



Aprenda a programar

Dia 3 - Aprendendo a programar (parte 1)

O que veremos

Revisão

-

Git e Github

Fundamentos de Programação

-

Comandos de programação

Tipos de dados, variáveis e memória

Lógica de programação

-

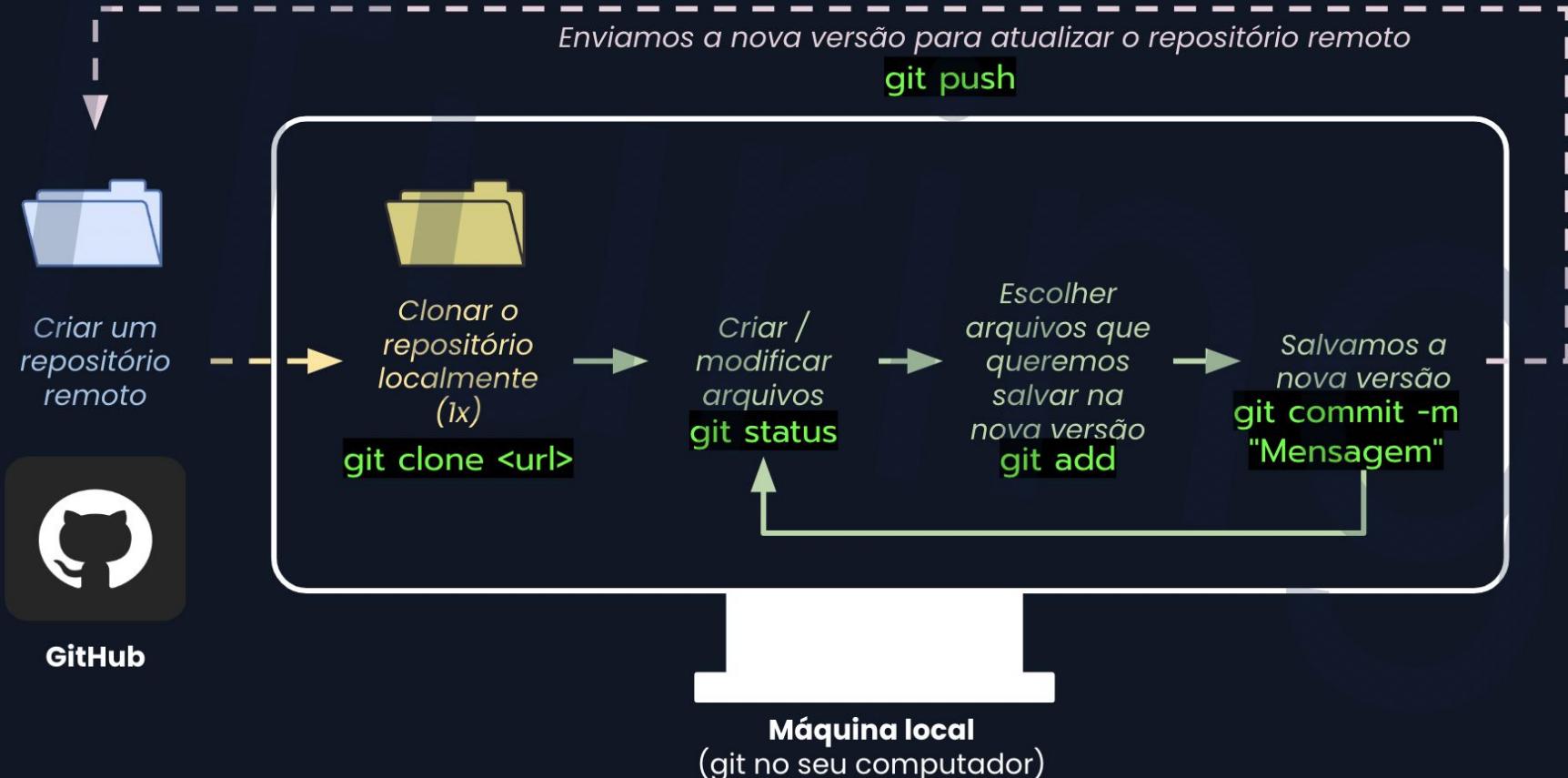
Operadores matemáticos

Estruturas condicionais e operadores relacionais:
a 'inteligência' por trás de tudo

Revisão git e GitHub

Criando um novo repositório...

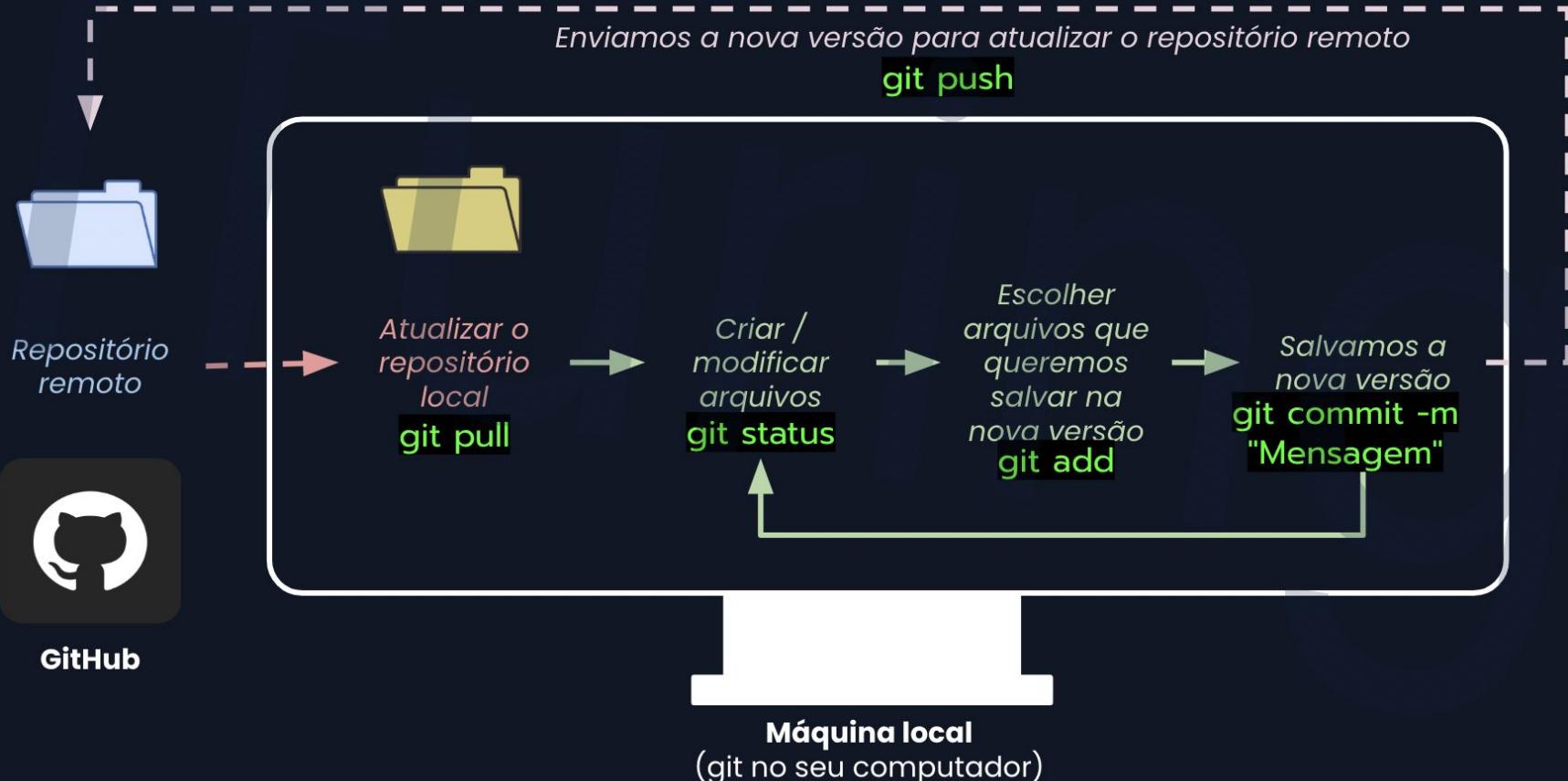
Revisão git e GitHub



Revisão git e GitHub

Mas e quando eu já criei o repositório no GitHub e também já tenho ele localmente ?

Revisão git e GitHub



Comandos de programação

Comandos de programação

O que faz um programa?



Informação entra

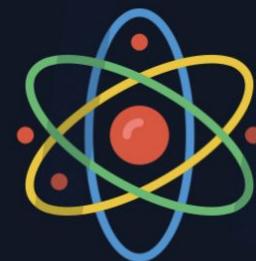
O programa
processa e faz algo

Devolve uma
informação
trabalhada

Comandos de programação

O que constitui um programa de computador?

Um programa de computador constitui de uma sequência de comandos que o computador sempre executa em ordem.



Comandos é a representação atômica
e elementar de uma ação
do computador.

TUDO EM PROGRAMAÇÃO SÃO COMANDOS!

Comandos de programação

O que constitui um programa de computador?

Um programa constitui de uma sequências de comandos que o computador sempre executa em ordem.

Os comandos **sempre** são executados pelo computador de cima para baixo.



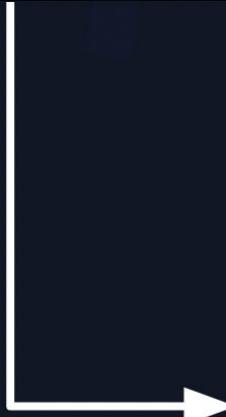
Da mesma forma que uma receita de culinária. Sempre executamos uma receita de cima para baixo.



Comandos de programação

Vamos escrever um programa? Vamos usar um comando do JavaScript.

console.log()

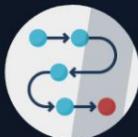


O console é um comando que exibe informações na tela.

Fundamentos de programação

Tipos de dados, variáveis e memória

Tipos de dados, variáveis e memória



Conforme vimos, o dado entra e viaja dentro do programa. Para isso ele precisa ficar **salvo em algum lugar**.



Esse lugar é a **memória** do programa. Mas precisamos ter uma forma de acessar esses **pedaços de memória** para usar o dado.



Esses espaços se chamam **variáveis**.

Tipos de dados, variáveis e memória

Uma **variável** é como se fosse uma caixinha criada na memória para que guardemos "coisas" dentro dela e possamos usar depois



Tipos de dados, variáveis e memória

Declarar uma variável é criar uma caixinha vazia com um nome específico (identificador/etiqueta)



Tipos de dados, variáveis e memória

A "coisa" que guardamos na caixa(variável)
damos o nome de **dado** ou **valor**



Tipos de dados, variáveis e memória

Atribuir um dado/valor a uma variável é o mesmo que guardar essa "coisa" na caixa



Tipos de dados, variáveis e memória

Podemos ter dados de diferentes **tipos**



Tipos de dados, variáveis e memória

As variáveis podem receber dados dos **tipos**:

- **Strings (texto)** - "Instrutora Jô"
- **Number (número)** - 12
- **Booleans** - podem ser true ou false
- **Arrays (Listas)** - Listas de qualquer tipo de variável (inclusive listas de listas)
- **Mapas** - pares de chave e valor*



* calma a gente vai ver juntos

Tipos de dados, variáveis e memória

Será que existe uma forma de saber o tipo do
dado que está dentro de uma variável? 

`typeof()`

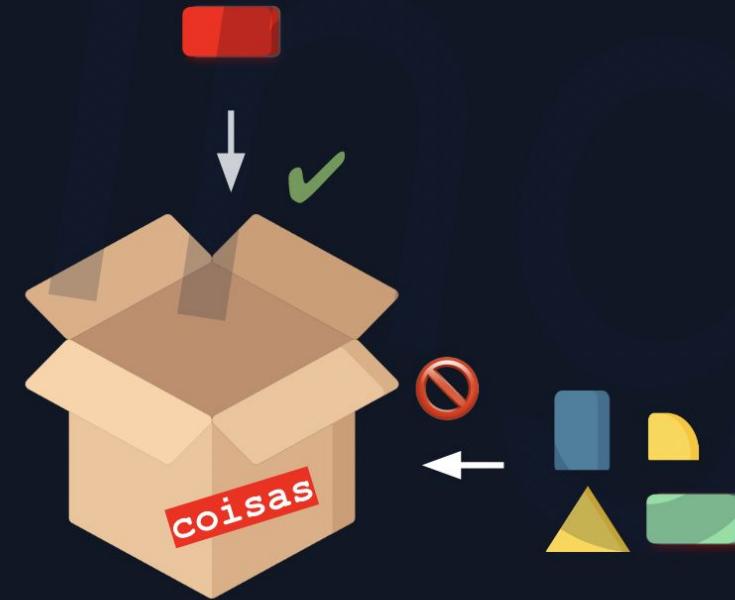
Tipos de dados, variáveis e memória

Tipagem dinâmica



Tipos de dados, variáveis e memória

Tipagem estática



Tipos de dados, variáveis e memória

Resumindo o que nós vimos até aqui:

Comando `console.log()`

Declaração de variáveis

Atribuição de valores

Comando `typeof()`

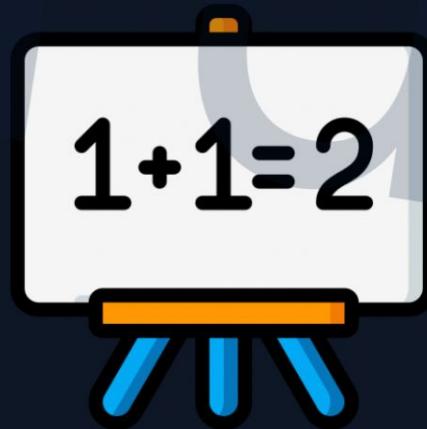
Tipagem estática e tipagem dinâmica

Lógica de programação

Operadores matemáticos (+, -, *, /)

Operadores matemáticos (+, -, *, /)

É aceita qualquer uma das operações básicas



- + soma
- subtração
- * multiplicação
- / divisão

Operadores matemáticos (+, -, *, /)

Vamos aproveitar o que já sabemos para entender como programas fazem contas:

- **Podemos somar números:** `console.log(5 + 5)`
- **Podemos somar variáveis:** `console.log(a + b)`
- **Podemos salvar resultados em variáveis:** `a = b + c`
- E por causa da tipagem dinâmica que vimos, podemos 'somar' textos

Operadores matemáticos (+, -, *, /)

Desafio!

Vamos tirar 5 minutos para pensar...
(rapidinho no VSCode)



Operadores matemáticos (+, -, *, /)

Desafio!

Vamos tirar 5 minutos para pensar...
(rapidinho no VSCode)

A operação "11" + 1 tem o resultado "111"

A operação "11" - 1 tem o resultado 10

Por que isso ocorre?

Lógica de programação

**Estruturas condicionais e
operadores relacionais:
a ‘inteligência’ por trás de tudo**

Condições são a “inteligência” do computador!



Não é inteligência artificial... isso é papo para outro dia.

Se (condição)
é verdade?

Faz uma coisa

sim

Faz outra coisa

não

Se (`idade > 18`)
é verdade?

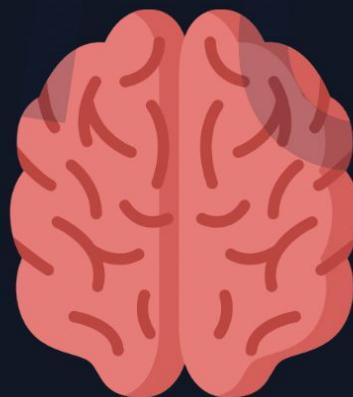
Pode jogar

sim

Não pode jogar

não

Mas como faz no código?



'Toda' linguagem de programação possui:

Declaração de variáveis

```
let variavel
```

Operadores aritméticos

```
+ SOMA
```

```
- SUBTRAÇÃO
```

```
* MULTIPLICAÇÃO
```

```
/ DIVISÃO
```

Comando de atribuição

```
variavel = 1
```

Operadores relacionais

```
> MAIOR QUE
```

```
< MENOR QUE
```

```
== IGUALDADE
```

```
>= MAIOR OU IGUAL
```

```
<= MENOR OU IGUAL
```

'Toda' linguagem de programação possui:

Comandos condicionais

```
if /* expressão */ {  
    // caso verdadeiro  
}
```

```
if /* expressão */ {  
    // caso verdadeiro  
} else {  
    // caso falso  
}
```

Palavras/símbolos reservadas

```
if  
else  
let  
+  
-  
/  
*  
(  
)  
{  
}
```

...

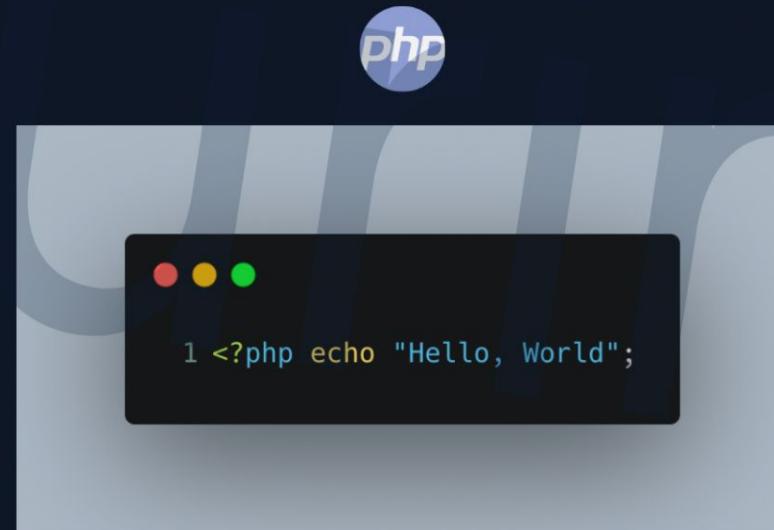
Mas cada linguagem possui **sua sintaxe**



Mas cada linguagem possui **sua sintaxe**



Mas cada linguagem possui **sua sintaxe**



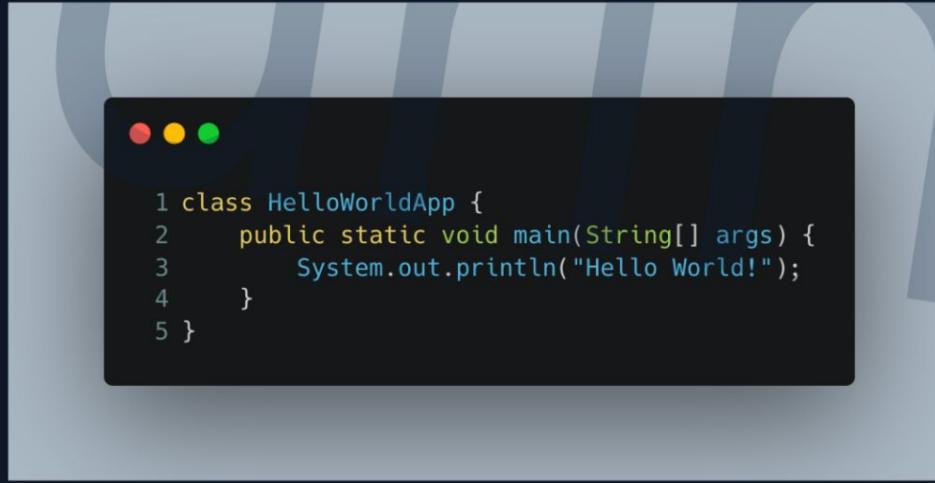
Mas cada linguagem possui **sua sintaxe**



A screenshot of a dark-themed code editor window. At the top, there are three circular window control buttons (red, yellow, green). Below them, the code for a 'Hello, world!' program is displayed:

```
1 using System;
2
3 class Program
4 {
5     static void Main(string[] args)
6     {
7         Console.WriteLine("Hello, world!");
8     }
9 }
```

Mas cada linguagem possui **sua sintaxe**



```
1 class HelloWorldApp {  
2     public static void main(String[] args) {  
3         System.out.println("Hello World!");  
4     }  
5 }
```

Mas cada linguagem possui **sua sintaxe**



```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     printf("hello, world\n");
6 }
```

Tarefa de hoje! Vamos olhar
juntos rapidinho