

## Javascript

Javascript é a linguagem de programação usada para adicionar interatividade ao seu site.

Por exemplo: ações quando botões são pressionados, validações em formulários, mudanças dinâmicas na página, animações e muito mais.

## Tipos de Saída

a maneira que o javascript tem para exibir informações, dados...

```
document.getElementById( " ").innerHTML="  ";
```

```
document.write('  ');
```

```
console.log("  "); para o console do navegador
```

```
window.alert( ); caixa de alerta
```

## Sintaxe JavaScript

é o conjunto de regras, como JavaScript programas são construídos.

### Programas JavaScript

Um programa de computador é uma lista de "instruções" para ser "executada" pelo computador. Em uma linguagem de programação, estas instruções de programa são chamados de declarações. JavaScript é uma linguagem de programação. Instruções JavaScript são separados por ponto e vírgula.

## Comentarios

```
//
```

```
/* */
```

## Variaveis

Variáveis são recipientes onde podemos armazenar informações que podem variar, ou seja, podem ter qualquer tipo de valor.

As regras gerais para a construção de nomes para as variáveis (identificadores únicos) são:

Os nomes podem conter letras, números, sublinhados e sinais de dólar.

Os nomes devem começar com uma letra

Nomes também pode começar com US \$ e \_ (mas não vamos usá-lo neste tutorial)

Os nomes são sensíveis caso (y e Y são variáveis diferentes)

As palavras reservadas (como palavras-chave de JavaScript) não podem ser usadas como nomes

No javascript existe tres palavras-chaves para declarar variaveis:

- var
- let
- const

## Concatenação

Ex:

```
var cadeia1 = "Bom dia";
```

```
var cadeia2 = "a todos";
```

```
var cadeia3 = cadeia1 + " " + cadeia2;
```

## Operadores

Os operadores javascript são usados para atribuir valores, comparar valores, executar operações aritmeticas...

- operadores aritmeticos(matematicos)
- operadores de atribuição
- operadores de sequencia
- operadores de comparação
- operador condicional (ternario)
- operadores logicos

#### Aritméticos JavaScript

**+** Adição

**-** Subtração

**\*** Multiplicação

**/** Divisão

**%** Módulo

**++** Incremento

**--** diminuição

ex:

```
var valor1, valor2, total;
```

```
valor1 = 5;
```

```
valor2 = 4;
```

```
total = valor1 + valor2;
```

operadores de atribuição

=      x = y      x = y

**+=**   x += y      x = x + y

**-=**   x -= y      x = x - y

**\*=**   x \*= y      x = x \* y

**/=**   x /= y      x = x / y

**%=**   x %= y      x = x % y

operadores de sequencia (concatenação)

Ex:

let text1 = "John";

let text2 = "Doe";

let text3 = text1 + " " + text2;

operadores de comparação

comparar valores

**==**   igual a

**===**   o mesmo valor e igual tipo

**!=**   não igual

!== não igual valor ou não igual tipo

> Maior que

< menor que

>= maior do que ou igual a

<= menor que ou igual a

? operador ternário

operador condicional (ternario)

?

operadores logicos

&& e

|| ou

! não (negação)

## Funções

Uma função javascript é um bloco de código projetado para executar uma tarefa específica

É como uma pequena fábrica onde você faz uma entrada e ele te dá uma saída.

Pode ser encarado como mini-programas projetados para fazer uma

tarefa que vai contribuir para todo código.

uma função javascript é executada quando algo invoca (chama-a)

## FUNÇÕES



```
function soma (valor1, valor2){  
    return valor1 + valor2;  
}
```

ex:

```
function soma(valor1, valor2){  
    return valor1 + valor2;  
}
```

### Objetos

objetos são basicamente variáveis com muitos valores dentro.

Ex: const carro = {marca: "ford", modelo: "ka", ano: 2015}

Os valores dentro de um objeto são chamados propriedades

Objetos também podem ter métodos. Um método é uma função colocadas dentro de uma propriedade.

## Eventos

Eventos são ações disparadas pela interação dos usuários na página. É o correto manejo desses eventos que tornam as páginas interativas e dinâmicas.

Existem muitos eventos. Segue os mais utilizados:

onclick: disparado quando recebe um click

ondblclick: disparado quando clique duplo

onmouseover: disparado quando o mouse está sobre

onmouseout: disparado quando o mouse é movido para fora do elemento.

onmousemove: disparado quando o mouse é movido no elemento

onmousedown: disparado quando o clique do botão foi pressionado.

onmouseup: disparado quando o clique do botão é liberado

onfocus: disparado quando o elemento recebe o foco. Valido para input,

onchange: disparado quando existe uma mudança no conteúdo. "ao mudar"

onblur: disparado quando o elemento perde o foco

onkeydown: disparado quando uma tecla é pressionada

onkeypress: disparado quando uma tecla é pressionada

onkeyup: disparado quando uma tecla é solta sobre um elemento

onload: disparado quando a página terminou de ser carregada. body

onresize: disparado quando há um redimensionamento da janela

Ex:

```
<button onclick="eventoClick()">clique aqui</button>
```

```
function eventoClick(){  
    alert('Acionou um evento de click');  
}
```

## Arrays

os arrays javascript são usados para armazenar varios valores em uma unica variavel. Diferente dos objetos, que funcionam com propriedades ou nomes que você dá para os itens dentro dele, os arrays não possuem propriedades. O item dentro dele é encontrado pela posição.

Imagine um array como uma lista de itens, com controle, por ordem de posição dentro dele.

Ex: const lista = ["arroz", "feijão", "macarrão", "leite"];

a lista[0] (lista na posição 0) vai conter o valor "arroz".

a lista[1] (lista na posição 1) vai conter o valor "feijão"

## Métodos Para Arrays (Matrizes)

- metodo join: O "join()" puxa elementos de um array e lista no formato de string
- metodo pop remover o ultimo item
- metodo push adicionar um item no fim
- metodo shift remover primeiro item
- metodo unshift adicionar um item no começo
- metodo delete
- metodo splice :No "splice()" podemos escolher um índice inicial e final para substituímos valores entre eles
- metodo concat: método "concat()", que serve para concatenar dois arrays
- metodo slice: O "slice()" extrai uma sessão de um array e retorna um novo array



- metodo sort
- ordem numerica
- pegar o maior numero de um array
- pegar o menor numero de um array
- filtragem de numeros

### if e else

Ex:

```
var interruptor = "on";
```

```
if (interruptor == "on"){  
    alert('A lampada está ligada');  
} else {  
    alert('A lampada esta desligada');  
}
```

### Switch

É usado para realizar diferentes ações com base em diferentes condições no mesmo bloco de verificação. Caso a condição não seja compatível não será executada e o valor padrão será acionado.

### Laço de repetição For

Laços oferecem um jeito fácil e rápido de executar uma ação repetidas vezes

### Eventos de tempo com javascript

Os eventos de tempo permitem a execução do código em intervalos de tempo

especificados. Esses intervalos de tempo são chamados de eventos de cronometragem.

Os dois metodos-chave para usar com Javascript são:

**setTimeout(function, tempo em milisegundos)**

executa uma função, depois de esperar um numero especificado de milissegundos

**setInterval(function, milliseconds)**

é o mesmo que setTimeout( ), mas repete a execução da função continuamente.

### Classes em javascript

Em 2015 foi introduzido no javascript as classes. As classes são um conceito antigo em programação e varias linguagens utilizam elas. Mas no javascript isso é relativamente novo, por isso mesmo programadores experientes não sabem muito bem utilizar.

Basicamente, as classes são como fabricas para criar objetos.

Pode se dizer que são funções especiais para criação de objetos.

Assim como uma fabrica da vida real precisa das maquinas para construir os objetos, as classes no javascript usam um metodo chamado constructor( ) para fabricar os objetos.

### Manipular datas em javascript

- comando base para pegar a data

```
let data = new Date();
```

```
console.log(data);
```

- pegar o ano atual com 4 digitos

```
let ano = data.getFullYear();
```

```
console.log(ano);
```

- pegar o mês atual (de 0 a 11, sendo o janeiro 0 e dezembro 11)

```
let mes = data.getMonth();
```

```
console.log(mes);
```

- pegar o dia do mes - 1 até 31

```
let diaMes = data.getDate();
```

```
console.log(diaMes);
```

- pegar o dia da semana ( 0 até 6)

```
let diaSemana = data.getDay();
```

```
console.log(diaSemana);
```

```
const diasDaSemana = ["Domingo", "Segunda", "Terça", "Quarta", "Quinta", "Sexta",  
"Sabado"];
```

```
let diaSemanaEscrito = diasDaSemana[data.getDay()];
```

```
console.log(diaSemanaEscrito)
```

- pegar a hora = de 0 até 23

```
let hora = data.getHours();
```

```
console.log(hora);
```

- pegar os minutos = 0 até 59

```
let minutos = data.getMinutes();
```

```
console.log(minutos);
```

- pegar segundos = 0 até 59

```
let segundos = data.getSeconds();
```

```
console.log(segundos);
```

- pegar os milisegundos = 0 até 999

```
let milisegundos = data.getMilliseconds();
```

```
console.log(milisegundos);
```

- pegar a data no padrão brasileiro dia/mes/ano

```
let dataBr = data.toLocaleString('pt-BR');
```

```
console.log(dataBr);
```

- comparar datas= maior ou menor ex: vencimento

```
var hoje = new Date();
```

```
var vencimento = new Date(2022, 10, 15);
```

```
if(hoje > vencimento){
```

```
    console.log("Sua conta está vencida!");
```

```
} else {
```

```
    console.log("Sua conta ainda vai vencer")
```

```
}
```

## JSON

json significa javascript object notation que traduzido para o portugues fica algo como notação de objeto javascript

Explicando de um modo simples json é basicamente uma forma de converter um objeto em texto e o contrario tambem, um texto e um objeto.

Ele é usado principalmente para transmitir dados entre sistemas de forma simples, já que o formato de texto é lido por praticamente toda linguagem de programação.

Para trabalhar com json no javascript usamos dois metodos:

JSON.parse( ) converte texto no padrão JSON em objetos

JSON.stringify( ) converte objetos em texto padrão JSON