

IOS – Instituto de
Oportunidade Social

JS 01 - Introdução ao JS



- Conhecer a linguagem de programação JavaScript;
- Familiarizar-se com as extensões VsCode voltadas ao Javascript;
- Iniciar a compreensão da sintaxe da linguagem;
- Criar sua primeira página utilizando JavaScript.

IOS – Instituto de
Oportunidade Social

Introdução ao JS



Introdução

JavaScript (ou apenas JS) é uma linguagem de programação, que está de acordo com a especificação **ECMAScript**. O JS é uma **linguagem de alto nível interpretada com tipagem dinâmica fraca e mutiparadigma** (protótipos, **orientado a objeto**, imperativo e, funcional).

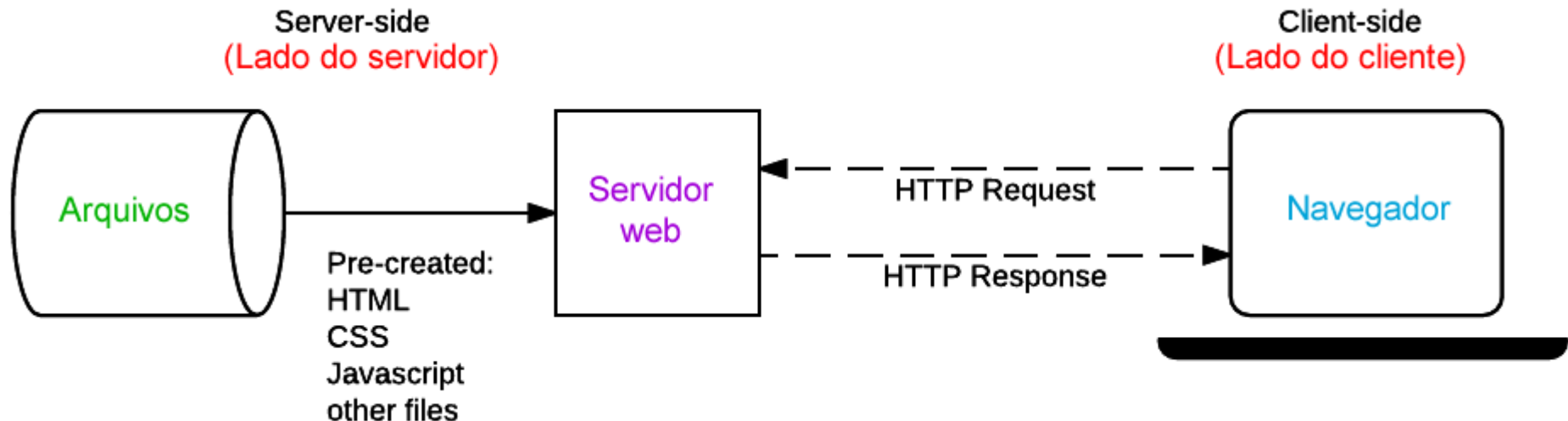
O **JS**, além do HTML e do CSS, é uma das **tecnologias bases** da World Wild **Web**. E, atualmente é uma das linguagens mais utilizadas do lado do cliente, mas também pode ser utilizada do lado do servidor por meio de ambiente como o **NodeJS**.

Client e Server

A Programação Web possui dois principais lados: programação do lado do **servidor (Back-End)** e programação do lado do **cliente (Front-End)**. Na programação do lado do servidor, o código executa em um servidor web. E nesse caso, os navegadores comunicam-se com **web servers (JSON)** utilizando **requisições HTTP** (*HyperText Transfer Protocol*). Ao clicar em um link em uma página da web, seja para enviar um formulário ou para fazer uma pesquisa, uma **HTTP request** (solicitação HTTP) é enviada do seu navegador para o servidor de destino.

Exemplo de Requisição e Resposta

Na programação do lado do **cliente**, os programas são executados no **computador do usuário** utilizando **scripts**, que são carregados juntos com os arquivos HTML e CSS.



Características

O JS é uma **linguagem imperativa e estruturada**, que suporta os elementos de sintaxe de programação estruturada da linguagem C como, por exemplo, **if**, **while**, **switch**, etc.

O JS também possui **tipagem dinâmica** e baseada em **objetos**. **Objetos JavaScript** são **arrays associativos**, potencializados com um protótipo e **cada chave** fornece o nome para **uma propriedade de objeto**. Propriedades e seus valores podem ser adicionadas, mudadas, ou deletadas em **tempo de execução**.

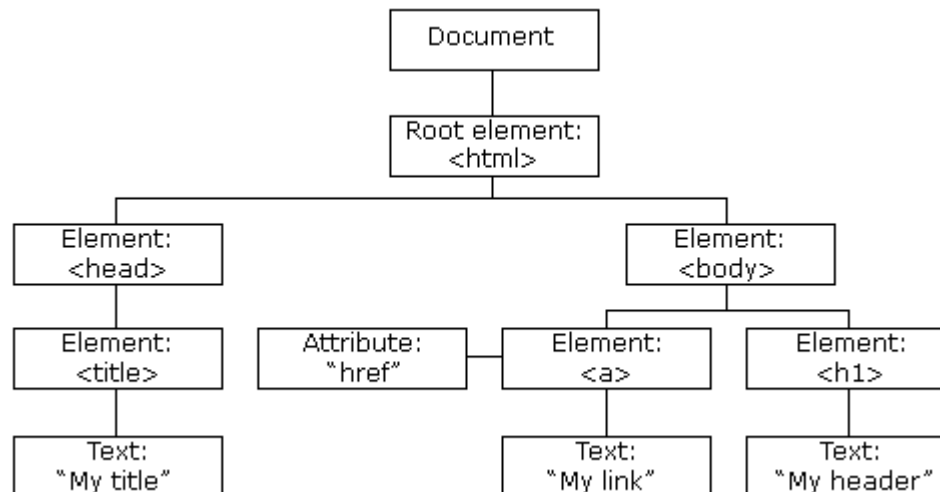
Vanilla JavaScript

Vanilla JavaScript ou **Vanilla JS** se referem ao JavaScript desenvolvido puramente, sem o suporte de qualquer estrutura ou biblioteca adicional.

Scripts escritos em Vanilla JS são códigos **JavaScript simples**. O que vamos aprender primeiramente aqui é programar códigos em Vanilla JS e depois aprenderemos como utilizar uma **biblioteca/framework** para auxiliar no desenvolvimento das aplicações web.

JavaScript e DOM

O HTML DOM (Document Object Model) permite o JavaScript acessar e modificar todos os elementos HTML em uma página web. O navegador cria o DOM da página com a estrutura de elementos e a árvore de objetos dessa página. O JS pode adicionar, alterar ou remover os elementos, os atributos e estilos CSS da página.

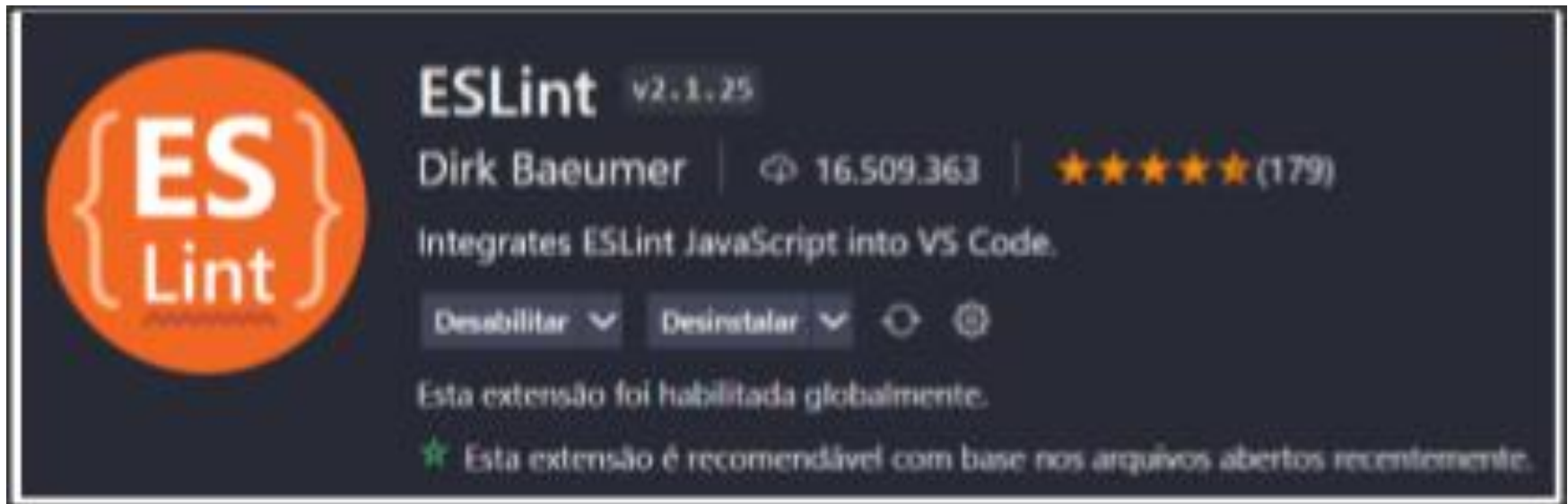


IOS – Instituto de
Oportunidade Social

Extensões do JS



A extensão **ES Lint** analisa estaticamente seu código para encontrar problemas rapidamente e muitos problemas encontrados pelo ESLint podem ser corrigidos automaticamente.



A extensão **JavaScript (ES6) code snippets** contém snippets para a sintaxe ES6 do JavaScript. Por exemplo, o snippet **imp** importa módulos inteiros (ex.: **import fs from 'fs'**);



A extensão **Bracket Pair Colorizer 2** colore pares de parênteses, colchetes ou chaves com a mesma cor, facilitando a visualização de blocos de comando.



A extensão **Prettier - Code formatter** formata e organiza o seu código, que impõe um estilo consistente ao analisar seu código e imprimi-lo novamente com suas próprias regras que levam em consideração o comprimento máximo da linha, agrupando o código quando necessário.



Instalando as extensões **ESLint** e **Prettier**

Passo a Passo na Apostila

Apostila 04.JavaScript

Página 8 a 10

IOS – Instituto de
Oportunidade Social

Sintaxe do JS



Comentários

Comentários são **anotações** inseridas no código fonte com o objetivo de descrever alguma lógica, instrução ou lembrete. Podem ser usados para: lembrar **algo importante** do desenvolvimento, criar secções de **organização** ou **cabeçalho** de função. Comentários são **ignorados** pelo **compilador** na verificação da sintaxe do código.

Comentários em JS

- Comentário de linha, que é iniciado por **//**
// Texto do comentário
- Comentário de bloco, que é iniciado por **/*** e finalizado por ***/**
/* Esse é um comentário tradicional. Ele pode ser dividido em várias linhas */



Importante! Comentários em uma linguagem de programação é muito importante e deve ser usado com consciência. Quando estamos ensinando uma nova linguagem de programação inserimos muitos comentários para que o aluno possa entender o comando ou instrução que está executada, mas no ambiente profissional, você deve

primeiro ver a política de documentação da empresa e códigos exemplos do sistema que você irá trabalhar.

Espaços em branco

O espaço em branco geralmente é insignificante, mas ocasionalmente é necessário usar o espaço em branco para separar sequências de caracteres que, de outra forma, seriam combinadas em um único token. Por exemplo:

```
let num = 3;
```

O espaço em branco entre a palavra reservada `let` e o nome da variável `num` é necessário e não deve ser removido, mas ou outros espaços em branco são opcionais.

Palavras reservadas

Os nomes de variáveis e funções em JS podem conter letras, dígitos ou underline e não podem utilizar as **palavras reservadas**:

abstract	else	instanceof	switch
Boolean	enum	int	Synchronized
break	export	interface	this
byte	extends	long	throw
case	false	native	throws
catch	final	new	transient
char	finally	null	true
Class	float	package	try
Const	function	private	typeof
continue	for	protected	var
debugger	goto	public	void
default	if	return	volatile
delete	implements	short	while
do	imports	static	with
double	in	super	

Números

O JavaScript tem um único tipo de número. Internamente, é representado como **ponto flutuante de 64 bits**, o mesmo que o **double do Java**. Portanto, em JS, não há diferença entre 1 e 1.0, esses números são interpretados como mesmo valor. Esta é uma conveniência significativa, porque problemas de estouro em inteiros curtos são completamente evitados e tudo que você precisa saber sobre um número é que ele é um número. Uma grande classe de erros de tipo numérico é evitada.

Strings

Strings são sequência de caracteres, que no JavaScript, devem ser envolvidas utilizando **aspas simples** ou **aspas duplas**, mesmo se elas contêm zero ou mais caracteres. Todos os exemplos mostrados abaixo são válidos para serem usados na linguagem JavaScript.

```
let nome = "Irmão do Jorel";
```

```
let nome = 'Irmão do Jorel';
```

```
let dados = "";
```

```
let dados = "";
```

```
const frutas = ["maçã", "manga", "pêra"];
```

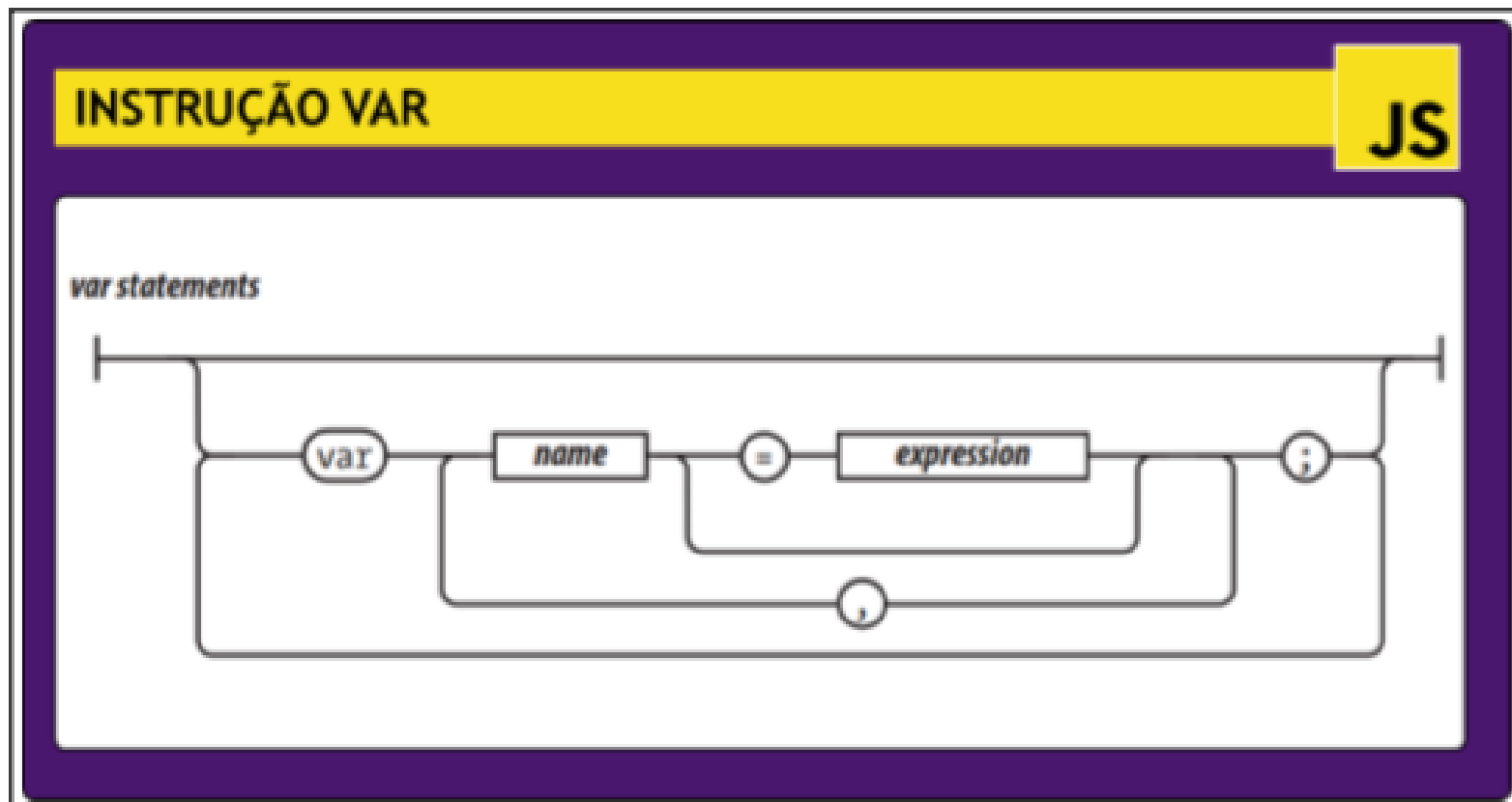
```
const frutas = ['maçã', 'manga', 'pêra'];
```

Se você fez a configuração da extensão **Prettier** como mostrada anteriormente, a configuração **"prettier.singleQuote": true** irá padronizar o uso de aspas simples em toda string no seu código.

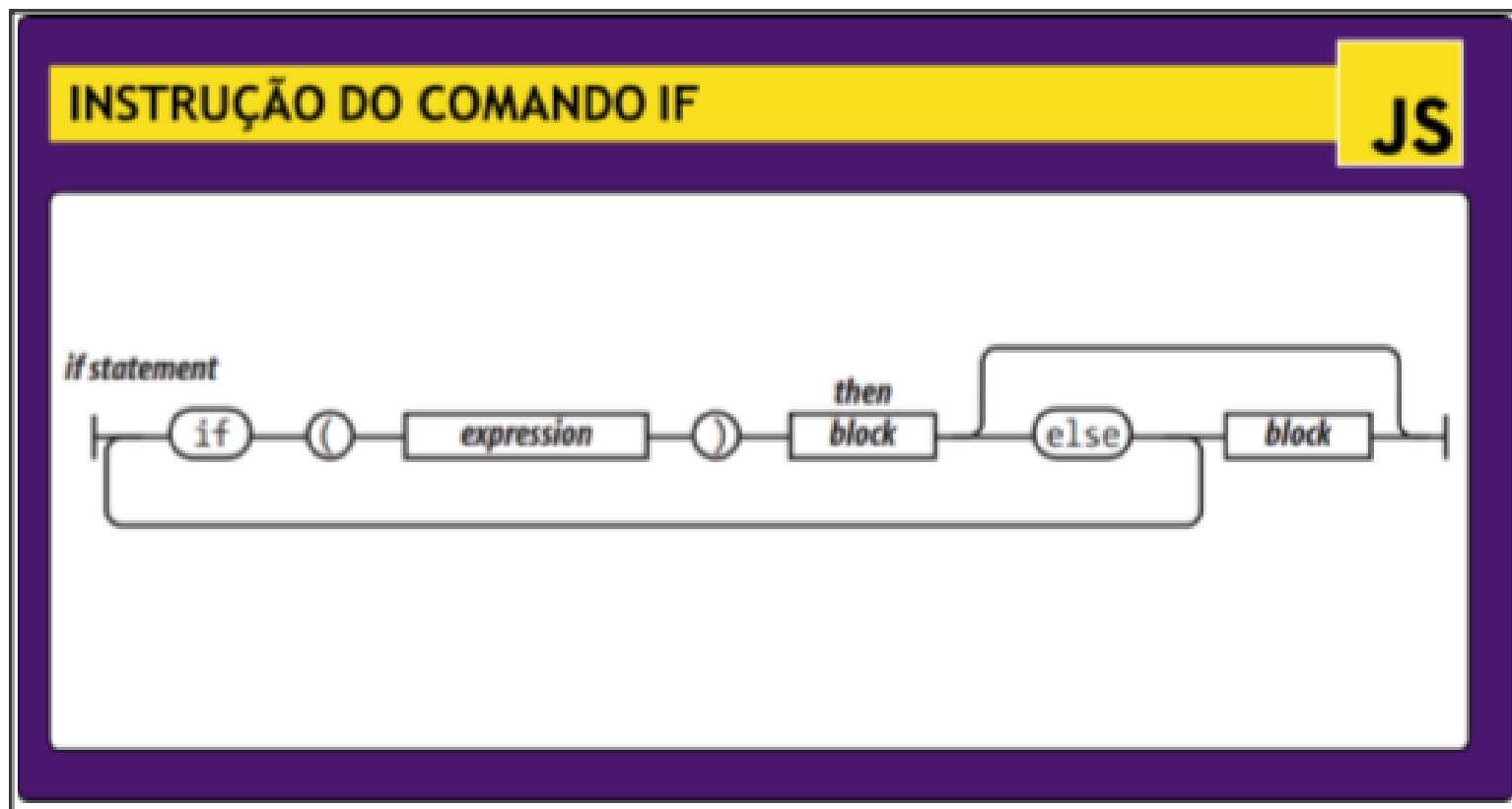
Instruções

Uma unidade de compilação contém um conjunto de instruções executáveis. Em navegadores web, cada marcação **<script>** fornece uma **unidade de compilação** que é **compilada e executada imediatamente**. Desse modo o JS possui uma sintaxe específica para interpretar corretamente as suas instruções. Você deve ficar atento a essas sintaxes para não inserir códigos errados no seu script. A seguir algumas sintaxes usada pelo JavaScript:

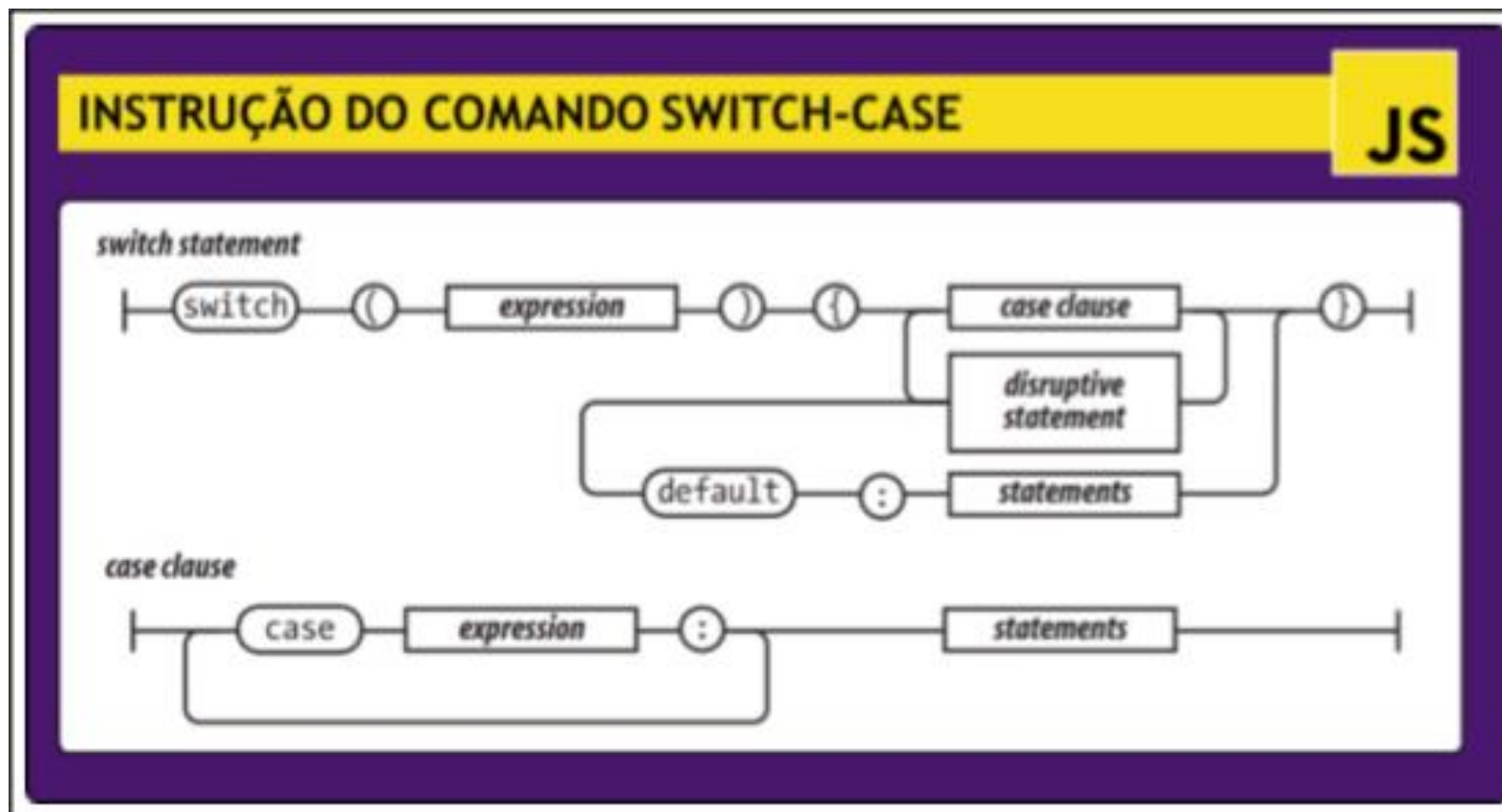
Instrução VAR



Instrução IF



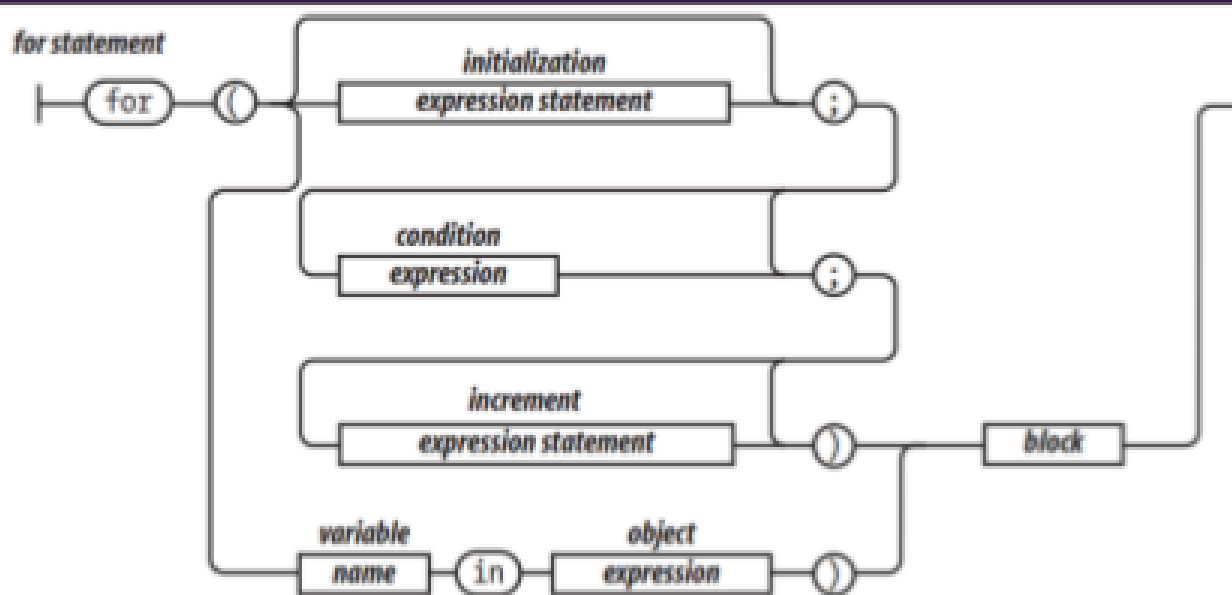
Instrução SWITCH-CASE



Instrução FOR

INSTRUÇÃO DO COMANDO FOR

JS



Instrução WHILE

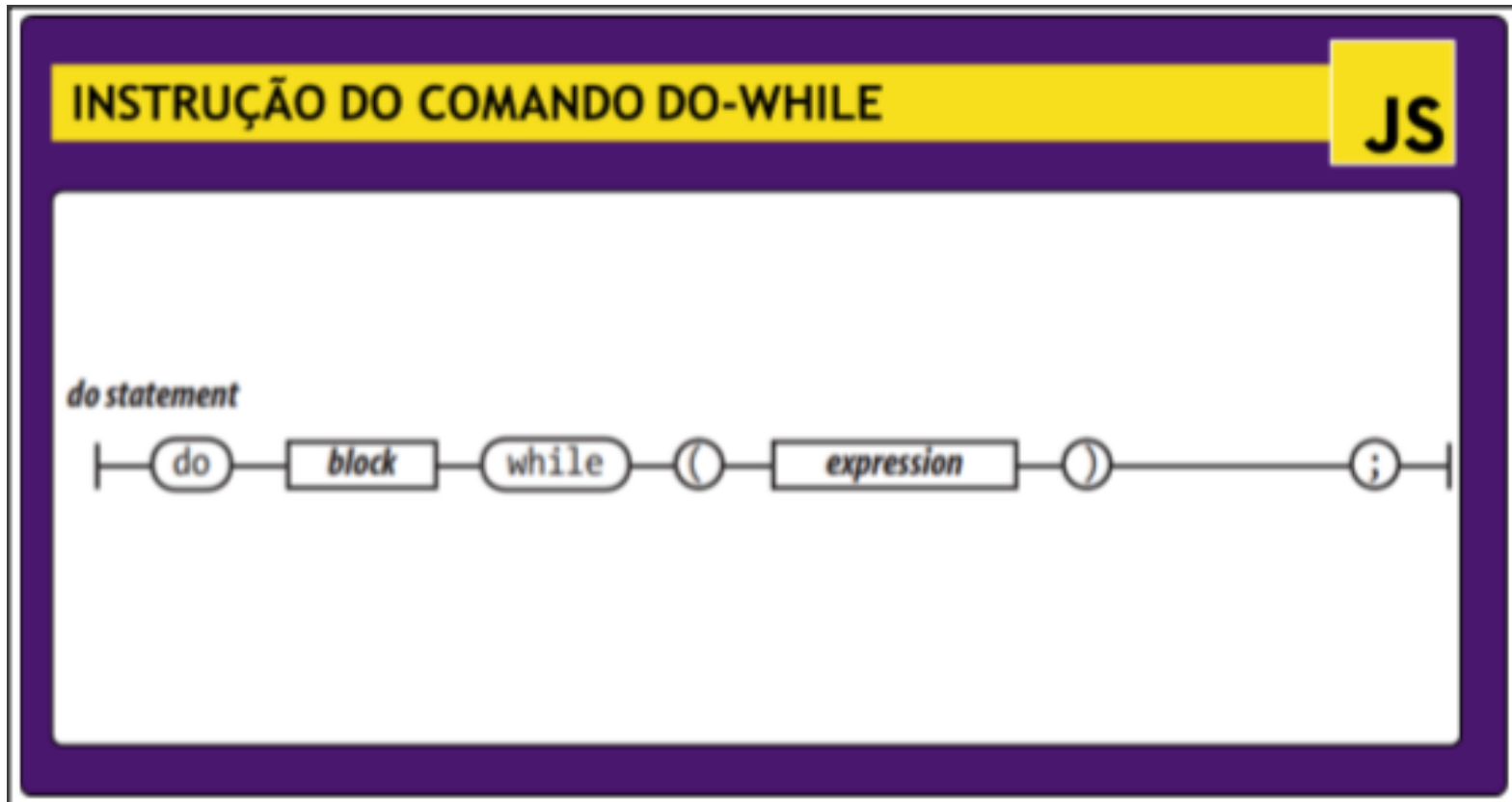
INSTRUÇÃO DO COMANDO WHILE

JS

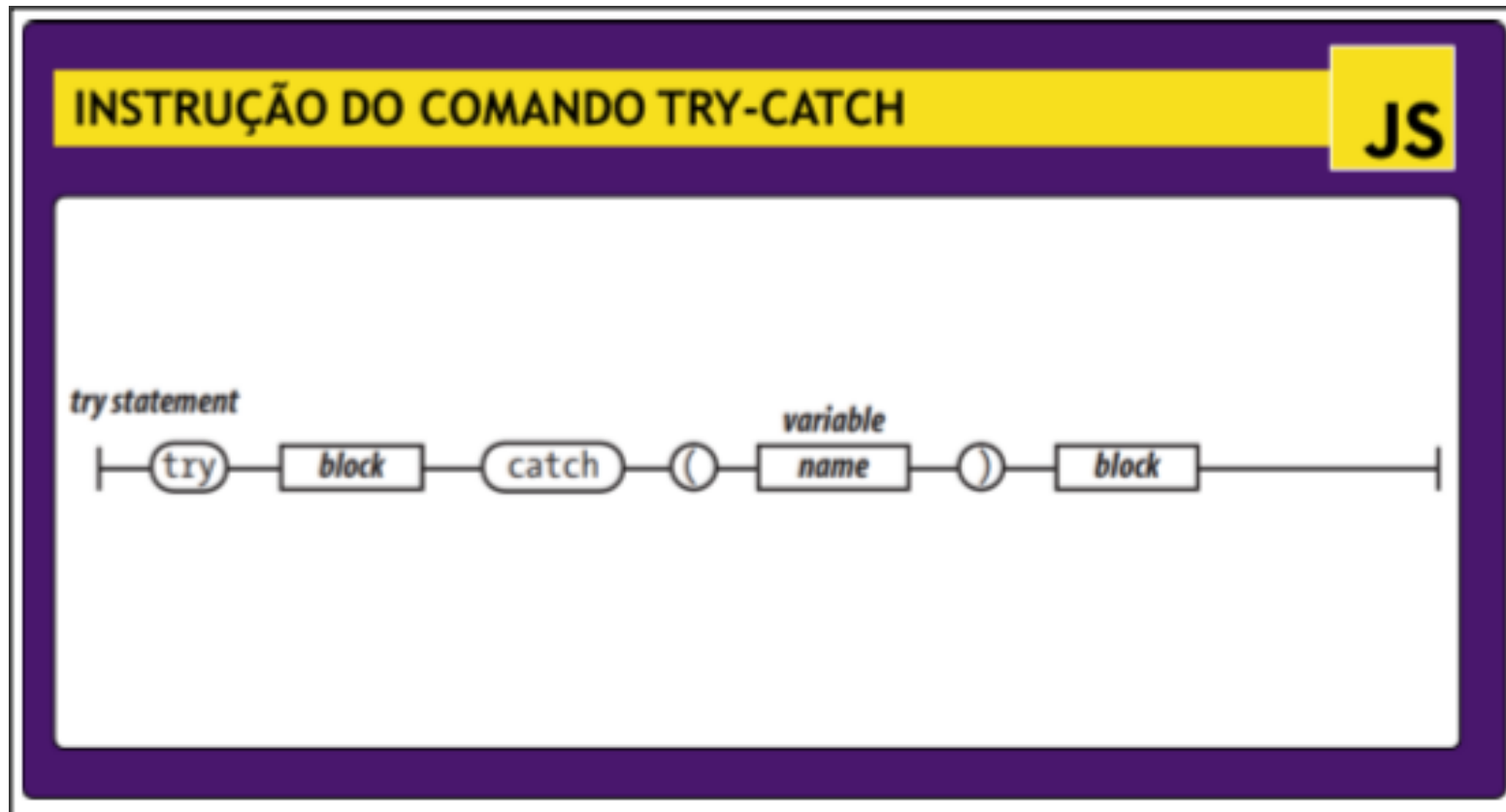
while statement



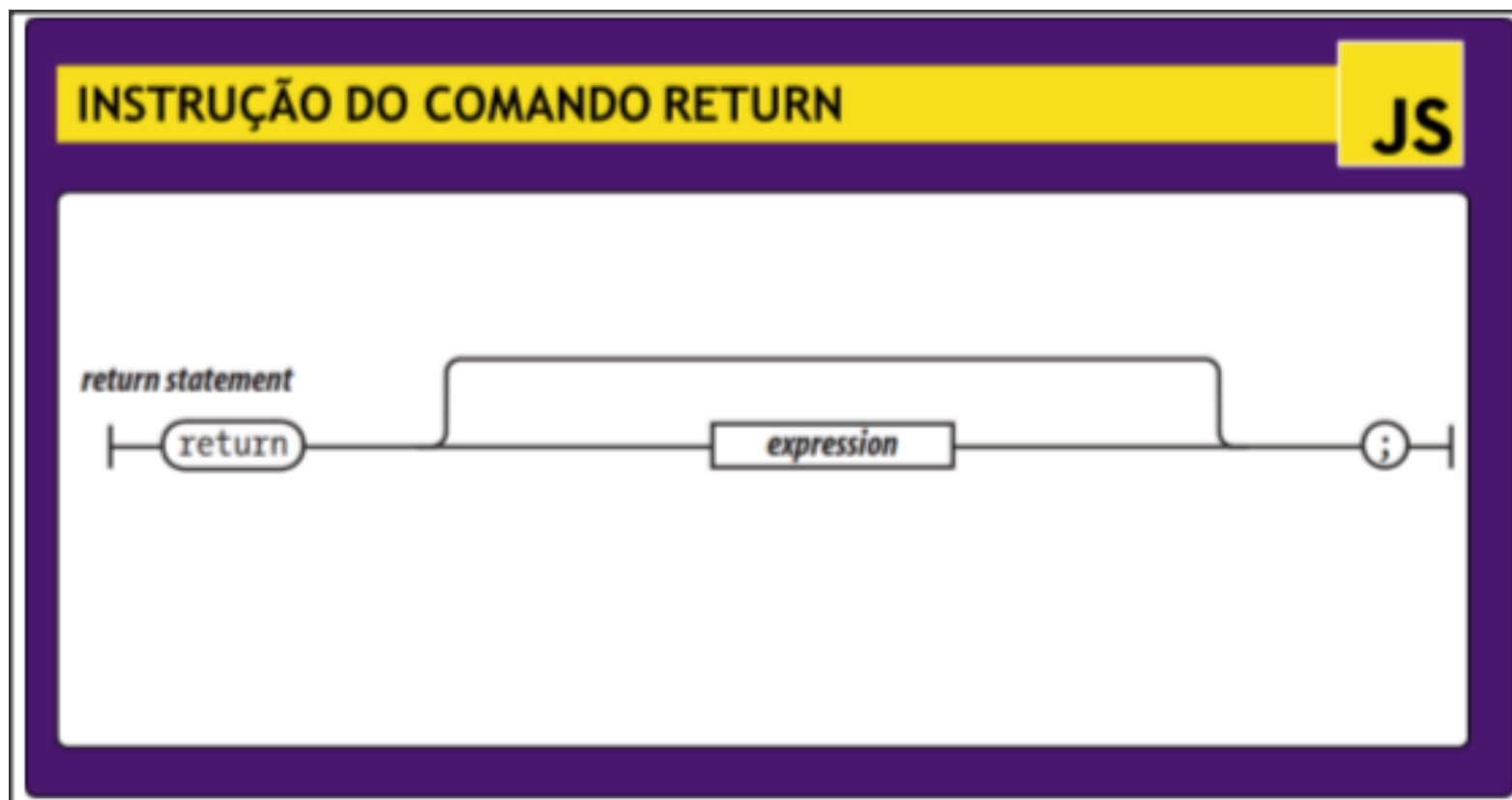
Instrução DO WHILE



Instrução TRY CATCH



Instrução RETURN



Elemento <script>

Um código JavaScript é anexado ao arquivo HTML através da marcação **<script>**. O elemento <script> exige marcações de **abertura e fechamento**, sendo assim, você deve inserir a marcação de abertura <script>, que indica o início do código JavaScript, e, também, deve inserir a marcação de fechamento </script>, que indica o final do código JavaScript. Você pode inserir o código JavaScript no documento HTML de duas formas:

Embutir o código JavaScript **no documento HTML** através do trecho de código:
`<script> // Aqui virá o código JavaScript </script>`

Vincular o arquivo de **script externo** com código JavaScript através do atributo src:
`<script src="arquivo-javascript.js"></script>`

IOS – Instituto de
Oportunidade Social

Vamos Praticar



Apostila de JS

04.JavaScript

Páginas 16 a 17

OBS: Acompanhar o passo a passo com o instrutor e após explicação do **Glossário** de termos que serão utilizados durante o treinamento.

IOS – Instituto de
Oportunidade Social

Exercícios



Instalar extensões: **ESLint** e **Prettier**

Subir arquivo do Vamos Praticar criado durante a aula para o GitHub e enviar o link através do Moodle.