

AULA 02 - PROGRAMAÇÃO FRONTEND

# INTRODUÇÃO AO JAVASCRIPT

Prof. Wagner Souza  
wagner.souza@uni9.pro.br

# RESUMO DA AULA

## TEMAS ABORDADOS

---

Introdução ao Javascript

Pontos importantes

Exemplos de códigos

Exercícios

# BRENDAN EICH

CRIADOR DA  
LINGUAGEM JAVASCRIPT

## JAVASCRIPT

Surgiu em torno de 1995

## ONDE ENCONTRÁ-LO?

<https://twitter.com/brendaneich>

<https://github.com/BrendanEich>



# BRENDAN EICH

## CRIADOR DA LINGUAGEM JAVASCRIPT

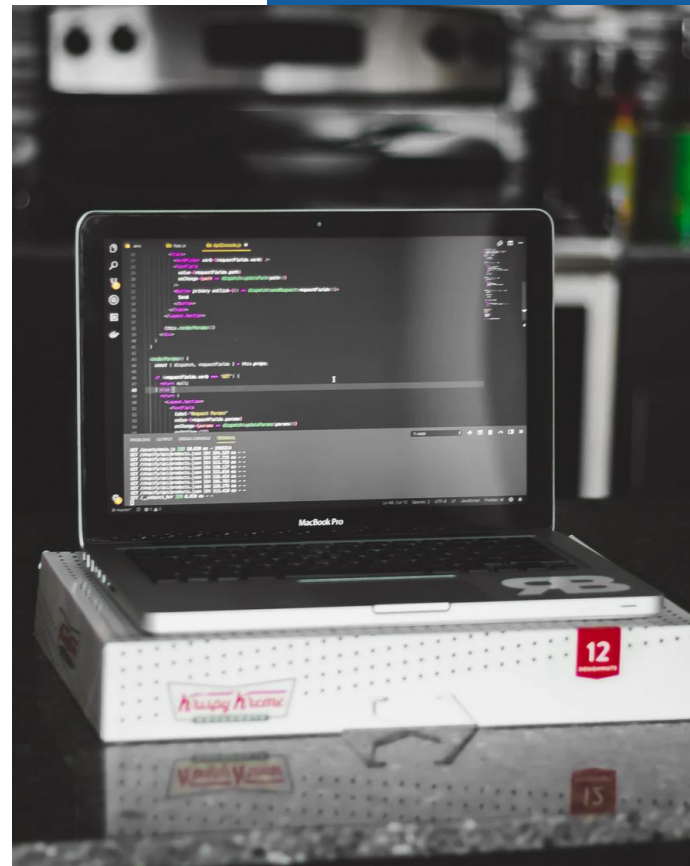
Brendan Eich é um programador de computadores americano e criador da linguagem de programação JavaScript. Ex-chefe do escritório de tecnologia na Mozilla Corporation.



# JAVASCRIPT

Linguagem interpretada

- MULTIPARADIGMA





# JAVASCRIPT != JAVA

## LINGUAGENS TOTALMENTE DISTINTAS

Existem diferenças cruciais entre as duas linguagens:

- Javascript é interpretada e pode ser usada tanto no Frontend como no Backend. Multiparadigma.
- Java é compilada e somente é usada no Backend. Orientada a objeto.

# TODOS PODEM VER O CÓDIGO JAVASCRIPT

PODEMOS DEIXÁ-LO ILEGÍVEL...

Existem diversas ferramentas para ofuscar nosso código "produtivo".

```
Welcome  JS eslintrc.es5.js  JS benchmark.js  JS stats.js  JS build.js
36     _resolve();
37   });
38   });
39 }
40
41 function getDefaultReactPath() {
42   return join(__dirname, 'remote-repo');
43 }
44
45 async function buildBenchmark(reactPath = getDefaultReactPath(), benchmark
46   // get the build.js from the benchmark directory and execute it
47   await require(join(__dirname, 'benchmarks', benchmark, 'build.js'))(
48     reactPath,
49     asyncCopyTo
50   );
51 }
52
53 async function getMergeBaseFromLocalGitRepo(localRepo) {
54   const repo = await Git.Repository.open(localRepo);
55   return await Git.Merge.base(
56     repo,
57     await repo.getHeadCommit(),
58     await repo.getBranchCommit('master')
59   );
60 }
61
62 async function buildBenchmarkBundlesFromGitRepo(
63   commitId,
64   skipBuild,
65   url = reactUrl,
66   clean
67 ) {
68   let repo;
69   const remoteRepoDir = getDefaultReactPath();
70
71   if (!skipBuild) {
72     if (clean) {
73       //clear remote-repo folder
74       await cleanDir(remoteRepoDir);
75     }
76     // check if remote-repo directory already exists
77     if (existsSync(join(__dirname, 'remote-repo')) {
78       repo = await Git.Repository.open(remoteRepoDir);
79       // fetch all the latest remote changes
80       await repo.fetchAll();
81       // push the repo to remote-repo folder
82       await repo.push(remoteRepoDir);
83     }
84   }
85 }
```

# JAVASCRIPT

## LINGUAGEM INTERPRETADA

---

BROWSER É O  
RESPONSÁVEL PELA  
INTERPRETAÇÃO





# O que da pra fazer com Javascript?



## SITES SIMPLES

Institucionais, portais, landing pages, blogs estáticos.



## E-COMMERCE

Lojas virtuais, venda de ingressos, compra de passagens aéreas.



## INTERNET BANKING

Transferências bancárias, pagamentos, investimentos.

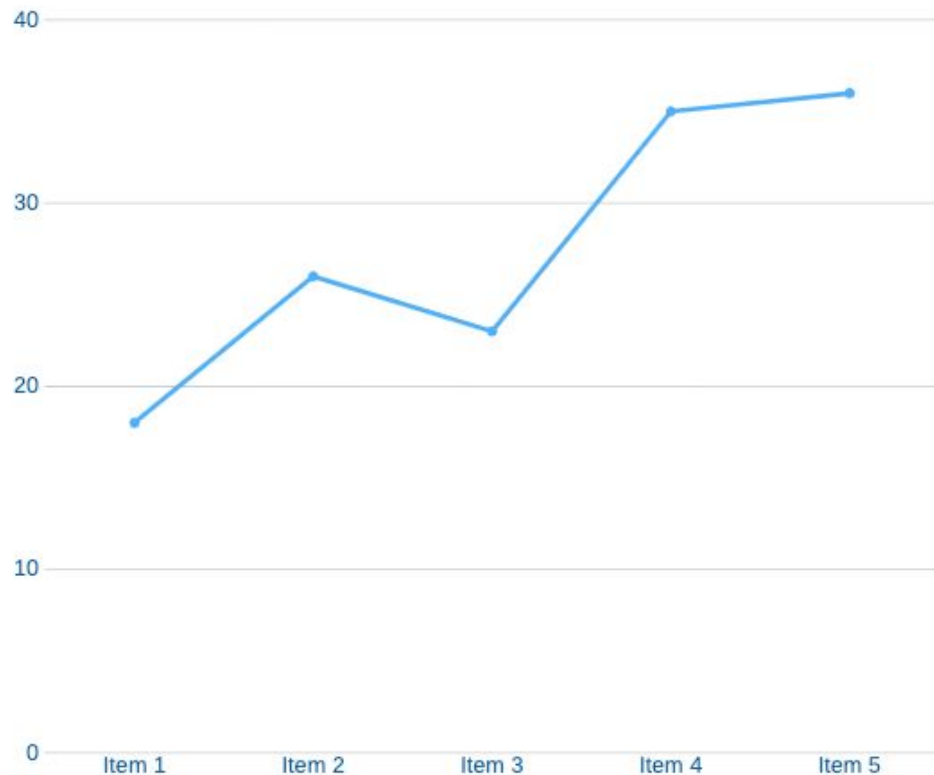


## PORTAIS

Sites de notícias, redes sociais.

# MUITAS APLICAÇÕES

JAVASCRIPT SE TORNOU  
MUITO POPULAR



# PRINCIPAIS NAVEGADORES

OS 5 MAIS USADOS SÃO:



Google Chrome



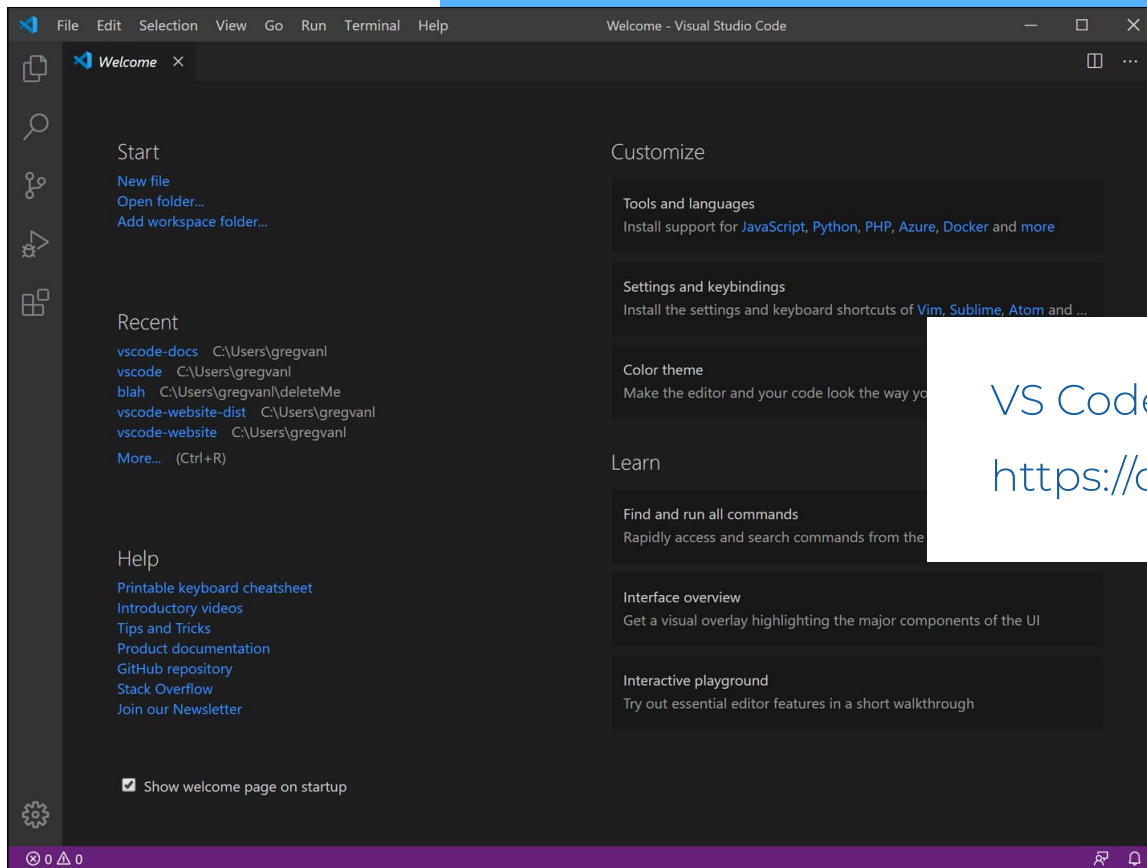
Mozilla Firefox



Safari



Opera



VS Code

<https://code.visualstudio.com/>

# SHOW ME THE CODE

## PRIMEIROS PASSOS

Vamos entender o conceito da linguagem

## NOSSO PRIMEIRO CÓDIGO

De fato, por a mão na massa

## MANIPULAÇÃO DO DOM

Alterar conteúdos da tela com javascript

## FUNÇÕES

Criando uma função

## FORMAS DE INSERIR JS NA PÁGINA

Temos três maneiras de fazer

# RESPONSABILIDADE DO JAVASCRIPT

## MANIPULAÇÃO DE DOM

Podemos trocar conteúdo, apagar, remover, inserir...

## COMPORTAMENTOS

Todo comportamento da sua tela é feito com javascript, por ex: abrir uma janela modal.

## POSSO MODIFICAR ESTILOS

Trocar cores, tamanho de fontes, imagens ...



# PRIMEIRO CÓDIGO

## ALERT();

Método javascript para exibir uma janela de diálogo do navegador com um alerta.

## ESSE MÉTODO RECEBE 1 PARAMÊTRO

Por Ex: `alert("Olá Mundo");` ou  
`alert(minhaVariavel);`



# DOM

DOCUMENT OBJECT MODEL

---

## RESUMIDAMENTE

TAGS HTML / CAMADAS





# MANIPULANDO DOM

## MANIPULANDO UM ELEMENTO

`document`

`.getElementById("nomeDoId")`

`.innerHTML = "Novo conteúdo";`



# CRIANDO E EXECUTANDO FUNÇÕES

## O QUE É UMA FUNÇÃO

Um bloco de código que faz alguma coisa.

## NUNCA SOBRECARREGUE-A

Crie funções simples, que resolvem pequenos problemas, funções grandes tendem a ficar complexas.

## EXISTEM 3 TIPOS DE FUNÇÕES

Anônima, nomeada e declarada





# FUNÇÃO - FUNCTION

```
function nomeDaFuncao(PARAMETROS) {  
    // COMANDOS  
}
```

# Três maneiras de vincular js

## INTERNO

Dentro da própria página html, podemos inserí-lo entre a tag `<head></head>` ou dentro da tag `<body></body>`

## DE UMA PASTA

Podemos ter também um arquivo dentro de um diretório do nosso projeto que contenha todos os nossos scripts.

## EXTERNO (URL)

Podemos chamar um script externo, de uma url na internet, podendo ser uma cdn, ou até mesmo um script de terceiros.

```
ws.on("message", m => {  
  let a = m.split(" ")  
  switch(a[0]){  
    case "connect":  
      if(a[1]){  
        if(clients.has(a[1])){  
          ws.send("connected");  
          ws.id = a[1];  
        }else{  
          ws.id = a[1]  
          clients.set(a[1], {clients: (position) => {  
            ws.send("connected")  
          }  
        }  
      }  
    }  
  }  
})
```

# OBRIGADO

AULA 02 - PROGRAMAÇÃO FRONTEND

WAGNER.SOUZA@UNI9.PRO.BR