



JavaScript

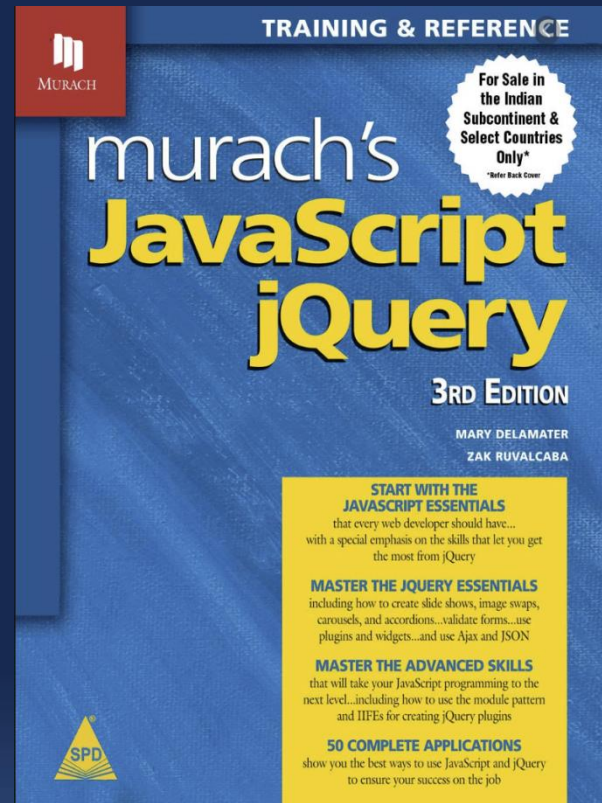
Unidade 1 – Introdução



Prof. Aparecido V. de Freitas
Doutor em Engenharia
da Computação pela EPUSP
aparecidovfreitas@gmail.com



Bibliografia



Introdução

- ✓ Linguagem que domina o mundo das aplicações **WEB**;
- ✓ Obrigatória para todo desenvolvedor **WEB**;
- ✓ Interpretada em uma página carregada em um Browser;
- ✓ Browsers vêm equipados com interpretadores JavaScript, conhecidos por **JavaScript engines**;
- ✓ Pode também ser executada em servidores **JavaScript-based** (**Node.js**)

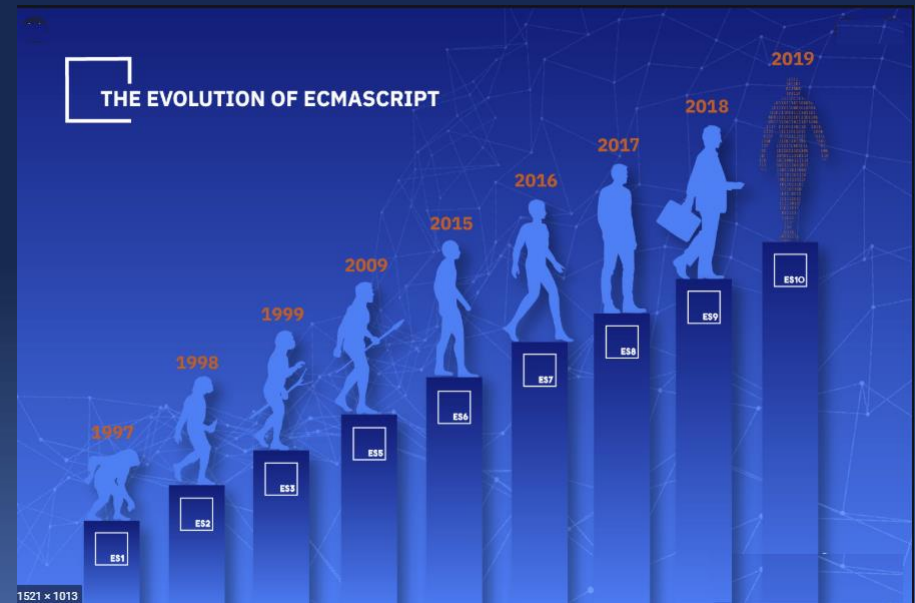


Histórico

- ✓ Disponível inicialmente no **Netscape Navigator** 2 com o nome LiveScript;
- ✓ 1997 - **JavaScript** foi normatizada pela **Ecma** International e renomeada para ECMAScript;
- ✓ A grande maioria atual dos browsers segue a especificação ECMA para implementar os engines JavaScript;



ECMA





O que posso fazer com JavaScript ?





O que posso fazer com JavaScript ?

- ✓ Uso principal para possibilitar **interação** entre usuários;
- ✓ Criado inicialmente para validação de formulários em input de dados;
- ✓ Graças aos avanços nos **JavaScript engines**, hoje pode ser considerada uma linguagem de uso geral;
- ✓ Exemplos de aplicações: Google Maps, Google Calendar, Office Web MS, etc





O que é necessário para se estudar Javascript ?





Setup de Ambiente

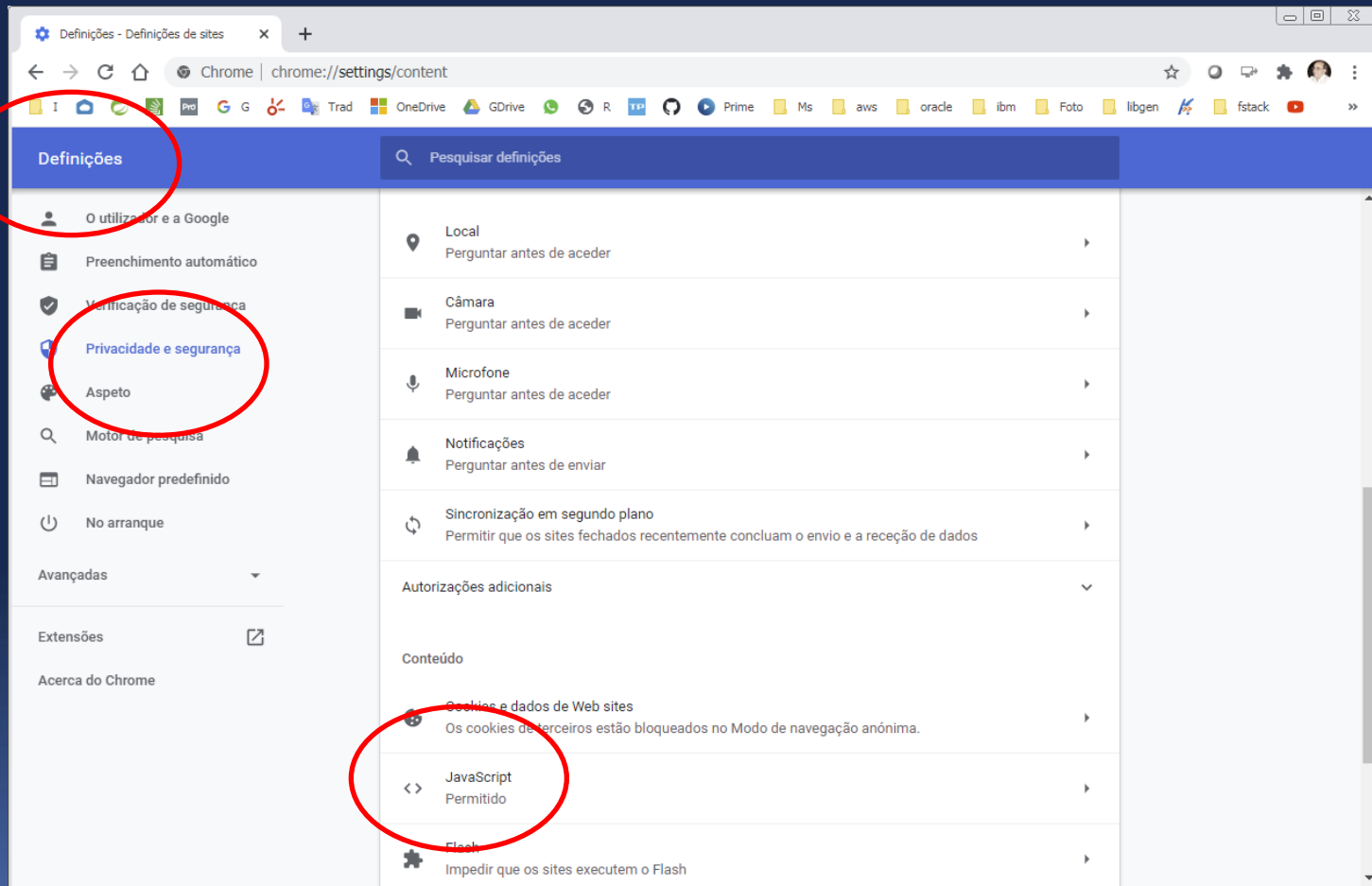
✓ Utilizaremos neste curso as seguintes ferramentas:





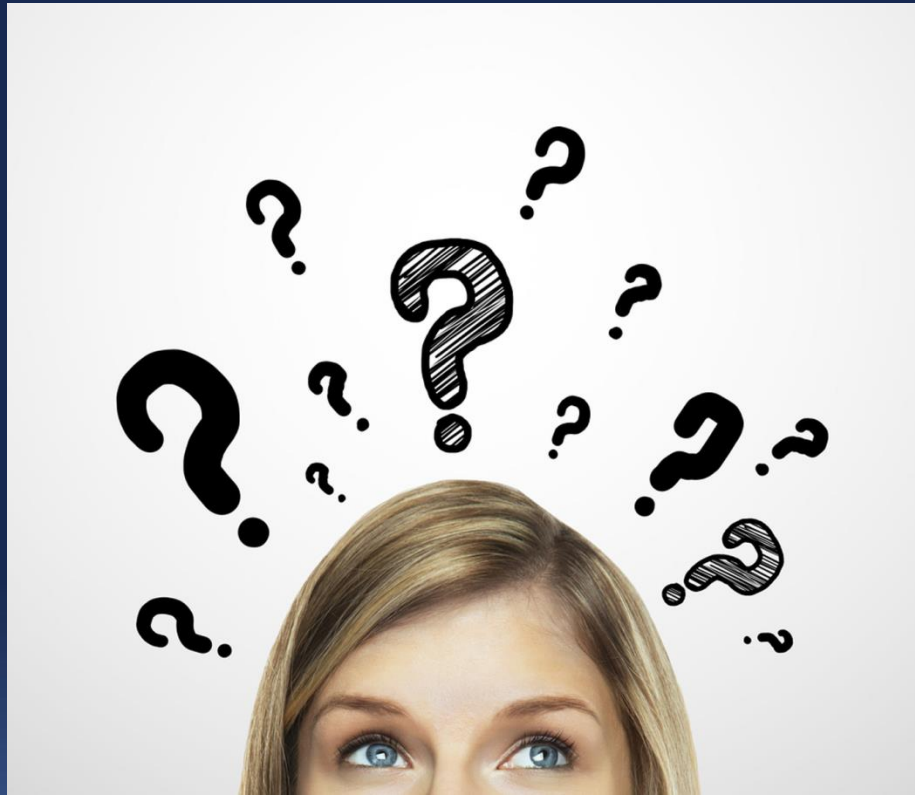
Uso de JavaScript pelos Browsers

- ✓ Por default, a maioria dos Browsers têm **suporte** ao JavaScript habilitado;
- ✓ Para estar certo de que JavaScript está habilitado no Google Chrome:





Como o JavaScript é embutido nas páginas HTML?





Como o JavaScript é embutido nas páginas HTML ?

- ✓ A inserção do código **JavaScript** nas páginas HTML é feita do mesmo modo que se inclui qualquer conteúdo em HTML;
- ✓ Usamos as tags **<script>** e **</script>** para marcar o início e o fim do nosso código JavaScript;
- ✓ Desenvolvedores **JavaScript** tipicamente incluem essas tags entre **<body>** e **</body>**;
- ✓ Essas marcações informam ao Browser que o texto delimitado pelos marcadores não é para ser exibido, e sim para ser interpretado pelo **JavaScript** engine.

```
<script>  
    // JavaScript goes here  
</script>
```



Linkando com um arquivo externo

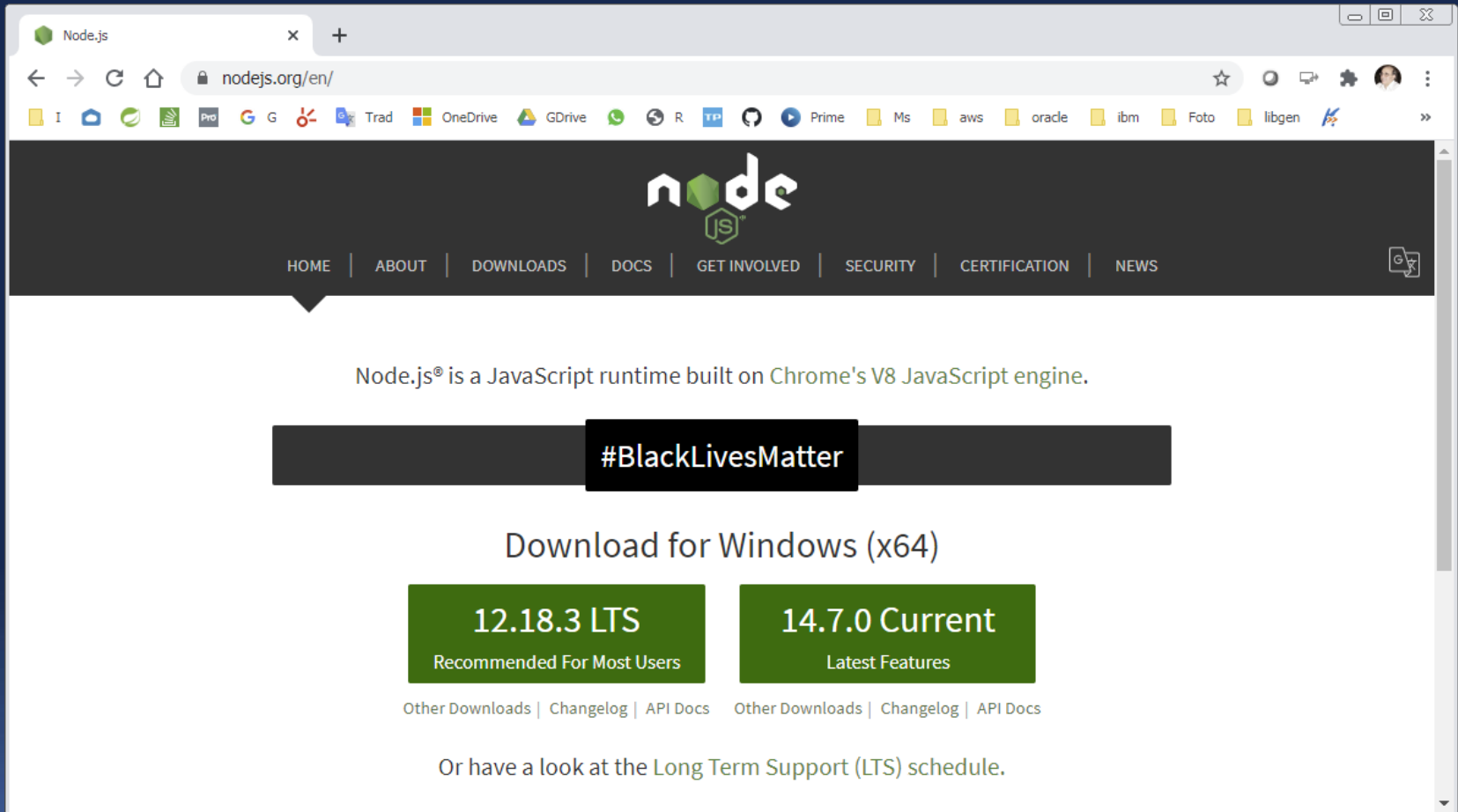
- ✓ Pode-se definir um arquivo externo **JavaScript** e linká-lo à página **HTML**;
- ✓ O arquivo externo deve ter - preferencialmente - a extensão **.js**
- ✓ Essa ligação é feita pelo atributo **src** da tag **<SCRIPT>**;
- ✓ O browser efetua a leitura do arquivo e o inclui como parte da página HTML;

```
<script src="MyCommonFunctions.js"></script>
```



Setup do ambiente – node.js

- Baixar de <https://nodejs.org>





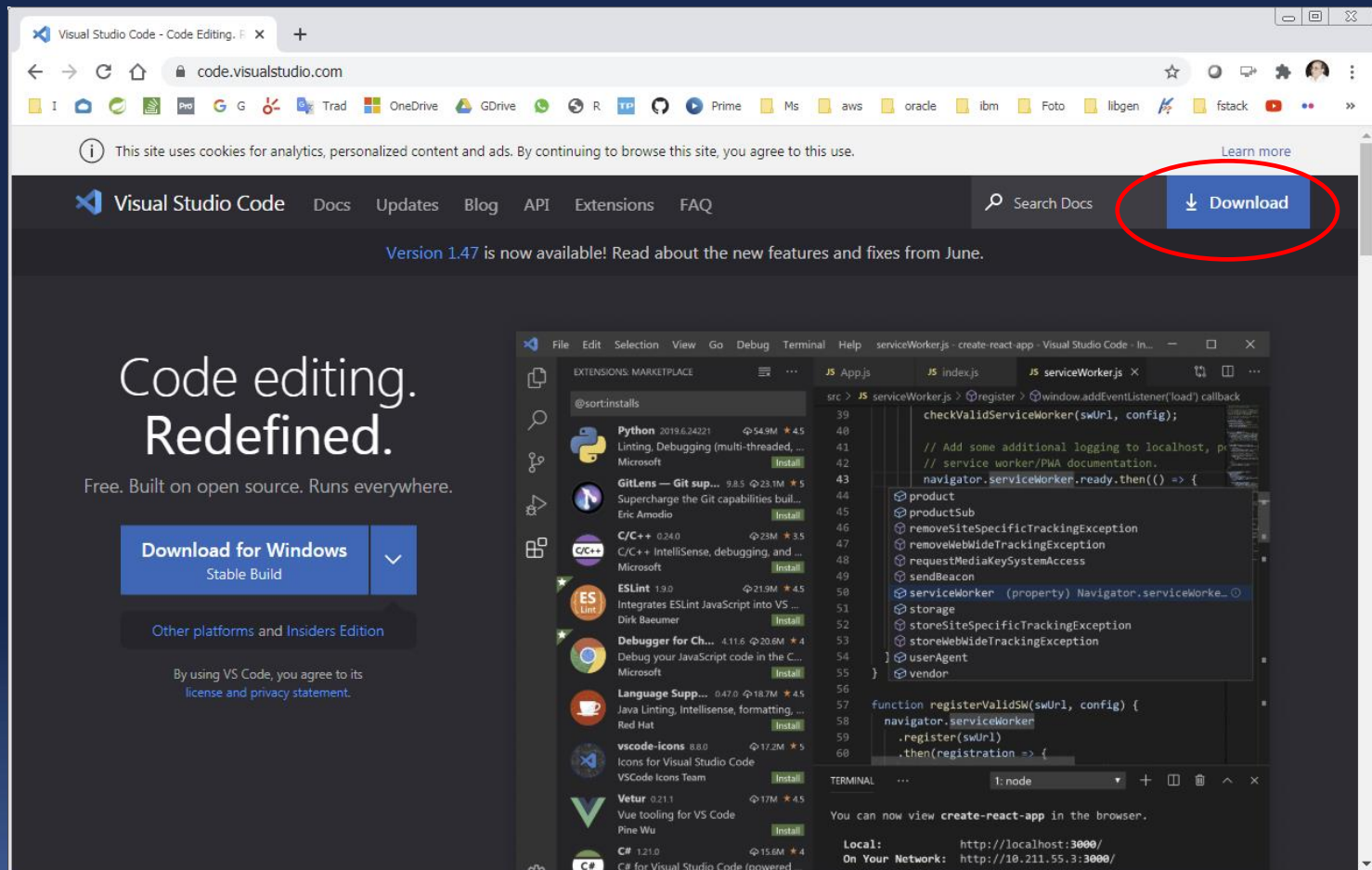
Setup do ambiente – node.js

- Após instalação => `node --version`

```
Command Prompt
C:\>node --version
v12.18.3
C:\>
```


Setup do ambiente – VS Code

- Baixar de: <https://code.visualstudio.com>





Setup do ambiente – VS Code

- Baixar de: <https://code.visualstudio.com>

Download Visual Studio Code - Mac x +

code.visualstudio.com/Download

Visual Studio Code Docs Updates Blog API Extensions FAQ Search Docs Download

Version 1.47 is now available! Read about the new features and fixes from June.

Download Visual Studio Code

Free and built on open source. Integrated Git, debugging and extensions.

↓ Windows
Windows 7, 8, 10

User Installer	64 bit 32 bit ARM
System Installer	64 bit 32 bit ARM
.zip	64 bit 32 bit ARM

↓ .deb
Debian, Ubuntu

↓ .rpm
Red Hat, Fedora, SUSE

.deb	64 bit
.rpm	64 bit
.tar.gz	64 bit

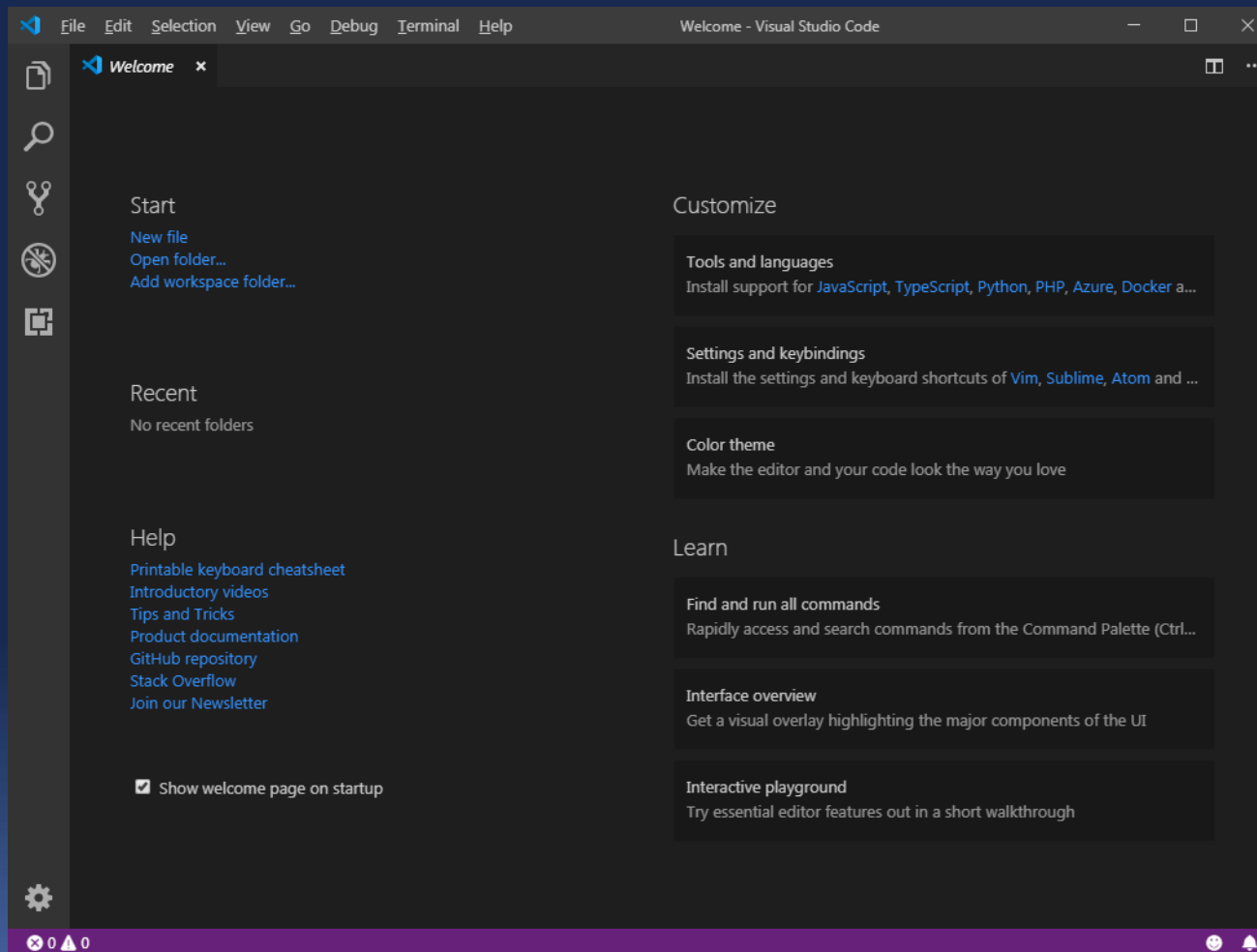
Snap Store

↓ Mac
macOS 10.10+

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

Setup do ambiente – VS Code

- Baixar de: <https://code.visualstudio.com>





Setup do ambiente – VS Code

- Baixar de: <https://code.visualstudio.com>

```
Command Prompt

C:\>code --version
1.47.3
91899dcef7b8110878ea59626991a18c8a6a1b3e
x64

C:\>
```



Gerando código js com VSCode

- Entrar com: `> code nomeArquivo.js`

```
Command Prompt
E:\Java_Full_Stack\JavaScript\Fontes_JavaScript\Unidade_1>code js1_01.js
```



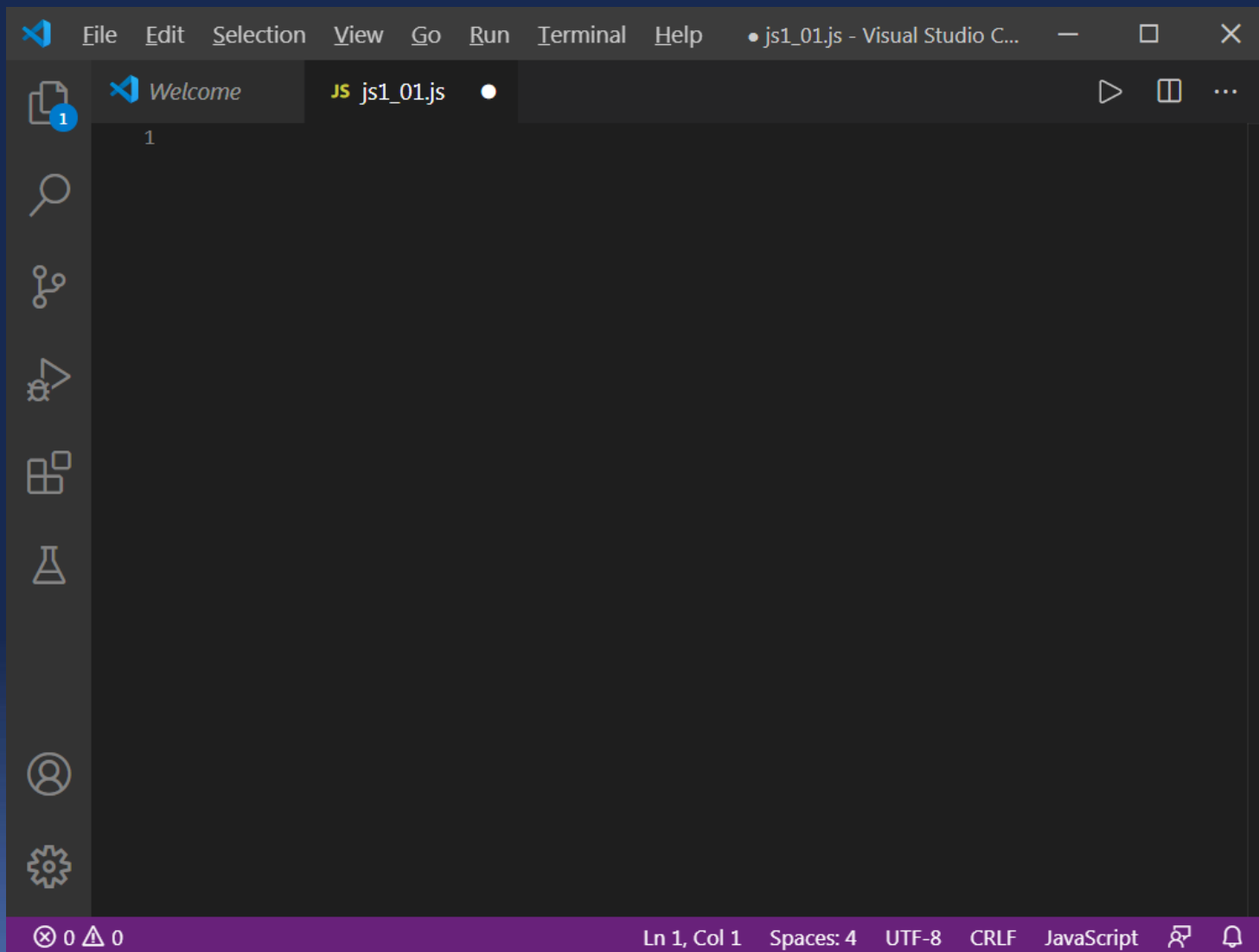
Gerando código js com VSCode

- Entrar com: `> code nomeArquivo.js`

```
Command Prompt
E:\Java_Full_Stack\JavaScript\Fontes_JavaScript\Unidade_1>code js1_01.js
```



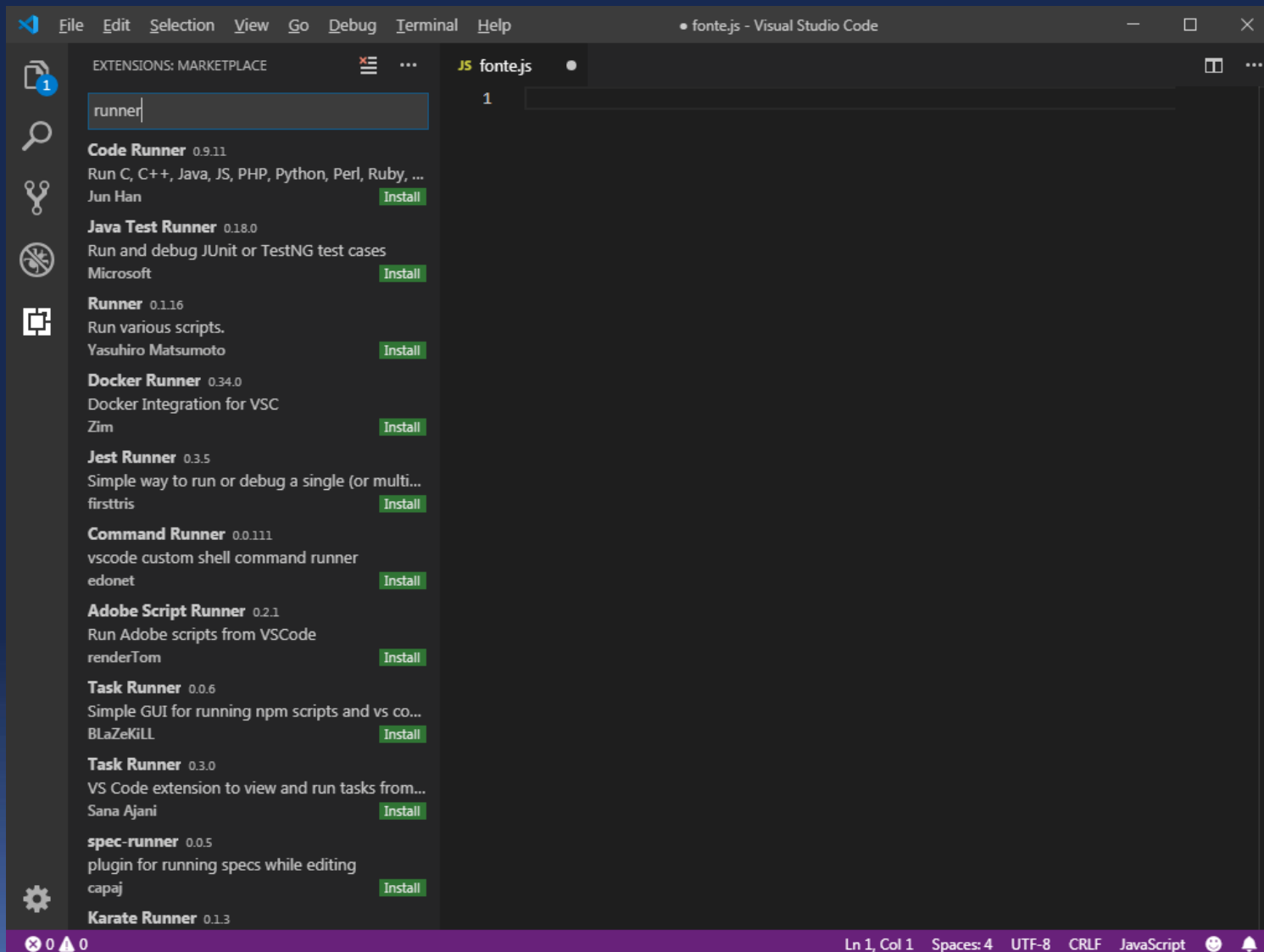

Gerando código js com VSCode





Instalar plugin para JavaScript

- View Extensions > Code Runner





Instalar plugin para JavaScript

- View Extensions > Code Runner

The screenshot shows the Visual Studio Code interface with the 'Code Runner' extension installed. The left sidebar displays the 'EXTENSIONS: MARKETPLACE' view with a search bar containing 'runner'. A list of extensions is shown, including 'Code Runner' (0.9.11) by Jun Han, which is marked as 'Installed'. The main editor area shows the 'Code Runner' extension details, including its name, version, author, and a list of supported languages. The extension is also marked as 'Installed' with an 'Uninstall' button. The bottom status bar shows the 'JS' language selected.

EXTENSIONS: MARKETPLACE

runner

Code Runner 0.9.11
Run C, C++, Java, JS, PHP, Python, Perl, Ruby, ...
Jun Han
✓ Installed

Java Test Runner 0.18.0
Run and debug JUnit or TestNG test cases
Microsoft
Install

Runner 0.1.16
Run various scripts.
Yasuhiro Matsumoto
Install

Docker Runner 0.34.0
Docker Integration for VSC
Zim
Install

Jest Runner 0.3.5
Simple way to run or debug a single (or multi...
firsttris
Install

Command Runner 0.0.111
vscode custom shell command runner
edonet
Install

Adobe Script Runner 0.2.1
Run Adobe scripts from VSCode
renderTom
Install

Task Runner 0.0.6
Simple GUI for running npm scripts and vs co...
BLaZeKILL
Install

Task Runner 0.3.0
VS Code extension to view and run tasks from...
Sana Ajani
Install

spec-runner 0.0.5
plugin for running specs while editing
capaj
Install

Karate Runner 0.1.3

Code Runner formulahendry.code-runner
Jun Han | 7,183,562 | ★★★★★ | Repository | License
Run C, C++, Java, JS, PHP, Python, Perl, Ruby, Go, Lua, Groovy, PowerShell, C...
✓ Installed Uninstall

Code Runner
chat on gitter

Run code snippet or code file for multiple languages: C, C++, Java, JavaScript, PHP, Python, Perl, Perl 6, Ruby, Go, Lua, Groovy, PowerShell, BAT/CMD, BASH/SH, F# Script, F# (.NET Core), C# Script, C# (.NET Core), VBScript, TypeScript, CoffeeScript, Scala, Swift, Julia, Crystal, OCaml Script, R, AppleScript, Elixir, Visual Basic .NET, Clojure, Haxe, Objective-C, Rust, Racket, AutoHotkey, AutoIt, Kotlin, Dart, Free Pascal, Haskell, Nim, D, Lisp, Kit, and custom command

Sponsors
codestream
Discuss, review, and share code with your team in VS Code. Links discussions about code to your code. Integrates w/ Slack, Jira, Trello, and Live Share. Try it free.
CODING 提供 DevOps 全流程应用，从需求到实现，一站式开发体验。 CODING is a one-stop platform for



Instalar plugin para JavaScript

The screenshot shows the Visual Studio Code interface with the 'EXTENSIONS: MARKETPLACE' view on the left. The 'Code Runner' extension by Jun Han is highlighted with a red circle. The extension is marked as 'Installed'. The main panel displays the 'Code Runner' details, including its description, repository, and sponsors.

Code Runner 0.9.11
Run C, C++, Java, JS, PHP, Python, Perl, Ruby, ...
Jun Han
✓ Installed

Code Runner formulahendry.code-runner
Jun Han | 7,183,562 | ★★★★★ | Repository | License
Run C, C++, Java, JS, PHP, Python, Perl, Ruby, Go, Lua, Groovy, PowerShell, C...

Code Runner
Run code snippet or code file for multiple languages: C, C++, Java, JavaScript, PHP, Python, Perl, Perl 6, Ruby, Go, Lua, Groovy, PowerShell, BAT/CMD, BASH/SH, F# Script, F# (.NET Core), C# Script, C# (.NET Core), VBScript, TypeScript, CoffeeScript, Scala, Swift, Julia, Crystal, OCaml Script, R, AppleScript, Elixir, Visual Basic .NET, Clojure, Haxe, Objective-C, Rust, Racket, AutoHotkey, AutoIt, Kotlin, Dart, Free Pascal, Haskell, Nim, D, Lisp, Kit, and custom command

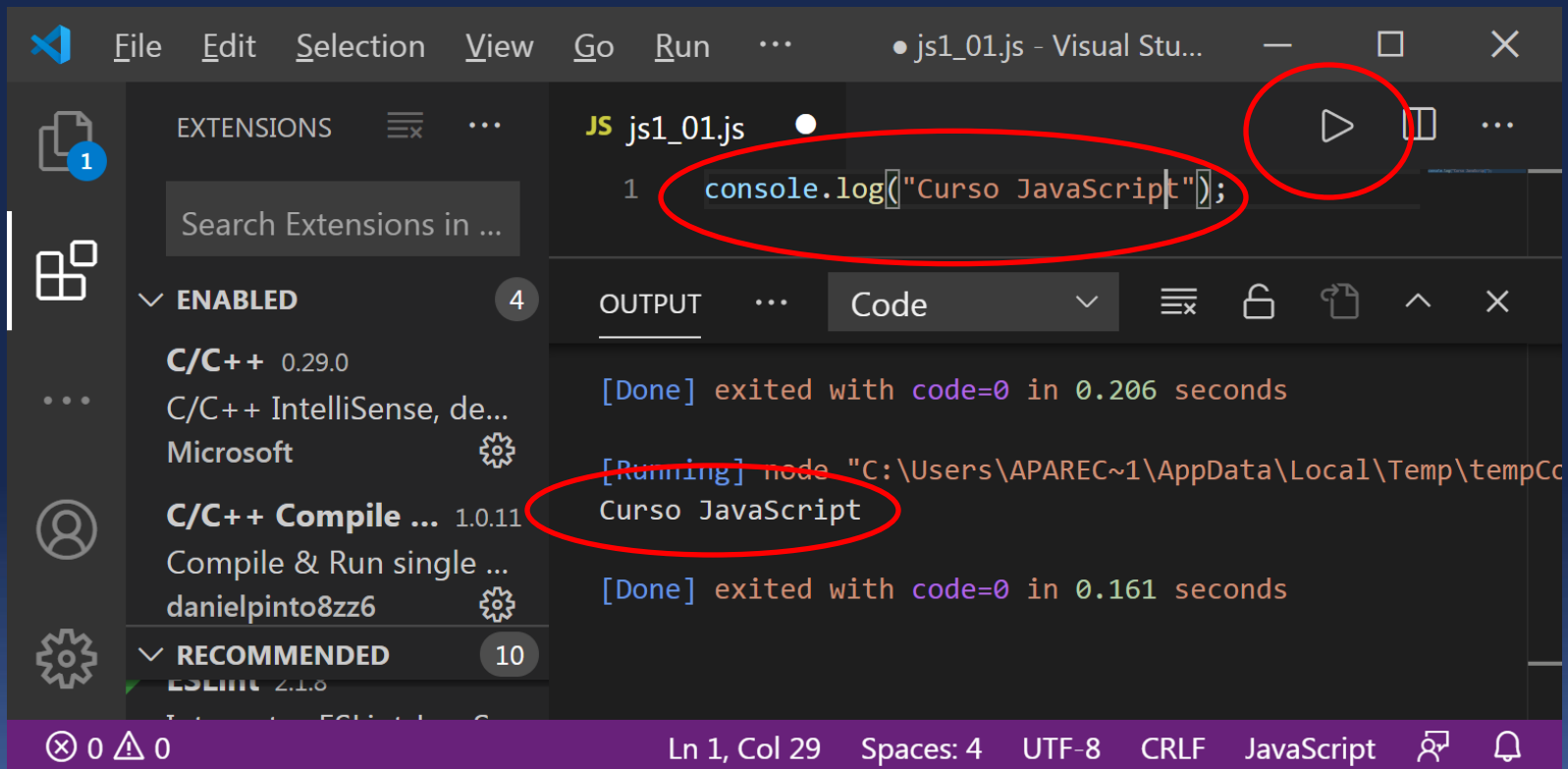
Sponsors
Discuss, review, and share code with your team in VS Code. Links discussions about code to your code. Integrates w/ Slack, Jira, Trello, and Live Share. [Try it free.](#)

CODING 提供 DevOps 全流程应用，从需求到实现，一站式开发体验。 CODING is a one-stop platform for entire software development lifecycle, from requirement to development.



Code Visual Studio – Plugin runner

- ✓ Com o plugin code runner disponível na **IDE Code Visual Studio**, pode-se executar o código JavaScript diretamente no Node.js;
- ✓ Ou seja, o arquivo **.js** é passado e executado pelo Node.js





Primeiro código JavaScript

Unidade_1_01.html

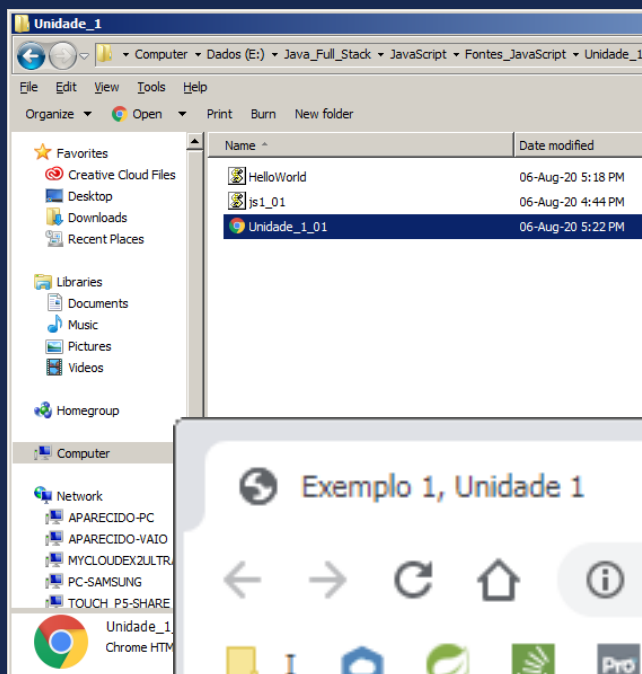
```
<!DOCTYPE html>

<html lang="en">
  <head>
    <meta charset="utf-8" />
    <title>Exemplo 1, Unidade 1</title>
  </head>
  <body bgcolor="white">
    <p>Parágrafo 1</p>
    <script>
      document.bgColor = "red";
    </script>
  </body>
</html>
```




Carregando no Browser

Unidade_1_01.html



```
<!DOCTYPE html>

<html lang="en">
  <head>
    <meta charset="utf-8" />
    <title>Exemplo 1, Unidade 1</title>
  </head>
  <body bgcolor="white">
    <p>Parágrafo 1</p>
    <script>
      document.bgColor = "red";
    </script>
  </body>
</html>
```





Observação

Unidade_1_01.html

- ✓ O atributo **bgcolor** da tag <body> está definindo a cor "white";
- ✓ Mas, a página exibida é "red";
- ✓ Para fins de script, a página é um "document" que possui muitas propriedades;
- ✓ JavaScript alterou a cor de fundo do documento para red.

```
<!DOCTYPE html>

<html lang="en">
  <head>
    <meta charset="utf-8" />
    <title>Exemplo 1, Unidade 1</title>
  </head>
  <body bgcolor="white">
    <p>Parágrafo 1</p>
    <script>
      document.bgColor = "red";
    </script>
  </body>
</html>
```





Renderização

- ✓ Quando o browser faz a carga de uma página **HTML**, o faz renderizando-a tag por tag de cima para baixo;
- ✓ Este processo é conhecido por "**parsing**";
- ✓ No exemplo anterior, o browser iniciou o **parsing** a partir da primeira tag <body> encontrada e definiu a cor do documento "white";
- ✓ Continuando o "**parsing**" ao encontrar o script **JavaScript**, a cor do documento foi alterada para "red".



Fluxo de Renderização

- ✓ Vejamos um segundo exemplo: [Unidade_1_02.html](#)

```
<!DOCTYPE html>

<html lang="en">
  <head>
    <meta charset="utf-8" />
    <title>Capítulo 1, Exemplo 2</title>
  </head>
  <body bgcolor="white">
    <p>Parágrafo 1</p>
    <script>
      // script Bloco 1
      document.bgColor = "green";
      alert("Primeiro Bloco JavaScript");
    </script>
    <p>Parágrafo 2</p>
    <script>
      // script Bloco 2
      document.bgColor = "red";
      alert("Segundo Bloco JavaScript");
    </script>
    <p>Parágrafo 3</p>
  </body>
</html>
```



Fluxo de Renderização

✓ Vejamos um segundo exemplo: [Unidade_1_02.html](#)

The image illustrates the rendering flow of a web page through three overlapping browser window screenshots. The top window shows the initial state with a white background and a tooltip that reads "Esta página diz: Primeiro Bloco JavaScript". The middle window shows the first paragraph rendered in green, with a tooltip that reads "Esta página diz: Segundo Bloco JavaScript". The bottom window shows the full page rendered in red, with three paragraphs labeled "Parágrafo 1", "Parágrafo 2", and "Parágrafo 3".



Exibindo resultados em uma página

Unidade_1_03.html

```
<!DOCTYPE html>

<html lang="en">
  <head>
    <meta charset="utf-8" />
    <title>Capítulo 1, Exemplo 3</title>
  </head>
  <body>
    <p id="results"></p>
    <script>
      document.getElementById("results").innerHTML = "Hello World!";
    </script>
  </body>
</html>
```




Exibindo resultados em uma página

Unidade_1_03.html

```
<!DOCTYPE html>

<html lang="en">
  <head>
    <meta charset="utf-8" />
    <title>Capítulo 1, Exemplo 3</title>
  </head>
  <body>
    <p id="results"></p>
    <script>
      document.getElementById("results").innerHTML = "Hello World!";
    </script>
  </body>
</html>
```

- ✓ Neste exemplo, o elemento `<p/>` tem um **ID** definido pelo atributo **id**;
- ✓ Este **ID** deve ser único na página, pois ele é usado pelo **JavaScript** para identificar o elemento seguinte linha:

```
document.getElementById("results").innerHTML = "Hello World!";
```

- ✓ Basicamente a linha acima diz: "Obtenha o elemento com o **ID** **"results"** e defina o **HTML** dentro desse elemento para **"Hello World!"** .