

Introdução ao Terminal e GIT



git

SHELL

Shell é uma interface que processa comandos.

É um programa que espera seu input, reconhece o comando, processa e, por fim, mostra um output.

Exemplos: bash, cmd, powershell, etc.

TERMINAL

Terminal é o programa que envolve e roda o shell. Historicamente, terminais eram computadores que serviam apenas para executar o shell. Hoje em dia, com pcs mais potentes e interfaces gráficas, existem emuladores de terminal. Eles são as janelas onde rodamos o shell e seus comandos.

Git Bash



Utilizaremos o Bash disponível no link abaixo:

<https://gitforwindows.org/>

Comandos do Bash

Navegar entre pastas

Entrar na pasta

```
> cd nome_da_pasta
```

Sair da pasta

```
> cd ..
```

Comandos do Bash

Lista arquivos dentro de uma pasta

Listagem simples

> ls

Listagem completa

> ls -la

Comandos do Bash

Lista arquivos dentro de uma pasta

Mostrar o caminho atual

> pwd

Comandos do Bash

Criar uma pasta

> `mkdir nome_da_pasta`

Criar uma arquivo

> `touch nome_do_arquivo`

Comandos do Bash

Remover um arquivo ou pasta

Remover arquivo

➤ `rm nome_arquivo.txt`

Remover pasta

➤ `rm -rf nome_da_pasta`

Comandos do Bash

Remover um arquivo ou pasta

Mover arquivo

- `mv nome_arquivo caminho_desejado`

Renomear arquivo ou pasta

- `mv nome_atual nome_desejado`



git

Lista de Comandos GIT

Clonar um repositório já existente

```
> git clone https://github.com/repositorio
```

Lista de Comandos GIT

Verificar estado dos arquivos no repositório

```
> git status
```

Lista de Comandos GIT

Adicionar um arquivo no stage

```
> git add nome_do_arquivo
```

Adicionar todos os arquivos no stage

```
> git add .
```

Lista de Comandos GIT

Informar uma mensagem ao commitar o arquivo

```
> git commit -m "mensagem"
```

Lista de Comandos GIT

Enviar arquivos para o repositório

➤ `git push`

Baixar alterações do repositório para sua máquina

➤ `git pull`

GITHUB





Vamos colocar a mão na massa!



Repositórios Públicos:

- GIT: <https://github.com/git/git>
- Bootstrap: <https://github.com/twbs/bootstrap>
- Marcus: https://github.com/akelesis/digitalhouse_js

Conteúdos Adicionais:

- Comandos Básicos de Terminal
<https://www.youtube.com/watch?v=Vx0mrgs5Kel>
- Você sabe o que é Git?
<https://www.youtube.com/watch?v=3FpYydr8ggc>
- Configurando chaves SSH - Github
<https://www.youtube.com/watch?v=iVUnXw64Ez8>
- Documentação do Git
<https://git-scm.com/doc>

JavaScript

Introdução e sintaxe

```
if (top != self)
function calcWidth() {
    var wW = 0;
    if (typeof window.innerWidth == 'number') {
        wW = window.innerWidth;
    } else if (document.documentElement && document
        wW = document.documentElement.clientWidth;
    } else if (document.body && document.body.c
        wW = document.body.clientWidth;
    }
    if (sH = document.documentElement.scrollHe
        var wH = window.innerHeight || document
        wW = !document.all && (sH > wH) ?
        ('menu', 'width'
```

Para que serve o JS?

Como é uma linguagem de programação, os navegadores trazem um mecanismo de JS que permite executar JS no cliente, tornando as páginas interativas.

Além disso, o JavaScript pode servir para gerenciar a lógica do back-end com o node.js.

JS pode servir para desenvolver...

Jogo:

[Cookie Clicker](#)

[Sonic](#)

Calculadora:

[Calculator JS](#)

[Calculator JS 2](#)

E muito mais

E-mails, bate-papos, gráficos de dados, relógios, calculadoras, formulários, etc...

Relógio:

[Relógio - CSS + JS](#)

Efeitos especiais:

[Efeito 1](#)

[Efeito 2](#)

[Efeito 3](#)



Variáveis

Como declarar uma nova variável:

```
var nomeDaVariavel = valor
```

```
let nomeDaVariavel = valor
```

Alterar valor de uma variável

```
nomeDaVariavel = valor
```

Tipos básicos

"pepe"	string
10.7	number
true	boolean
{ }	object
[]	array
function() { }	function
null	null
undefined	undefined

Operadores aritméticos

<code>var numA = 5</code>	atribuição
<code>num1 + num2</code>	adição, subtração
<code>num1 * num2</code>	multiplicação, divisão
<code>num1 % 3</code>	módulo
<code>num1++</code>	incremento
<code>num1--</code>	decremento

Operadores de atribuição

<code>var numero = 10</code>	
<code>numero += 2</code>	<code>numero = numero + 2</code>
<code>numero -= 2</code>	<code>numero = numero - 2</code>
<code>numero *= 2</code>	<code>numero = numero * 2</code>
<code>numero /= 2</code>	<code>numero = numero / 2</code>

Operadores de comparação simples

==	igualdade de valor
!=	valor diferente
<=	menor e igual
>=	maior ou igual

Operadores de comparação estrita

===	igualdade de valor e tipo
!==	valor e tipo diferente

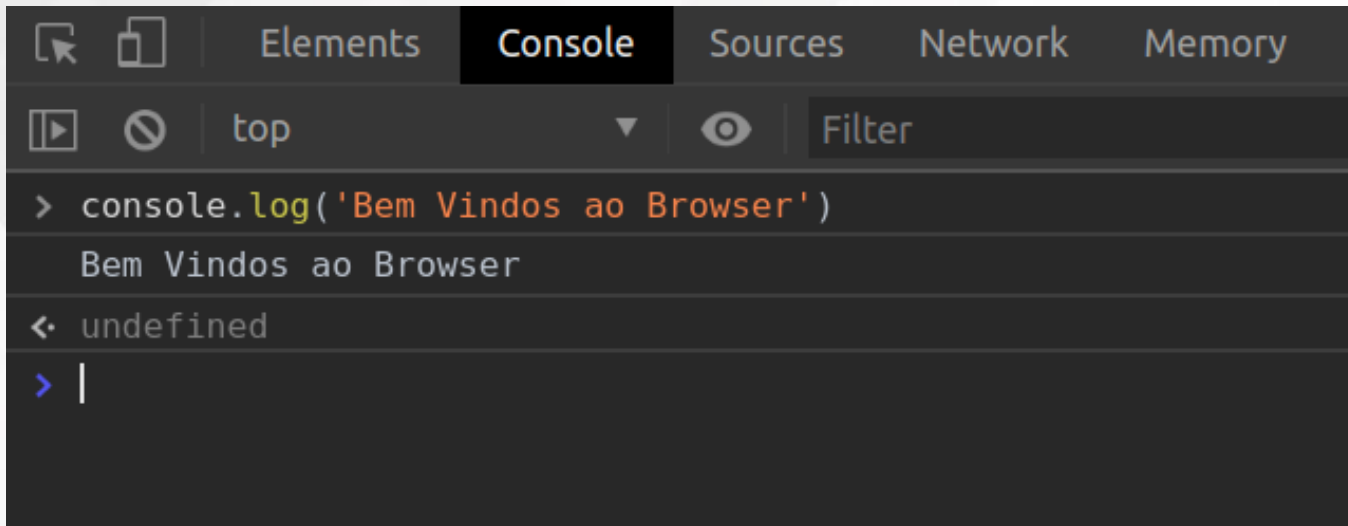
Operadores Lógicos


&&	Operador and (e)
	Operador or (ou)
!	Operador de negação

A linha de comando

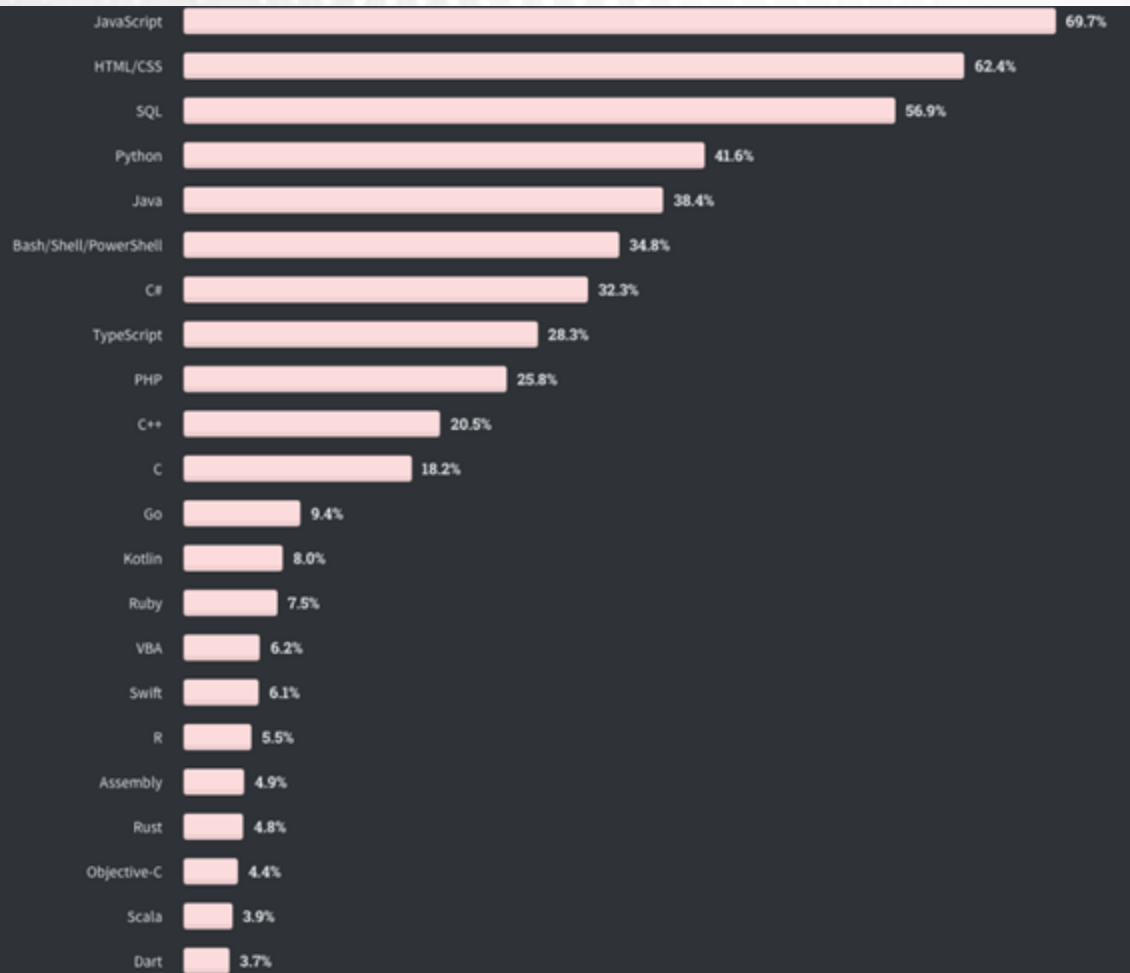
Os navegadores trazem uma linha de comando integrada para interpretar JavaScript.

Normalmente usamos F12 para abri-la e começar a escrever o código.





*“JavaScript é a linguagem mais popular
pelo oitavo ano consecutivo em
Fullstack, de acordo com a
StackOverflow” - [Stackoverflow 2020](#)*



Aproximadamente 70% dos desenvolvedores que responderam ao questionário do Stackoverflow de 2020 programam em Javascript

IF - Sintaxe

```
if (false) {  
    // fazer isto caso a condição seja  
    verdadeira  
} else {  
    // fazer isto caso a condição seja falsa  
}
```

IF - Exemplo

```
let diaDaSemana = 'Domingo'

if (diaDaSemana == 'Domingo') {
    console.log('Hoje é dia de churrasco!')
} else {
    console.log('Hoje não é dia de churrasco :(')
}
```

Resultado: Hoje é dia de churrasco!

IF ternário - Sintaxe

condição ? **expressão1** : **expressão2**

condição Qualquer expressão **booleana**

expressão1 Expressão retornada se **condição** for **true**

expressão2 Expressão retornada se **condição** for **false**

IF ternário - Exemplo

```
var diaDaSemana = 'Segunda-feira'
```

```
diaDaSemana === 'Domingo'
```

```
? console.log('hoje é dia de churrasco!')
```

```
: console.log('hoje não é dia de churrasco :(')
```

Resultado: Hoje não é dia de churrasco :(

Switch



```
var fruta = "mamão";
```

```
switch (fruta) {  
  case "abacaxi":  
    console.log("Me ajuda a descascar esse  
abacaxi?");  
    break;  
  case "mamão":  
    console.log("É mamão com açúcar!");  
    break;  
  default:  
    console.log("É outra fruta");  
}
```



For - Sintaxe

```
for (inicio; condicao; incremento) {  
    // fazer isso enquanto a condição for  
    verdadeira  
}
```


For - Exemplo

```
for (let i = 0; i < 3; i++) {  
  console.log('Olá ' + i)  
}
```

Resultado: Olá 0 Olá 1 Olá 2

For - Break

```
for (let i = 0; i < 4; i++) {  
    console.log(`Olá ' + i);  
    if(i === 1){  
        break; // interrompe o loop FOR  
    }  
}
```

Resultado: Olá o Olá 1

While - Sintaxe

```
while (condição) {  
    // executar enquanto a condição  
    for verdadeira  
}
```

While - Exemplo

```
let numero = 0
while (numero < 3) {
  console.log('Olá ' + numero)
  numero++
}
```

Resultado: Olá 0 Olá 1 Olá 2

Do While - Sintaxe

```
do {  
    // código a ser executado  
} while (condição);
```

O do while irá executar o código que está entre o do {} antes de verificar a condição.

Próxima aula - DOM

