguagem robusta, usada para desenvolvimento de aplicações web e mobile.

C++: linguagem de alto desempenho, usada para jogos e sistemas embarcados.

JavaScript: linguagem usada para desenvolvimento web front-end.

C#: linguagem usada para desenvolvimento de aplicações Windows e web.

qual a melhor linguagem de programação?
depende

linguagens de programação | como escolher

curva de aprendizado

comunidade ativa

documentação

domínio da linguagem

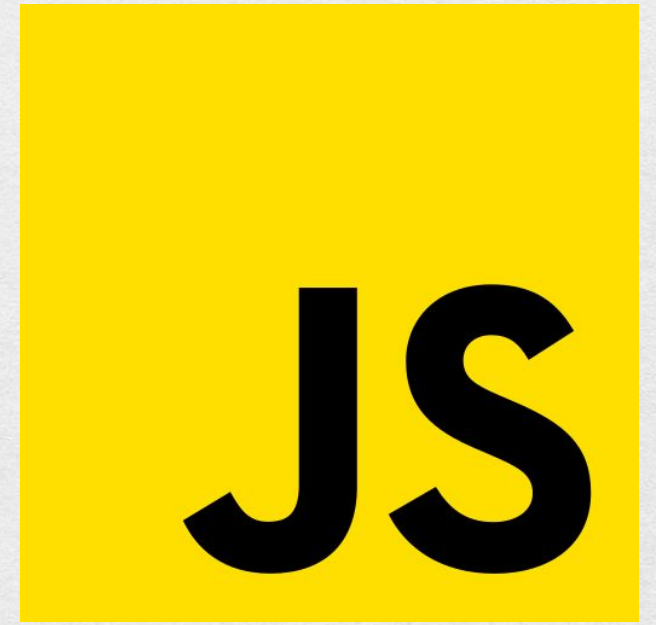
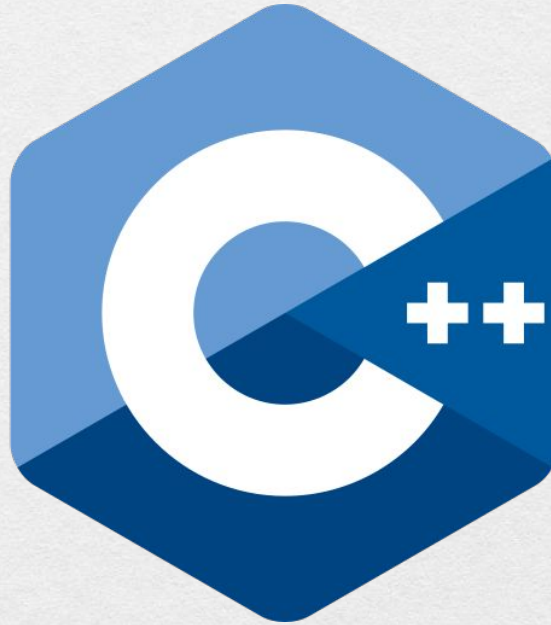
programador não pode ter linguagem de estimação

estar disposto a aprender várias linguagens faz parte da
magia de desenvolver software

you won't know everything about a language, it's impossible

researching and reading documentation is normal and expected

linguagens de programação



variáveis

são espaços na memória do computador que reservamos para armazenar os dados que desejarmos utilizar durante a execução do nosso programa, funcionam como gavetas

cada linguagem de programação tem uma forma distinta de declarar variáveis

no mundo do desenvolvimento de software, temos algumas convenções que precisamos seguir na hora de dar nome à variáveis nos nossos programas

variáveis

algumas regras comuns entre várias linguagens de programação ao nomear variáveis são:

- não podemos usar palavras reservadas nem nomes de funções para nomear uma variável
- não podemos utilizar acentos e outros caracteres especiais como -,.,@# e por aí vai
- não podemos utilizar espaços no nome de uma variável
- não podemos começar o nome de uma variável com um número

variáveis

em Python, utilizamos um padrão de escrita chamado **snake_case** onde separamos as palavras nos nomes das nossas variáveis por _

então, se eu tiver uma variável para representar o salário líquido e outra para o salário bruto, vou nomeá-las assim: **salario_liquido salario_bruto**

em JavaScript, utilizamos o padrão de escrita chamado **camelCase** onde utilizamos letra maiúscula no início de cada palavra, nesse padrão a primeira letra do nome da variável sempre é minúscula

as variáveis de salário representadas acima, ficariam assim em camelCase: **salarioLiquido salarioBruto**

em C++ as pessoas utilizam ambos os padrões, porém se você inicia um projeto em um padrão, não deve mudá-lo no meio do caminho.

variáveis

vamos ver como declarar e atribuir valores à variáveis nas três linguagens

nos nossos programas, quando **declaramos** uma variável estamos dizendo ao nosso computador: “por favor, reserve um espaço na memória pra mim, sempre que eu quiser acessá-lo, vou usar esse nome aqui”

quando queremos **atribuir** um valor a uma variável, estamos dizendo “neste espaço da memória, guarde esta informação aqui”

entrada de dados

podemos receber dados de entrada do usuário e armazená-los em variáveis para que possamos operar sobre esses valores nos nossos programas

```
a = int(input())
```

```
b = int(input())
```

```
soma = a + b
```



```
#include <iostream>
using namespace std;
int main() {
    int a, b, soma;

    cin >> a;
    cin >> b;

    soma = a + b;

    cout << "A soma de a + b é: " << soma << endl;
    return 0;
}
```



entrada de dados

podemos receber dados de entrada do usuário e armazená-los em variáveis para que possamos operar sobre esses valores nos nossos programas

```
const readline = require('readline').createInterface({
  input: process.stdin,
  output: process.stdout
});

readline.question('Digite o valor de a: ', (a) => {
  readline.question('Digite o valor de b: ', (b) => {
    let soma = parseInt(a) + parseInt(b);
    console.log(`A soma de a + b é: ${soma}`);
    readline.close();
  });
});
```

JS

tipos de dados

as variáveis que declaramos sempre possuem tipos, esses tipos são importantes para que o computador sempre saiba quanto de memória ele deve alocar para cada variável

alguns dos tipos que temos são:

- int
- float
- char
- boolean
- string
- double
- long int
- long double

tipos de dados

algumas linguagens como C++ possuem tipagem estática, o que quer dizer que as variáveis não podem mudar de tipos no decorrer da execução do programa

em linguagens como Python e JavaScript, temos a tipagem dinâmica, indicando que podemos mudar o tipo de uma variável no decorrer da execução dos meus programas

tipos de dados

também podemos fazer **casting** de tipos isso significa que podemos alterar um tipo de dado para ou, por exemplo, quando desejamos fazer alterações do tipo

transformar a string “45” no inteiro 45

transformar o número decimal 9621.32 na string “9621.32”

strings

são cadeias de caracteres, elas podem representar textos e palavras nos nossos programas

```
a = 'Alô, bom dia'
```



```
const a = "duas palavras"
```

JS

```
#include <iostream>
#include <string>

using namespace std;

int main() {
    string inputString = "uma
palavra";

    return 0;
}
```



tipos de dados

faça um programa que declare 5 variáveis:

- a primeira deve guardar o seu nome
- a segunda deve guardar sua idade
- a terceira deve guardar sua altura
- a quarta variável deve usar sua idade e o ano atual para determinar o ano que você nasceu
- a quinta variável deve guardar uma string contendo a frase “<nome> nasceu no ano de <quarta variavel>”

mostre na tela o valor guardado na quinta_variável

estruturas de controle - condicional

a estrutura de condicional serve para que nosso programa tome diferente caminhos dependendo de determinadas condições definidas com base nos valores das variáveis do nosso programa

```
#include <iostream>
using namespace std;
int main() {
    int a, b, soma;

    cin >> a;

    if (a > 50) {
        cout << "maior que 50" << endl;
    } else {
        cout << "menor ou igual a 50" << endl;
    }

    return 0;
}
```



```
a = int(input())

if a > 50:
    print('maior que 50')
else:
    print('menor ou igual a 50')
```



estruturas de controle - condicional

a estrutura de condicional serve para que nosso programa tome diferente caminhos dependendo de determinadas condições definidas com base nos valores das variáveis do nosso programa

```
const readline = require('readline').createInterface({
  input: process.stdin,
  output: process.stdout
});

readline.question('Digite um número: ', (input) => {
  const a = parseInt(input);

  if (a > 50) {
    console.log('maior que 50');
  } else {
    console.log('menor ou igual a 50');
  }

  readline.close();
});
```

JS

tipos de dados

faça um programa que leia uma frase do usuário e mostre na tela a quantidade de letras maiúsculas, a quantidade de letras minúsculas, a quantidade de números e a quantidade de espaços presentes na frase lida

estruturas de controle - repetição

funções

modulos

manipulação de arquivos

paradigmas de programação



Escola Tecnológica
Desembargador Paulo Feitoza

 | www.fpf-etech.com

 | www.fpftech.com

 | [/fpf](#)
 | [@escola.etech](#)
 | [@fpf.tech](#)
 | [@fpftech](#)