



@tiacademybrasil

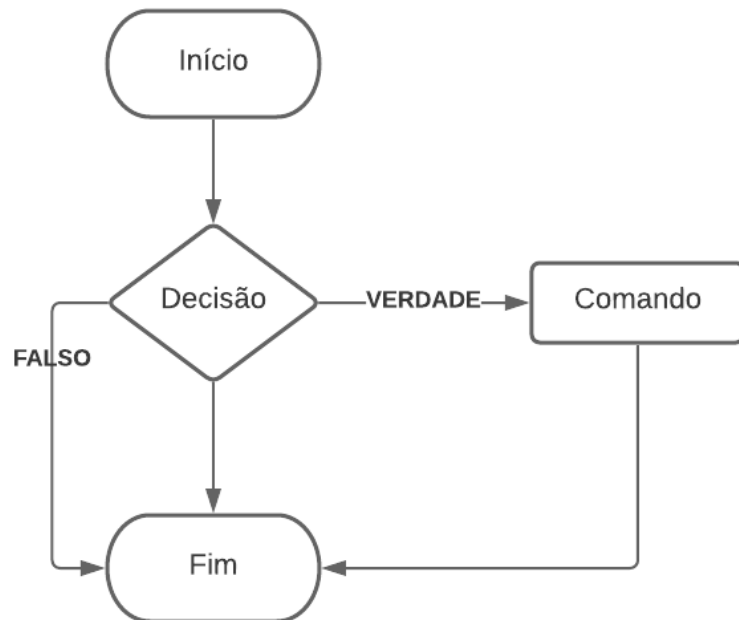
JAVASCRIPT

AULA 3 | Prof. Marcelo

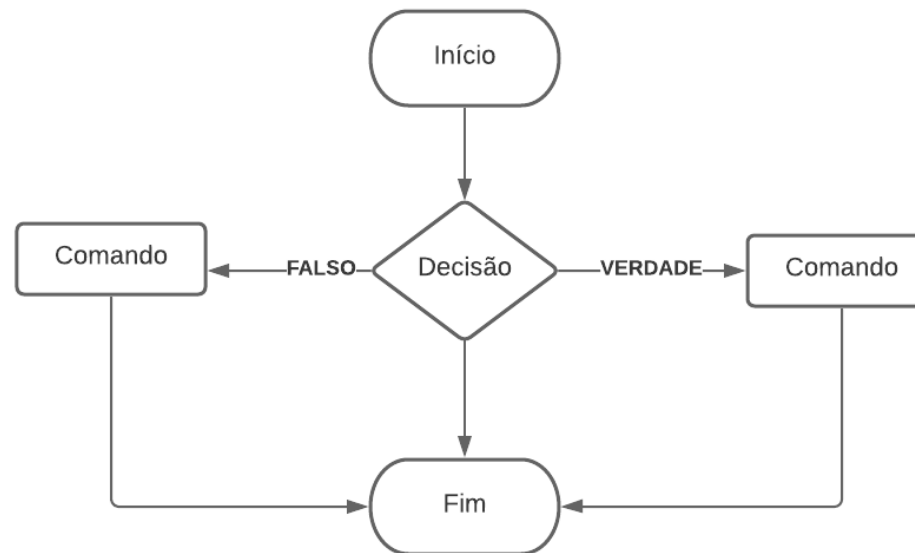
- Estrutura de controle (condicionantes)
 - Operadores de comparação
- Funções
- var, let e const
- Objetos*

Condicionantes

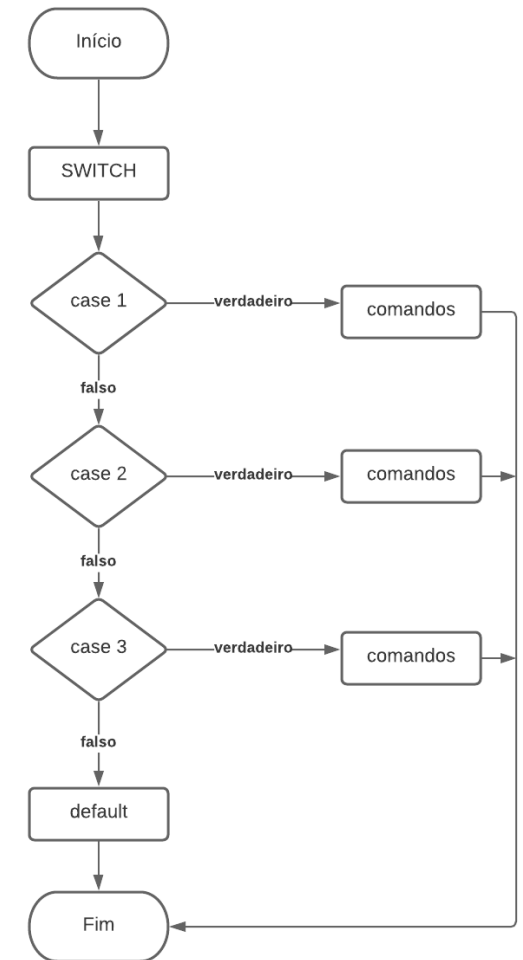
FLUXO DE DECISÃO



if



if/else



switch

Operadores

Na maioria dos casos os operadores lógicos e de comparação são usados em um fluxo de decisão.

Hoje

- Objetos.
- DOM
- Arrays

Objetos - Literais

Um **objeto literal** é uma **lista separada com vírgulas**. Entre o nome da propriedade e seu valor, que pode ser um tipo de dado ou uma função separados **por dois-pontos**, colocados entre chaves. Ex:

```
var produto1 = {  
    descricao : "Biscoito",  
    fornecedor: "Bauduco"  
}
```

Objetos - Operador new

O operador **new** cria e **inicializa um novo objeto**. A palavra-chave **new** deve ser seguida de uma **chamada de função**.

Uma **função usada dessa maneira é chamada de construtora** e serve para inicializar um objeto recém-criado. **JavaScript básica contém construtoras internas para tipos nativos**.

Ex.:
var obj = new Object()
var arr = new Array()
var dta = new Data()

Objetos - Operador new

Criando o objeto Produto com o operador new. Exemplo:

```
var obj  = new Object()  
var arr  = new Array()  
var dta  = new Data()
```


Objetos - Operador new

Criando um objeto Pessoa por meio de uma função, O **operador new** cria uma instancia de um **tipo de objeto** definido pelo usuário ou de um dos tipos :

```
function Pessoa(nome, sobrenome, idade){  
  
    this.nome;  
    this.sobrenome;  
    this.idade;  
  
}  
  
var p1 = new Pessoa("Fulano", "Da Silva", "27");
```

A **palavra reservada this** refere-se ao objeto atual em que o código está sendo escrito nesse caso o this se refere ao objeto Pessoa.

DOM

Criando um objeto Pessoa por meio de uma função, O **operador new** cria uma instancia de um **tipo de objeto** definido pelo usuário ou de um dos tipos :

```
function Pessoa(nome, sobrenome, idade){  
  
    this.nome;  
    this.sobrenome;  
    this.idade;  
  
}  
  
var p1 = new Pessoa("Fulano", "Da Silva", "27");
```

A **palavra reservada this** refere-se ao objeto atual em que o código está sendo escrito nesse caso o this se refere ao objeto Pessoa.

DOM - Document Object Model

DOM - Document Object Model

Document Object Model, ou DOM, é uma estrutura interna fundamental para representar e manipular o conteúdo de documentos HTML.

```
<html>
  <head>
    <title>Document</title>
  </head>
  <body>
    <h1>Apenas um HTML</h1>
    <p>Isto é um <i>exemplo</i> </p>.
  </body>
</html>
```

Em uma tradução livre DOM seria “Modelo de Objeto de Documento”

JS

```
<html>
```

```
  <head>
```

```
    <title>Document</title>
```

```
  </head>
```

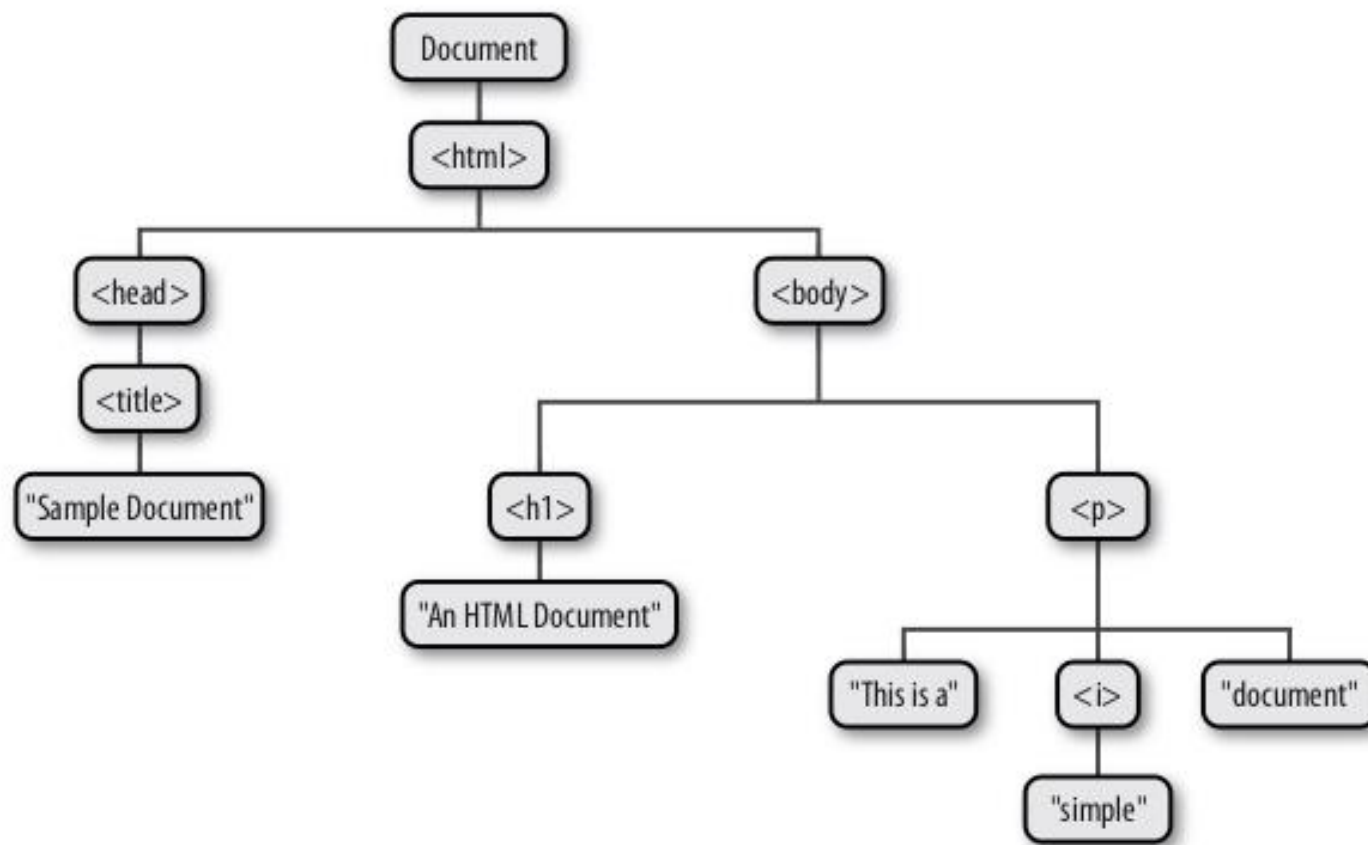
```
<body>
```

```
  <h1>Apenas um HTML</h1>
```

```
  <p>Isto é um <i>exemplo</i> </p>.
```

```
</body>
```

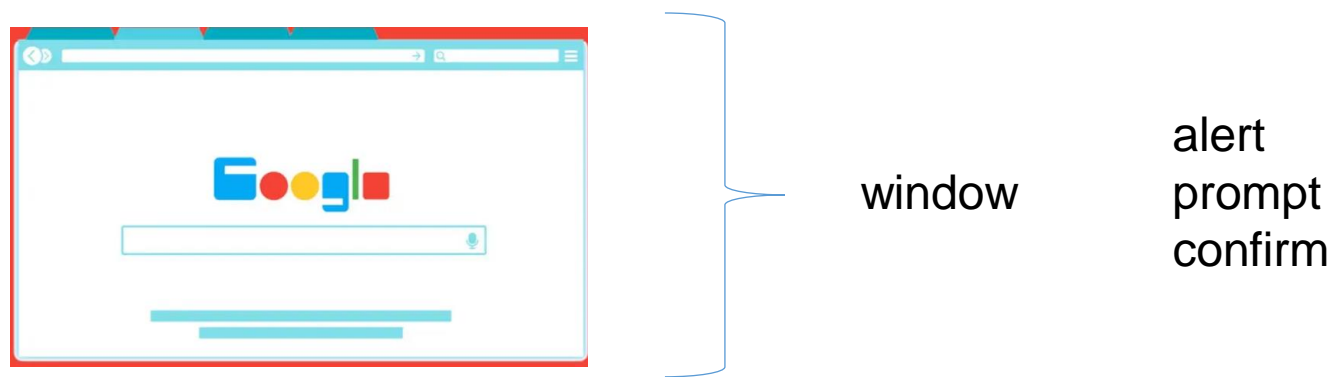
```
</html>
```



OBJETO WINDOW

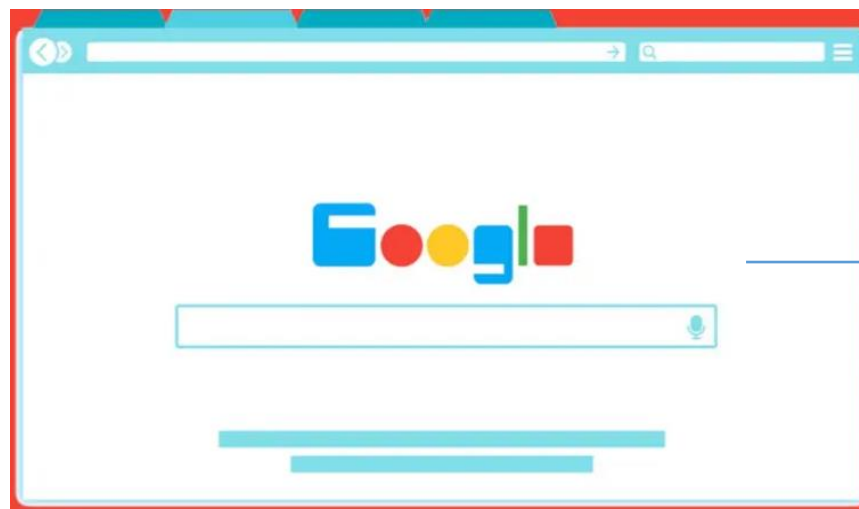
O principal **objeto carregado dinamicamente** é o WINDOW (“window”) . Este objeto é o principal ponto de entrada para todos os recursos de Javascript do lado do cliente.

O objeto window, como o nome diz representa uma janela do navegador Web e é referenciado através do identificador window.



PROPRIEDADE DOCUMENT

A **DOCUMENT** “documento” é uma propriedade do objeto *window* e refere-se a estrutura **HTML** de uma página. Esta estrutura pode ser manipulada e ou acessada por meio de outros métodos.



document

document.write(string) é o mesmo que window.document.write(string)

ALGUMAS PROPRIEDADES DOCUMENTs

document.**location.href**: direciona para outra url

document.**getElementById**(el) - Retorna o elemento DOM/html pelo id

EVENTOS

Eventos no DOM (*Dom Events*) notificam o código as ações que ocorrem durante a interação de navegação do usuário. O **addEventListener()** irá registra uma única “espera” de evento em um único alvo.

ALGUMAS PROPRIEDADES DOCUMENTs

document.**location.href**: direciona para outra url

document.**getElementById**(el)

document.**querySelector**(el)

ADDEVENTLISTENER “LISTA DE EVENTOS”

O `addEventListener()` irá registra uma única “espera” de evento em um único alvo.

ALGUMAS PROPRIEDADES DOCUMENTs

document.**location.href**: direciona para outra url

document.**getElementById**(el)

document.**querySelector**(el)

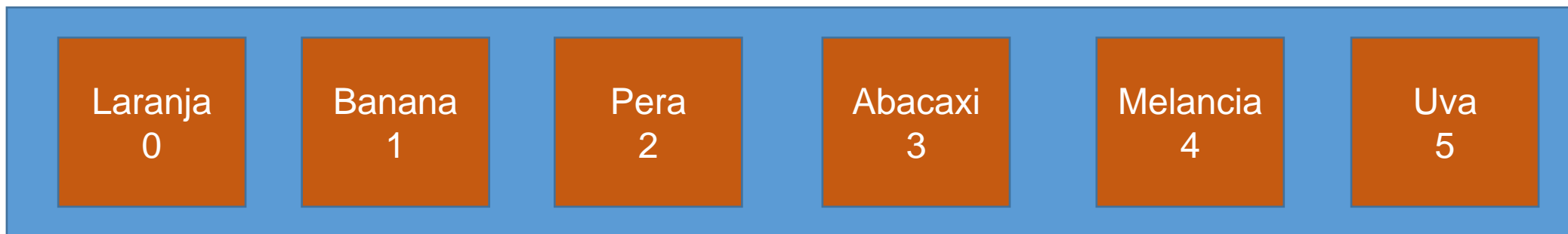
ADDEVENTLISTENER “LISTA DE EVENTOS”

O `addEventListener()` irá registra uma única “espera” de evento em um único alvo.

ARRAYS

Um **array** é um **conjunto ordenado de valores**. Cada valor é chamado de elemento e cada elemento tem uma posição numérica no array, conhecida como *índice*.

Array Frutas



```
var frutas = new Array("Laranja", "Banana", "Pera", "Abacaxi", "Melancia", "Uva");
```

```
var f = ["Laranja", "Banana", "Pera", "Abacaxi", "Melancia", "Uva"];
```

ARRAYS - comprimento de um array

Um array quando criado e indexado seus elementos irá possuir um “comprimento” este comprimento será recuperado pela propriedade length.

```
console.log(frutas.length)
```

ARRAYS - alguns métodos

Método	Descrição	Exemplo
concat()	Junta os elementos de dois ou mais arrays e retorna uma cópia com os elementos juntos	<pre>var ar1 = ["Carro", "Moto"]; var ar2 = ["Navio", "Patins", "Skate"]; var transp = ar1.concat(ar2);</pre>
indexOf()	Procura por um elemento específico no array e retorna a sua posição	<pre>var cores = ["Verde", "Azul", "Amarelo"]; var pos = cores.indexOf("Azul"); //Retorna 1</pre>
join()	Junta todos os elementos de um array em uma string	<pre>var cores = ["Verde", "Azul", "Amarelo"]; var paleta = cores.join();</pre>
push()	Insere um novo elemento no final do array	<pre>var cores = ["Verde", "Azul", "Amarelo"]; cores.push("Vermelho");</pre>
pop()	Remove o último elemento do array	<pre>var cores = ["Verde", "Azul", "Amarelo"]; cores.pop();</pre>
reverse()	Inverte a ordem dos elementos do array	<pre>cores.reverse();</pre>
shift()	Remove o primeiro elemento do array	<pre>var cores = ["Verde", "Azul", "Amarelo"]; cores.shift();</pre>
sort()	Ordena os elementos do array em ordem crescente	<pre>cores.sort();</pre>
toString()	Converte um array em string e retorna essa string	<pre>var cores = ["Verde", "Azul", "Amarelo"]; cores.toString();</pre>
unshift()	Insere um novo elemento no início do array	<pre>var cores = ["Verde", "Azul", "Amarelo"]; cores.unshift("Vermelho");</pre>
splice()	Corta o array em um ponto indicado.	<pre>var num = [0,1,2,3,4,5,6,7,8,9]; num.splice(2,4); //Remove 4 elementos a partir do índice 2 //Resultado: [0,1,2,3,6,7,8,9]</pre>