



Aprenda a programar

Dia 2 – Aprendendo a programar e
adquirindo o hábito com GitHub, parte 1

O que veremos

Introdução ao git e ao GitHub
Escrevendo o primeiro código e fazendo um commit
Subindo o código para o 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

Relembrando...

Preparando a caixa de ferramentas de programação



O que as pessoas geralmente utilizam no computador para escrever código e programar computadores?

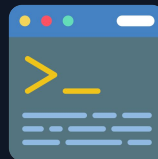


Editor de texto

(não é Microsoft Word e nem Google Docs!!!)



Um sistema de controle de versões para ver o histórico de tudo que você está codificando



Um terminal para se comunicar com o sistema operacional e executar o programa que você codificou

Preparando a caixa de ferramentas de programação



E dessas ferramentas, quais vamos utilizar aqui no curso?

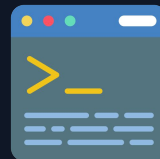


Editor de texto VSCODE

(não é Microsoft Word e nem Google Docs!!!)



Veja e controle o histórico do que você está codificando com o git



Um terminal para se comunicar com o sistema operacional e executar o programa que você codificou

Linguagens de programação* que serão utilizadas no curso:



JavaScript

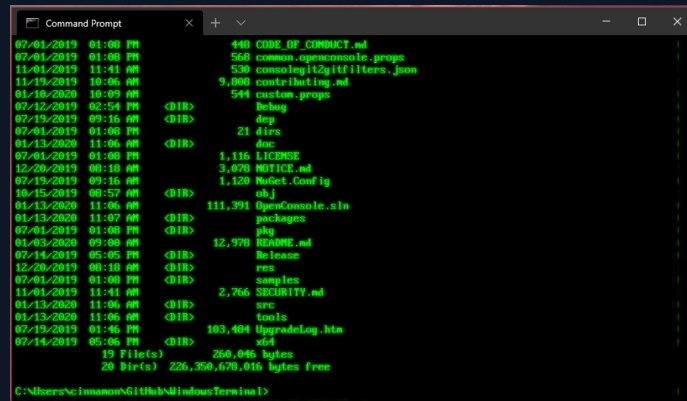


*html e css não são linguagens de programação. São linguagens com um outro propósito que vamos explicar mais pra frente

Introdução ao terminal

Principais comandos nativos do terminal:

- pwd - mostrar em que pasta estamos
- ls - listar tudo que está dentro de uma pasta
- cd - mudar de diretório
- mkdir - criar uma pasta/dirertório
- touch - criar um arquivo
- cat - mostrar todo o conteúdo do arquivo
- head - mostrar o que está no início do arquivo
- grep - pesquisar conteúdo em arquivos



```

Command Prompt
C:\Users\Ncinnamon6\Github\WindowsTerminal>ls
440 CODE_OF_CONDUCT.md
560 common.openconsole.props
530 consolelogit2gitfilters.json
9,000 contributing.md
544 custom.props
  debug
  dep
  dist
  docs
  1,116 LICENSE
  3,070 NOTICE.md
  1,120 NuGet.Config
  obj
  111,391 OpenConsole.sln
  packages
  pkg
  12,970 README.md
  Release
  src
  samples
  2,766 SECURITY.md
  src
  tools
  103,404 UpgradedLog.htm
  x64
19 File(s)      260,046 bytes
20 Dir(s)      226,350,670,016 bytes free
C:\Users\Ncinnamon6\Github\WindowsTerminal>

```

O poder do GitHub para a sua carreira em programação

Retomando... por que GitHub?



Para compartilhar repositórios (pastas) de código.



Um repositório (pasta) não contém somente o código, mas um histórico de tudo que ele já fez naquele código.



Ele dá visibilidade aos seus projetos, ao que você cria de código e deixa disponível lá. **Recrutadores do mundo inteiro estão sempre olhando o GitHub!**



Ele permite não só a visibilidade, mas também o trabalho em equipe, controlando as alterações no código e versões do projeto.

Introdução ao git e ao GitHub

Antes de tudo vamos instalar...

Repos Locais e Repos Remotos

Já estamos usando a palavra "repo" para você ir se acostumando. E claro, repo = repositório. Relembrando, é uma pasta em que está ocorrendo controle de versão pelo Git.

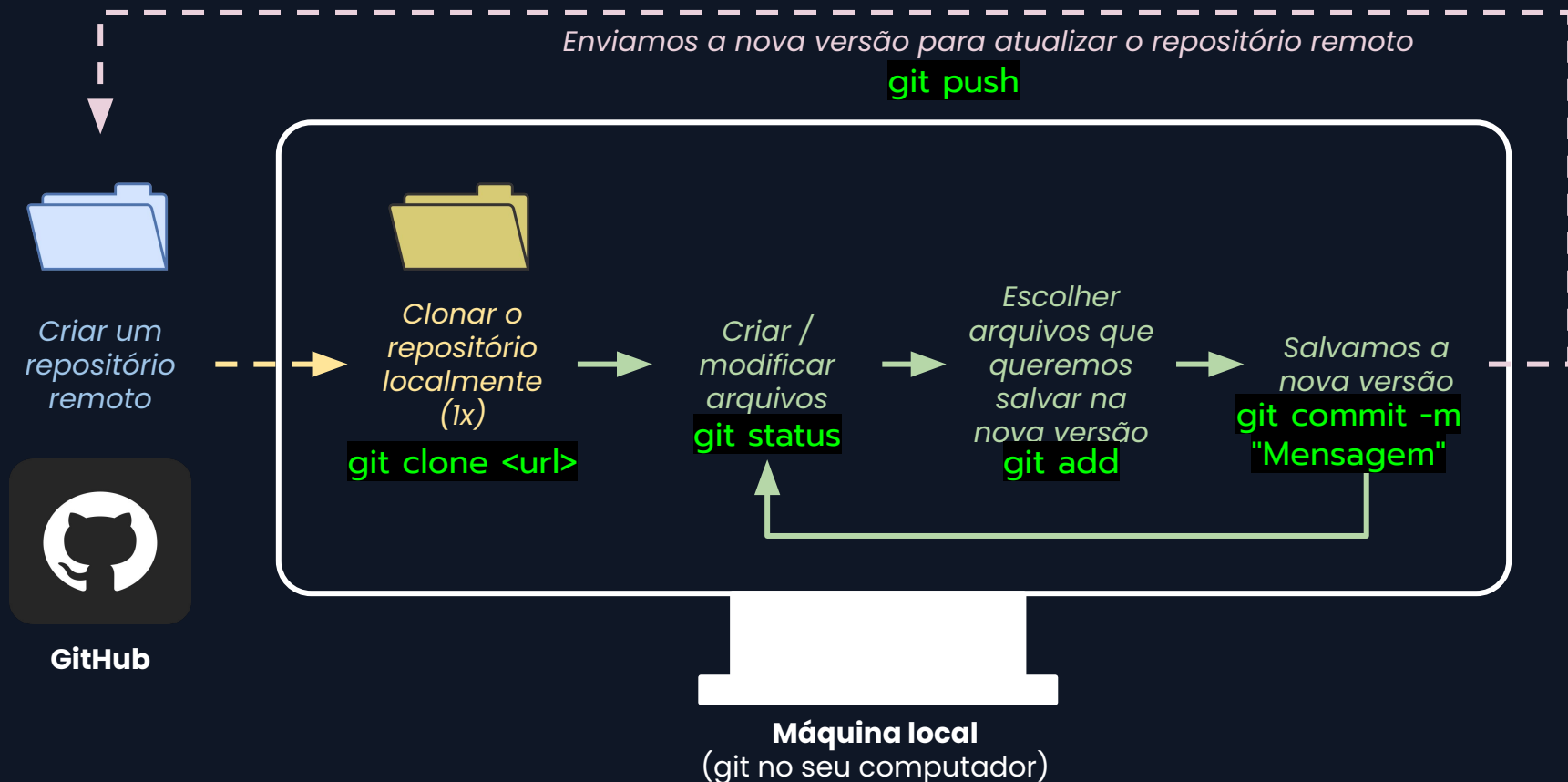


Repositório local fica na sua máquina. É onde você trabalha e faz suas alterações, implementa seu código.



Repositório remoto é o que fica no GitHub. Quando você termina de implementar algum código na sua máquina, você joga essas alterações para o remoto. Em um time de developers, os outros membros da equipe poderão ver as suas alterações. Você também pode pegar as alterações que o time fez no repositório remoto (no GitHub) e trazer para sua máquina (repositório local).

Introdução ao git e ao GitHub



Comandos de programação

Comandos de programação

O que faz um programa de computador?

Comandos de programação

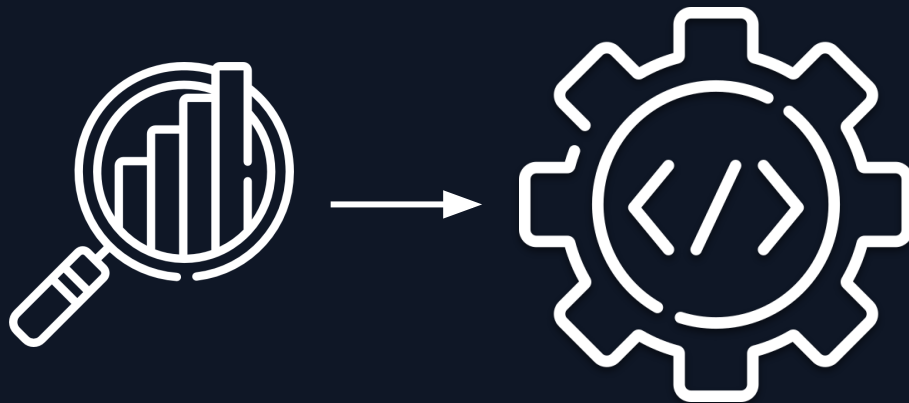
O que faz um programa de computador?



Informação entra

Comandos de programação

O que faz um programa de computador?



Informação entra

**O programa
processa e faz algo**

Comandos de programação

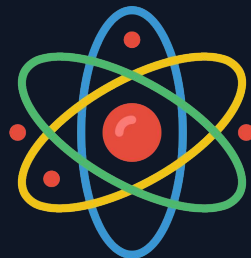
O que faz um programa de computador?



Comandos de programação

O que constitui um programa de computador?

Um programa de computador constitui de uma sequência de comandos (ou instruções) que o computador sempre executa em ordem.



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

TUDO EM PROGRAMAÇÃO SÃO COMANDOS (OU INSTRUÇÕES)!

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.

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

O que constitui um programa de computador?

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



Comando 1

Comando 2

Comando 3

Comando 4

Comando 5

Comando 6

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.

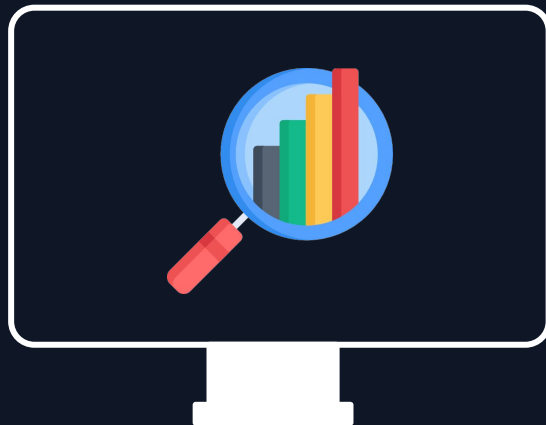


Comando 1	1	<code>let nome = "Maria"</code>
Comando 2	2	<code>let email = "maria@maria.com"</code>
Comando 3	3	<code>let profissao = "publicitária"</code>
Comando 4	4	<code>let idade = 33</code>
Comando 5	5	<code>let viagemDosSonhos = "Bali"</code>
Comando 6	6	
Comando 7	7	<code>let mensagem = "Olá, " + nome + "! Somos aqui da agência de turismo Viagem dos Sonhos. Estamos te escrevendo este email, pois acabamos de confirmar as compras das passagens aéreas para " + viagemDosSonhos + ". Confirma pra gente alguns dados? A sua idade é " + idade + ", você é " + profissao + " e o seu email é " + email + "? Ficamos no aguardo. Muito obrigado e boa viagem! :D"</code>
Comando 8	8	
Comando 9	9	<code>console.log(mensagem)</code>

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.

Comandos de programação

Bora praticar um pouco o `console.log()`!

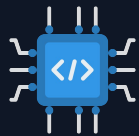
Dúvidas?

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/Objetos** - pares de chave e valor*

Hoje só trabalharemos com estes tipos de dados



* calma a gente vai ver juntos

Tipos de dados, variáveis e memória

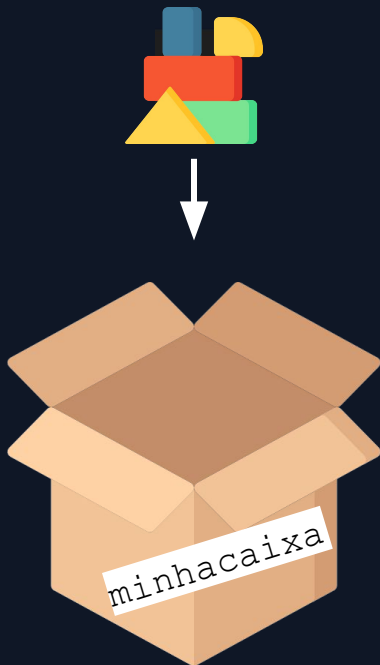
Declarar uma variável é criar uma caixinha vazia com um nome específico (identificador/etiqueta)



```
let minhacaixa
```

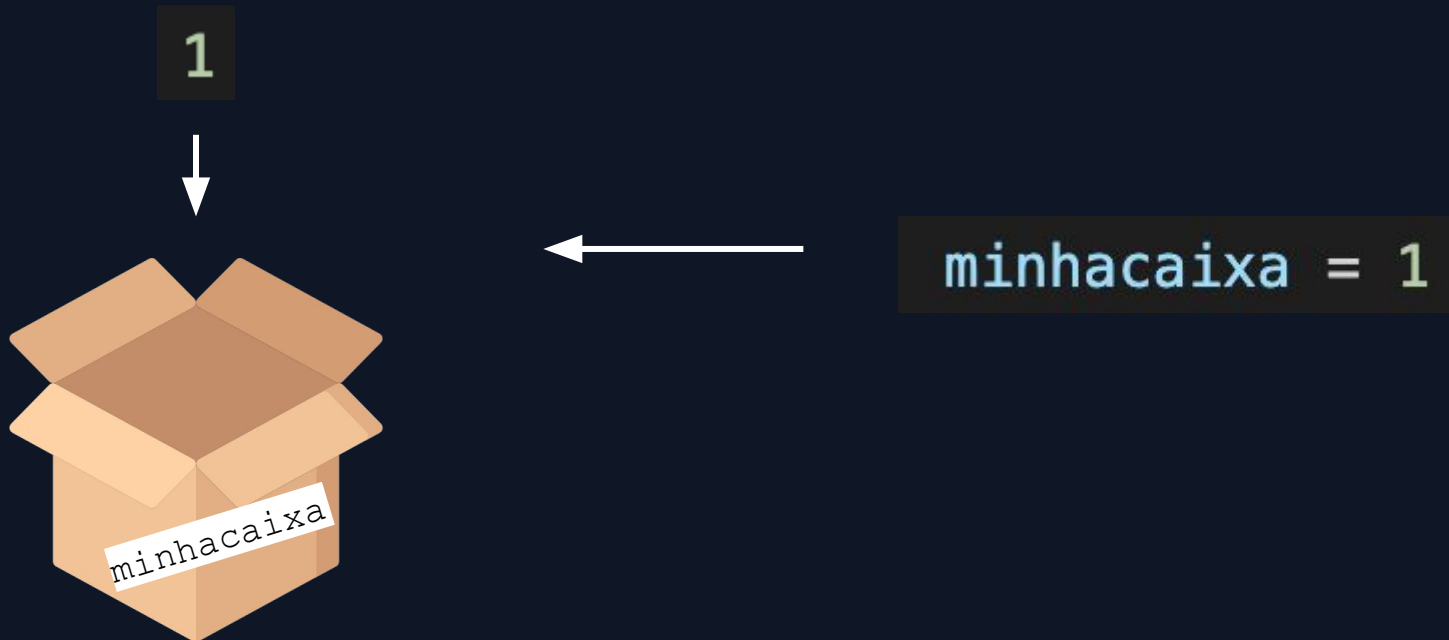
Tipos de dados, variáveis e memória

Atribuir um dado/valor a uma variável é o mesmo que guardar esse valor na caixa



Tipos de dados, variáveis e memória

Atribuir um dado/valor a uma variável é o mesmo que guardar esse valor na caixa



Tipos de dados, variáveis e memória

Atribuir um dado/valor a uma variável é o mesmo que guardar esse valor na caixa

"Texto aqui"



```
minhacaixa = "Texto aqui"
```

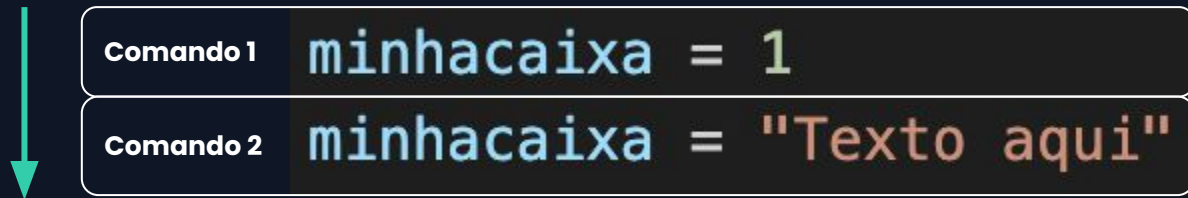
Tipos de dados, variáveis e memória

Diferente de uma caixa normal que utilizamos no dia-a-dia, em uma variável **só é possível guardar um único valor.**

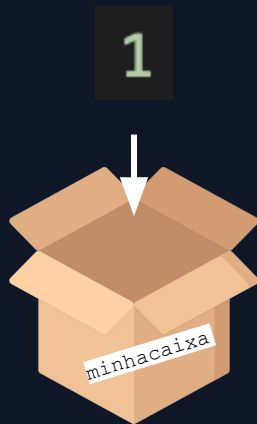


Tipos de dados, variáveis e memória

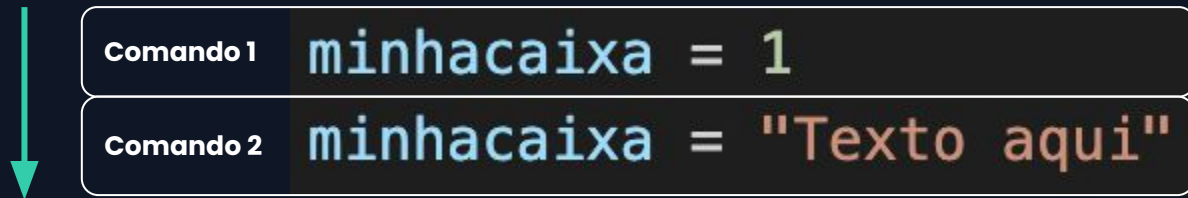
Ou seja, se tivermos a seguinte sequência de comandos:



Quando o computador executar o **comando 1**, a caixa terá o valor:



Tipos de dados, variáveis e memória




Mas quando ele terminar de executar o **comando 2**, o valor que a caixa vai ter armazenado é o valor:

`"Texto aqui"`



Ou seja, ao final desses 2 comandos, o valor final dentro da caixa será "Texto aqui"

Tipos de dados, variáveis e memória

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

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

Resumindo o que nós vimos até aqui:

Comando `console.log()`

Declaração de variáveis

Atribuição de valores

Comando `typeof()`

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

Tarefa do Dia 2 vamos disponibilizar agora!

Prazo final de envio da tarefa do Dia 2: sexta-feira (26/08)

Gravação e slides da aula de hoje tentaremos disponibilizar
ainda hoje.