

03-Lógica de Programação com JavaScript

Lógica de Programação com JavaScript

➤ Conhecendo o Ambiente Clássico

➤ Variáveis em JavaScript

➤ Comentários

➤ Operadores Lógicos

➤ Operações Matemática

➤ Funções

➤ Condições com if/else

➤ Condições Ternárias no JavaScript

➤ Condições com Switch

➤ Loops e Iterações

➤ setInterval() e setTimeout()

➤ Trabalhando com a DOM

➤ Manipulando a DOM

➤ Manipulando Elementos Dinamicamente

➤ Manipulando Estilos

➤ Arrays em JavaScript

➤ Arrow function

➤ Destructuring - Rest

➤ Destructuring - Spread

➤ Projeto de Conclusão



Lógica de Programação com JavaScript

➤ Conhecendo o Ambiente Clássico

Conteúdo
HTML



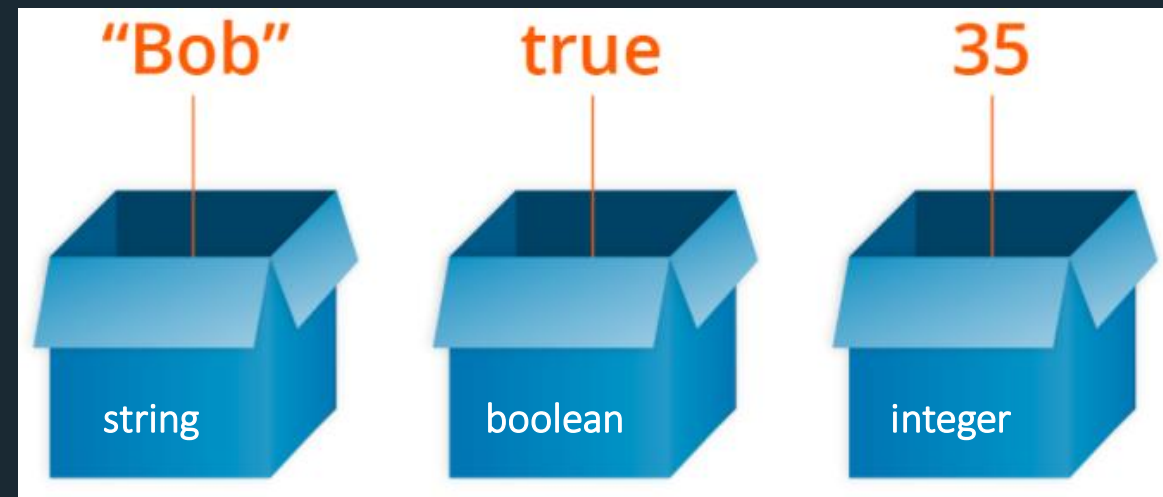
Design
CSS

Interação
JavaScript

Lógica de Programação com JavaScript

➤ Variáveis em JavaScript

JavaScript é uma linguagem de tipagem dinâmica, Isso significa que ela não exige declarações de tipos de dados, pois é capaz de escolher que tipo utilizar dinamicamente para cada variável, podendo alterá-lo durante a execução do programa.



Nota: No JavaScript, todas as instruções em código deve terminar com um ponto e vírgula (;) — seu código pode até funcionar sem o ponto e vírgula em linhas únicas, mas provavelmente não irá funcionar quando estiver escrevendo várias linhas de código juntas. Tente pegar o hábito de sempre incluir o ponto e vírgula.

Lógica de Programação com JavaScript

➤ Variáveis em JavaScript

Tipos de variáveis

//Números

```
var minhaldade = 13;
```

//Strings (cadeias de texto)

```
var meuTexto = 'Olá Mundo';
```

//Booleans (booleanos)

```
var estouVivo = true;
```

```
var estouVoando = false;
```

//Arrays

```
var meuNomeArray = ['Chris', 'Bob', 'Jim'];
```

```
var meuNumeroArray = [10,15,40];
```

//Objetos

```
var cachorro = { nome : 'Totó', raca : 'Dálmata' };
```



Tipos primitivos de dados que podem ser declarados em uma linguagem fortemente tipada.

TIPO	TAMANHO
boolean	1 bit
byte	1 byte
short	2 bytes
char	2 bytes
int	4 bytes
float	4 bytes
long	8 bytes
double	8 bytes

Lógica de Programação com JavaScript

➤ Variáveis em JavaScript

Como foi mostrado no slide anterior com a palavra chave `var` é possível declarar uma variável, porém, em `JavaScript` existem mais duas palavras chaves que podem ser usadas para declarar variáveis são `let` e `const`.

No decorrer do curso até fixar o conceito será usado apenas a palavra chave `var`.

Arquivo para treinar em sala de aula; [Variaveis.html](#)

Lógica de Programação com JavaScript

➤ Comentários

Em JavaScript os comentários podem ser declarados com `//` para uma única linha ou `/* */` para múltiplas linhas.

Os comentários são importantes para colocar informações adicionais em sua linha de código e scripts de forma a explicá-lo para você mesmo em caso de manutenção futura, para você lembrar o que ele faz e, também para as outras pessoas da sua equipe, pois você está escrevendo um código para ser entendido por humanos também.

Por exemplo;

```
// isso é um comentário de uma única linha
```

```
/* isso é um comentário de múltiplas linhas  
isso é um comentário de múltiplas linhas  
isso é um comentário de múltiplas linhas */
```

Lógica de Programação com JavaScript

➤ Operadores Lógicos

Operadores de comparação

Operador	Descrição	Exemplos que retornam verdadeiro
Igual (==)	Retorna verdadeiro caso os operandos sejam iguais.	<pre>3 == var1 "3" == var1 3 == '3'</pre>
Não igual (!=)	Retorna verdadeiro caso os operandos não sejam iguais.	<pre>var1 != 4 var2 != "3"</pre>
Estritamente igual (===)	Retorna verdadeiro caso os operandos sejam iguais e do mesmo tipo. Veja também Object.is e igualdade em JS .	<pre>3 === var1</pre>
Estritamente não igual (!==)	Retorna verdadeiro caso os operandos não sejam iguais e/ou não sejam do mesmo tipo.	<pre>var1 !== "3" 3 !== '3'</pre>
Maior que (>)	Retorna verdadeiro caso o operando da esquerda seja maior que o da direita.	<pre>var2 > var1 "12" > 2</pre>
Maior que ou igual (>=)	Retorna verdadeiro caso o operando da esquerda seja maior ou igual ao da direita.	<pre>var2 >= var1 var1 >= 3</pre>
Menor que (<)	Retorna verdadeiro caso o operando da esquerda seja menor que o da direita.	<pre>var1 < var2 "12" < "2"</pre>
Menor que ou igual (<=)	Retorna verdadeiro caso o operando da esquerda seja menor ou igual ao da direita.	<pre>var1 <= var2 var2 <= 5</pre>

Arquivo para treinar em sala de aula; [operadoresLogicos.html](#)

Lógica de Programação com JavaScript

➤ Operações Matemática

“Operadores aritméticos tem valores numéricos (literais ou variáveis) como seus operadores e retornam um valor numérico único. Os operadores aritméticos padrões são adição (+), subtração (-), multiplicação (*), e divisão (/).”

Mas também temos os operadores;

//Exponenciação (**)

3 ** 2 // 9

//Incremento (++)

// Posfixo

var x = 3;

y = x++; // y = 3, x = 4

// Prefixo

var a = 2;

b = ++a; // a = 3, b = 3

//Decremento (--)

// Posfixo

var x = 3;

y = x--; // y = 3, x = 2

// Prefixo

var a = 2;

b = --a; // a = 1, b = 1

Arquivo para treinar em sala de aula; **operacaoMatematica.html**

Fonte: https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Operators/Arithmetic_Operators

Lógica de Programação com JavaScript

➤ Funções

“Funções são blocos de construção fundamentais em JavaScript. Uma função é um procedimento de JavaScript - um conjunto de instruções que executa uma tarefa ou calcula um valor. Para usar uma função, você deve defini-la em algum lugar no escopo do qual você quiser chamá-la.”

A definição da função (também chamada de declaração de função) consiste no uso da palavra chave `function`, seguida por:

- Nome da Função.
- Lista de argumentos para a função, entre parênteses e separados por vírgulas.
- Declarações JavaScript que definem a função, entre chaves `{ }`.

Expressão de função e Função anônima

Embora a declaração de função acima seja sintaticamente uma declaração, funções também podem ser criadas por uma expressão de função. Tal função pode ser anônima; ela não tem que ter um nome.

Arquivo para treinar em sala de aula;
`funcoes.html`

Fonte: <https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Guide/Fun%C3%A7%C3%B5es>

Lógica de Programação com JavaScript

➤ Exercícios

1. Crie uma função que faça a soma de x e y.
2. Crie uma função que faça a subtração de z e w.
3. Crie uma função que faça a multiplicação de a e b.
4. Crie uma função que faça a divisão de c e d.
5. Crie uma função que mostre o resto da divisão de um numero impar dividido por 2.
6. Crie uma função que retorne seu endereço na seguinte ordem; Rua nomeRua numero x, Barueri , São Paulo - SP.
7. Crie uma função que retorne o tamanho de uma string.
8. Crie uma função que recebe um string minúscula e retorne maiúscula.
9. Crie uma função que recebe um string maiúscula e retorne minúscula.
10. Para que serve o CSS?
11. Para que serve o HTML?
12. Para que serve JavaScript em uma página da web?
13. O que é case-sensitive?
14. Instale as extensões Live Server, Bracket Pair Colorizer, Live Share, Material Icon Theme e Color Highlight em seu VSCode.
15. Instale a fonte Fira Code em seu Windows.

Lógica de Programação com JavaScript

➤ Condições com if/else

Declarações condicionais:

Uma declaração condicional é um conjunto de comandos que são executados caso uma condição especificada seja verdadeira. O JavaScript suporta duas declarações condicionais: if...else e switch, mas nesta aula iremos ver somente if/else.

Declaração if...else:

Use a declaração if para executar alguma declaração caso a condição lógica for verdadeira. Use a cláusula opcional else para executar alguma declaração caso a condição lógica for falsa. Uma declaração if é declarada da seguinte maneira:

```
if (condicao) {  
    declaracao_1;  
} else {  
    declaracao_2;  
}
```

Arquivo para treinar em sala de aula;
IFcondicional.html
IFcondicional2.html
operadoresLogicos.html

Lógica de Programação com JavaScript

➤ Condições Ternárias no JavaScript

```
//Operadores lógicos AND, OR, NOT
```

```
/*Em diversas linguagens esses operadores são representados  
pelos seguintes caracteres AND pelo &&, OR pelo ||, NOT pelo !=
```

```
no  
É possível também armazenar o valor da operação em uma variável
```

```
formato boolean (veremos na próxima aula).
```

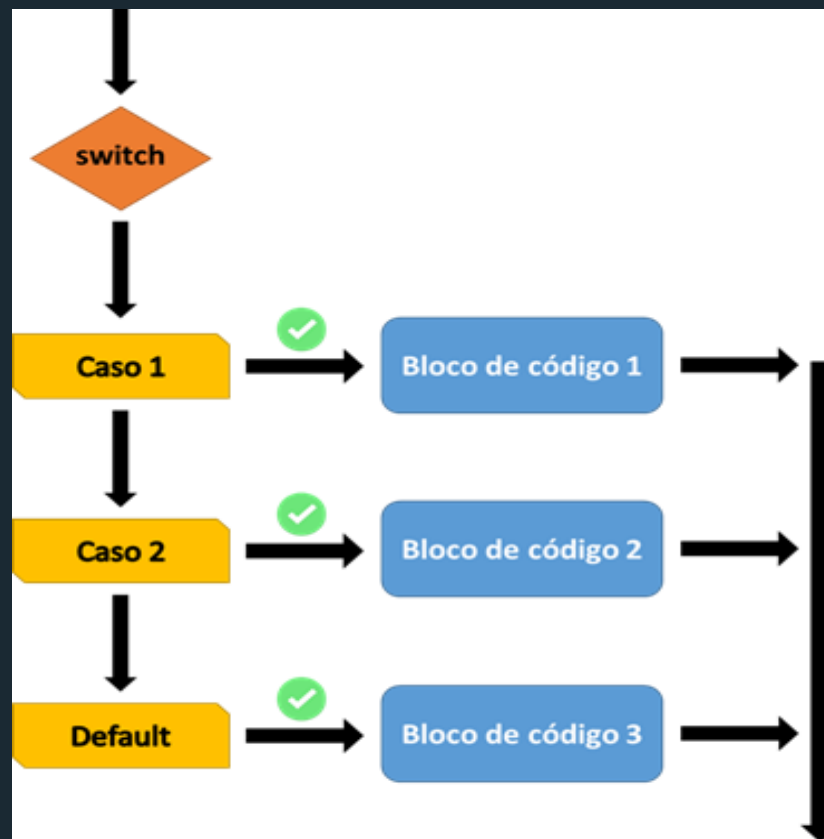
```
Para mais comparações veja a página 8.
```

```
*/
```

Lógica de Programação com JavaScript

➤ Condições com Switch

```
var Animal = 'Girafa';  
switch (Animal) {  
  case 'Vaca':  
  case 'Girafa':  
  case 'Cachorro':  
  case 'Porco':  
    alert('Esse animal irá para Arca de Noé');  
    break;  
  case 'Dinossauro':  
  default:  
    alert('Esse animal não vai.');
```



Arquivo para treinar em sala de aula;
[SwitchCondicional.html](#)

Fonte:

<https://www.devmedia.com.br/javascript-switch/39761>

<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Statements/switch>

https://www.w3schools.com/js/js_switch.asp

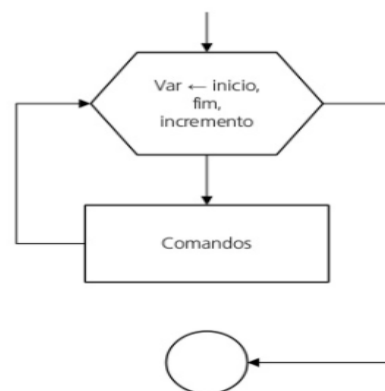
Lógica de Programação com JavaScript

➤ Loops e Iterações

JavaScript – Estruturas Estrutura de Repetição

for

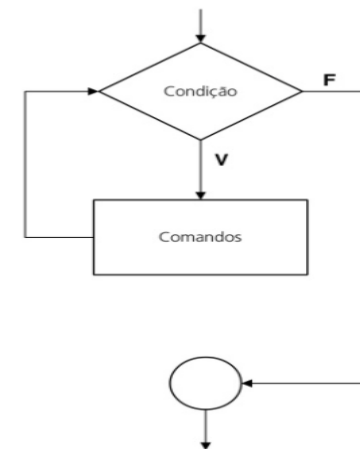
```
for (var i=0; i<3; i++){  
    //comandos  
}
```



JavaScript – Estruturas Estrutura de Repetição

while

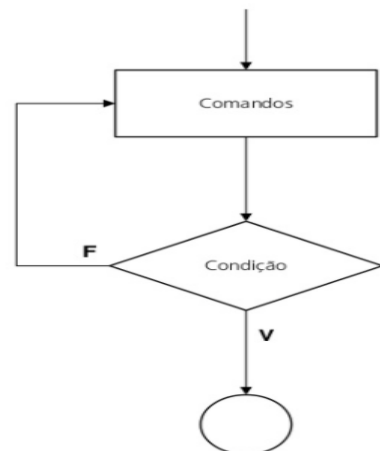
```
while (i<3){  
    //comandos  
    i++  
}
```



JavaScript – Estruturas Estrutura de Repetição

do while

```
do{  
    //comandos  
    i++  
} while (i<3);
```



Arquivo para treinar em sala de aula;
[LoopDeRepeticao.html](#)

Lógica de Programação com JavaScript

➤ Exercícios

1. Qual dia da semana é hoje?

Use o `Date().getDay()` junto com `Switch` para pegar a data do sistema e informar o dia da semana quando a condição for verdadeira.

2. Criar um código com um loop `for` e exiba o resultado.

3. Criar um código com um loop `while` e exiba o resultado.

4. Criar um código com um loop `do while` e exiba o resultado.

5. O que os loops `for...of` e `for...in` fazem faça uma exemplo de cada.

Fonte para pesquisa:

https://www.w3schools.com/js/js_loop_for.asp

https://www.w3schools.com/js/js_loop_while.asp

<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Statements/do...while>

<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Statements/while>

Lógica de Programação com JavaScript

➤ setInterval() e setTimeout()

setTimeout()

Executa um bloco específico uma vez depois de um determinado tempo.

setInterval()

Executa um bloco específico repetidamente com um intervalo fixo entre cada chamada.

Um intervalo dentro do Javascript, é uma função que executa dentro de um intervalo predefinido por você programador.

No exemplo a função irá imprimir de forma recorrente a cada 5 segundos, nesse caso é feito uma chamada para que a função seja executada quando o código rodar por que ela foi referenciada dentro de um código e o resultado dela seve com input para uma outra execução feita pelo código.

Obs: O tempo sempre é em milissegundos

Arquivo para treinar em sala de aula;
timeOut.html

Fonte: https://www.w3schools.com/jsref/met_win_setinterval.asp
https://developer.mozilla.org/pt-BR/docs/Learn/JavaScript/Asynchronous/Timeouts_and_intervals

Lógica de Programação com JavaScript

➤ Exercícios

Nesta aula iremos conhecer e aprender algumas formas de manipular os elementos na DOM.

“O Document Object Model (DOM) é uma interface de programação para os documentos HTML e XML. Representa a página de forma que os programas possam alterar a estrutura do documento, alterar o estilo e conteúdo. O DOM representa o documento com nós e objetos, dessa forma, as linguagens de programação podem se conectar à página.”

O que é a DOM: <https://www.youtube.com/watch?v=HOv9CqqAZk0>

Arquivo para treinar em sala de aula;
TrabalhandoaDOM.html

Fonte: [https://developer.mozilla.org/pt-BR/docs/DOM/Referencia do DOM/Introdu%C3%A7%C3%A3o](https://developer.mozilla.org/pt-BR/docs/DOM/Referencia_do_DOM/Introdu%C3%A7%C3%A3o)

Lógica de Programação com JavaScript

➤ Trabalhando com a DOM

Assista o vídeo abaixo e baseado em todo conteúdo já visto no curso, crie um formulário básico de sua preferência.

<https://www.youtube.com/watch?v=UftSB4DaRU4>

Fonte: [https://developer.mozilla.org/pt-BR/docs/DOM/Referencia do DOM/Introdu%C3%A7%C3%A3o](https://developer.mozilla.org/pt-BR/docs/DOM/Referencia_do_DOM/Introdu%C3%A7%C3%A3o)

Lógica de Programação com JavaScript

➤ Manipulando a DOM

Como vimos na aula anterior a DOM basicamente é a árvore de elementos dos HTML, ela é composta pelos elementos que são distribuídos na página como botões, caixa de texto etc.

Arquivo para treinar em sala de aula;
ManipulandoDOM.html

Fonte: [https://developer.mozilla.org/pt-BR/docs/DOM/Referencia do DOM/Introdu%C3%A7%C3%A3o](https://developer.mozilla.org/pt-BR/docs/DOM/Referencia_do_DOM/Introdu%C3%A7%C3%A3o)

Lógica de Programação com JavaScript

➤ Manipulando Elementos Dinamicamente

Criar elementos dinamicamente no corpo da página html com Javascript

Neste exemplo vamos criar um link e um input do tipo numérico, este é um exemplo básico que nos possibilita ver de forma prática o funcionamento do JavaScript e, que além de manipular os elementos já existentes na DOM também é possível cria-los de forma dinâmica.

Arquivo para treinar em sala de aula;
TrabalhandoElementos.html

Fonte: [https://developer.mozilla.org/pt-BR/docs/DOM/Referencia do DOM/Introdu%C3%A7%C3%A3o](https://developer.mozilla.org/pt-BR/docs/DOM/Referencia_do_DOM/Introdu%C3%A7%C3%A3o)

Lógica de Programação com JavaScript

➤ Exercícios

Exercício Anterior:

Assista o vídeo abaixo e baseado em todo conteúdo já visto no curso, crie um formulário básico de sua preferência.

<https://www.youtube.com/watch?v=UftSB4DaRU4>

Exercício Atual:

Adicione o que você aprendeu nesta ao exercício anterior

Fonte: [https://developer.mozilla.org/pt-BR/docs/DOM/Referencia do DOM/Introdu%C3%A7%C3%A3o](https://developer.mozilla.org/pt-BR/docs/DOM/Referencia_do_DOM/Introdu%C3%A7%C3%A3o)

Lógica de Programação com JavaScript

➤ Manipulando Estilos

Nesta aula iremos criar uma elemento visual na DOM

Para este exemplo iremos criar um elemento visual com a finalidade de mostra que também é possível criar elemento com javascript puro, porem é mais recomendado o uso do **css** par esta finalidade.

Aula complementar ao tópico anterior.

Arquivo para treinar em sala de aula;
ManipulandoEstilos.html

Fonte: [https://developer.mozilla.org/pt-BR/docs/DOM/Referencia do DOM/Introdu%C3%A7%C3%A3o](https://developer.mozilla.org/pt-BR/docs/DOM/Referencia_do_DOM/Introdu%C3%A7%C3%A3o)

Lógica de Programação com JavaScript

➤ Arrays em JavaScript

Interação com arrays.

Nesta aula iremos aprender manipulações básicas com array usando as funções Map, reduce, filter, find e includes.

Arquivo para treinar em sala de aula;
TrabalhanArrays.html

Fonte: <https://programadriano.medium.com/javascript-conhecendo-map-filter-e-reduce-ce072d8f0ec5>

Lógica de Programação com JavaScript

➤ Arrow functions

Arrow functions

São basicamente funções anônimas com algumas funcionalidades especiais. Nesta aula iremos aprender com usar Arrow functions.

```
()=>{ }
```

Arquivo para treinar em sala de aula;
arrowFunction

Fonte: https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Functions/Arrow_functions