

# PROGRAMAÇÃO PARA WEB

Desenvolvimento Web com  
Javascript

MSc. Leonardo Rocha

- Unidade de Ensino: 03
- Competência da Unidade: Conhecer e compreender a linguagem javascript e seus recursos para desenvolvimento web
- Resumo: Compreender e utilizar os principais recursos da Linguagem JavaScript.
- Palavras-chave: JavaScript, Integração, funções, variáveis.
- Título da Teleaula: Desenvolvimento Web com Javascript

Teleaula nº: 03

---

# SP1

- Você acaba de receber a incumbência de corrigir um formulário de uma página web de uma empresa que está iniciando uma campanha que pretende “ouvir” a opinião de seus clientes. Um dos canais de comunicação é o formulário disponível em sua página criada exclusivamente para essa campanha. Ao finalizar o envio dos dados, não se sabe se os dados estão sendo enviados corretamente. Você precisa criar um jeito de mostrar uma mensagem de confirmação de envio em Javascript.
-

## SP2

- O seu cliente deseja que os dados enviados pelos usuários sejam fidedignos. Principalmente a informação de e-mail que é valiosa para o cliente pois ela será utilizada em campanhas futuras de marketing e divulgação de seus produtos. Por este motivo, você precisa validar o campo e-mail para que a informação seja, minimamente, fiel.
-

Conceitos

---

# A linguagem JavaScript

---

# Introdução ao JavaScript

Esta linguagem é uma forte aliada do HTML e CSS3 e contribui grandemente para lidar com características de desenvolvimento *front-end*. Permite melhorar muito a interatividade de um site. Essa linguagem conta com história interessante de evolução. Teve nome de Cmm, ScriptEase, Mocha, LiveScript e finalmente surgiu o JavaScript.

---

# JavaScript é integrado

O código JavaScript é integrado ao HTML5. Para isso, utiliza-se a tag `<script></script>`, uma tag existente no HTML5. Ele é tão poderoso que permite a criação de jogos 2D e 3D animados com ligação a banco dados e muito mais.

Linguagem interpretada de tipagem dinâmica, ou seja, é possível, por exemplo, mudar o tipo de uma variável em tempo de execução.

# Tipos de dados em Javascript

Alguns dos tipos de dados são:

Primitives

Boolean

Null

Undefined

Number

String



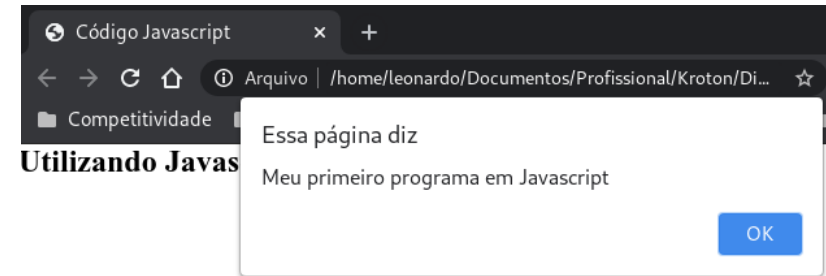
# Exemplo básico (Olá mundo!)

Cada navegador mostrará essa mensagem de a seu modo!

```
<!DOCTYPE html>
<html lang="pt-br">

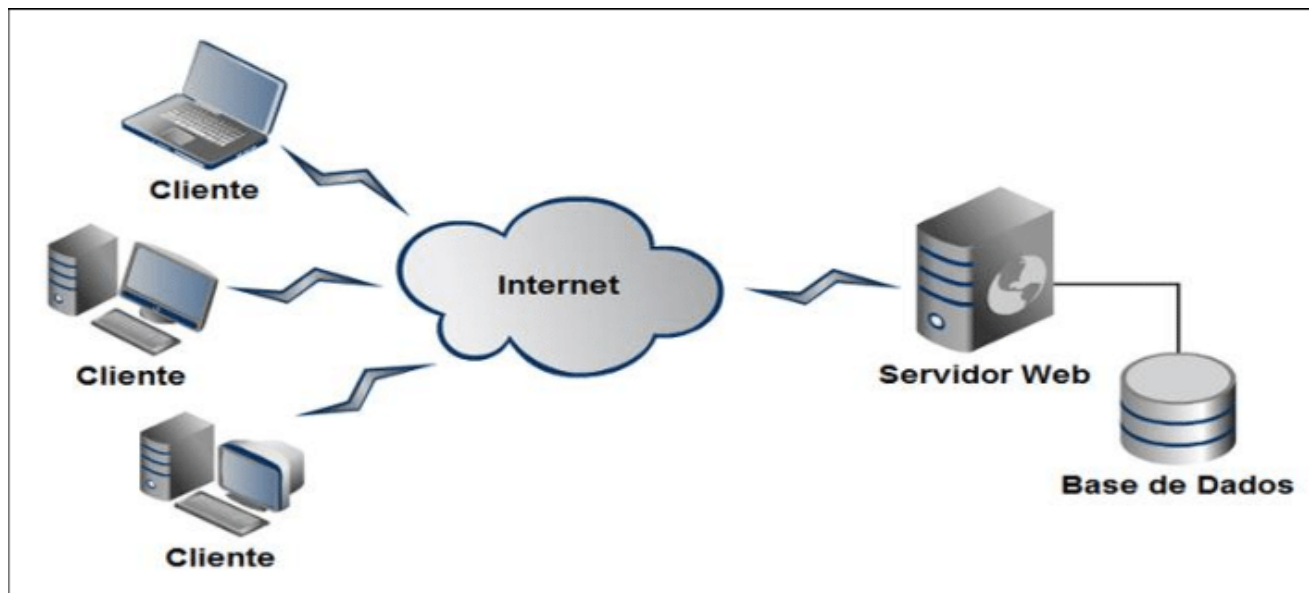
  <head>
    <title>Código Javascript</title>
    <meta charset="UTF-8"/>
    <link rel="stylesheet" href="_css/estilo.css"/>
  </head>
  <body>
    <h2>Utilizando Javascript</h2>
    <script type="text/javascript">
      alert("Meu primeiro programa em Javascript");
    </script>

  </body>
</html>
```



# Arquitetura Cliente-Servidor

JavaScript roda do lado do cliente, ou seja, nos navegadores de internet.



Fonte: <https://bit.ly/3c8gYc0>

# Funções

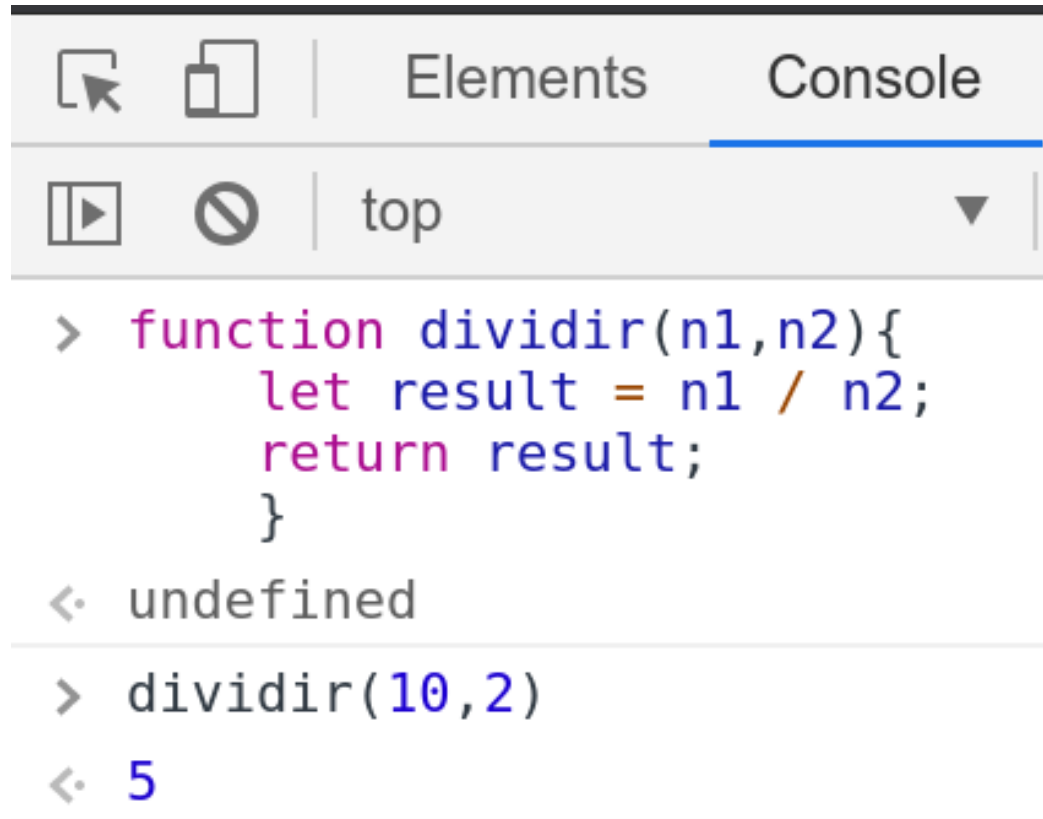
As funções são formas de encapsular funcionalidades que se deseja utilizar. A função guarda um determinado procedimento para que, sempre que você precisar dele novamente, possa chamá-lo apenas pelo nome da função. Isso evita redundância no código e diminui o seu tamanho.

Estrutura:

`Alert();` → Exibe uma caixa pop-up no navegador.

---

# Exemplo de função



The image shows a screenshot of a web browser's developer console. At the top, there are tabs for 'Elements' and 'Console', with 'Console' being the active tab. Below the tabs, there are icons for 'Run' (a play button) and 'Stop' (a circle with a diagonal line), followed by a dropdown menu showing 'top'. The console log contains the following text:

```
> function dividir(n1,n2){  
    let result = n1 / n2;  
    return result;  
}  
  
⏪ undefined  
  
> dividir(10,2)  
  
⏪ 5
```

Fonte: <https://mzl.la/310sG1U>

# Escopo de declaração

Em JavaScript aparece com mais clareza as definições escopo de variável.

Escopo global é onde variáveis são declaradas para uso em todo o código.

Escopo de função é a variável que é declarada dentro de uma determinada função.

Variável de bloco são aquelas criadas em blocos de instruções (IF, FOR, While, etc).

# Operadores matemáticos

Os operadores são essenciais para construção de estruturas de controle. Eles estão presentes em todas as linguagens de programação. Os principais são:

<i>+ soma e concatenação</i>	<i>/</i>
<i>-</i>	<i>% (mod)</i>
<i>*</i>	<i>** (radiciação)</i>

---

# Operadores lógicos

Muito utilizado, sobretudo, nas estruturas de decisão, muito comum em programação. São eles:

AND Lógico (&&) - *Expressão 1 && Expressão 2*

OR Lógico (||) - *Expressão 1 || Expressão 2*

NOT Lógico (!) - *!Expressão*

Outros tipos de operadores: <https://mzl.la/312tAe2>

---

# Operadores relacionais

== Igual

=== Igual ou do mesmo tipo

!= Diferente

!== Diferente e de tipos diferentes

> Maior que

< Menor que

>= Maior ou igual

<= Menor ou igual

---



# Estruturas condicionais

Já que falamos há pouco sobre estrutura de condição, vamos ver uma em Javascript? Veja:

```
let num1 = 8;
```

```
let num2 = 3;
```

```
If (num2 ** num1 >= 10) {
```

```
  Console.log(num2**num1)
```

```
else {
```

```
  console.log("Numero menor que 10") }
```

---

# Estruturas de repetição (laços de iteração)

Nada mais é do que repetir uma determinada instrução até que uma condição seja alcançada.

Entre elas temos o While e o For.

Exemplo simples de for:

```
for (let numero = 0; numero < 10; numero += 2){  
  console.log(numero);  
}
```

---

# Estrutura de repetição (laços de iteração)

Agora um exemplo utilizando o While. Veja:

```
n = 1;
```

```
x = 0;
```

```
while (n < 10) {
```

```
    n++;
```

```
    x += n;
```

```
    console.log(x);
```

```
}
```

---

Conceitos

---

# JavaScript – Parte I

---

# DOM

Document Object Model ou DOM é considerada uma interface de programação para HTML e é muito utilizado em páginas web pois possibilita realizar alterações na estrutura de um documento, resolvendo parcela significativa de problemas do lado do cliente. Por exemplo, você modifica apenas um campo da página se desejar sem necessitar fazer novas requisições ao servidor.

---

# Exemplo de DOM

```
<!DOCTYPE html>
<html lang="pt-br">

  <head>
    <title>Código Javascript</title>
    <meta charset="UTF-8"/>
    <script type="text/javascript">
      window.onload = function() {
        var heading = document.createElement("h2");
        var heading_text = document.createTextNode("Teste de criação");
        heading.appendChild(heading_text);
        document.body.appendChild(heading);
      }
    </script>
  </head>
  <body>

  </body>

</html>
```

Fonte: <https://mzl.la/3175Nd0>

# DOM

Com ele é possível ao usuário fazer customizações na página ou mudar o seu layout sem que seja necessário atualizá-la. Alguns métodos:

`getElementById()` - retorna elemento que estiver com Id informado.

`getElementByClass()` - Retornar um HTML Collection de todos os elementos que contiverem o nome da classe informada.

`getElementsByTagName()` - Retorna os elementos contendo a tag name informada.

Fonte: <https://mzl.la/3175Nd0>

---

# Eventos

Além dos métodos, ainda existem os eventos dos elementos DOM que são os responsáveis por ampliar as opções de interatividade da página. Alguns deles são:

- onclick – usuário clica em um elemento
- onchange – mudança em elemento HTML
- onload – Página é carregada



Resolução da SP

---

**SP1**

---

Algumas informações são extremamente importantes na descrição da Situação Problema. Primeiro que se trata de uma campanha, portanto, conclui-se que a informação coletada com esse formulário é valiosa para a organização. A ideia é bem simples e consiste em mostrar uma mensagem de confirmação de envio dos dados do formulário em JavaScript.

---

```
<h4>Fale Conosco</h4>
<form method="post" action="">
  <div>
    <label>Nome</label>
    <input type="text" id="IDnome" name="nome" placeholder="Nome aqui"/>
  </div><br>
  <div>
    <label>Assunto</label>
    <input type="text" size="40" id="assunto"/>
  </div><br>
  <div>
    <label>Mensagem</label><br><br>
    <textarea placeholder="Decreva o problema" id="mensagem" rows="5" cols=30></textarea>
  </div>
  <div>
```

```
<script>
  function envia(){
    var nome = document.getElementById("IDnome");

    if (nome.value != ""){
      alert(nome.value + "os seus dados foram enviados com sucesso")
    }
  }
</script>
```

---

rofissional/Kroton/Disiplinas/Programação%20para%20Web/website/formulario\_sem\_bootstrap.html

Design Estudos Indicadores

Assunto

Mensagem

Nome

Nome aqui

Assunto

Mensagem

Decreva o problema

Enviar

Mensagem de envio

Leonardo Rocha, os seus dados foram enviados com sucesso

Continuar

rofissional Revistas Software

Interação

# Dúvidas?

---

Conceitos

---

# A linguagem JavaScript – Parte II

---

# Formulários

Com ele é possível ao usuário fazer customizações na página ou mudar o seu layout sem que seja necessário atualizá-la. Alguns métodos:

`GetElementById()` - retorna elemento que estiver com Id informado.

`GetElementsByClass()` - Retornar um HTML Collection de todos os elementos que contiverem o nome da classe informada.

`GetElementsByTagName()` - Retorna os elementos contendo a tag name informada.

---



# Formulário

Um exemplo simples:

```
<!DOCTYPE html>
<html lang="pt-br">

  <head>
    <title>Verificação</title>
    <meta charset="UTF-8"/>
    <script type="text/_javascriptcript"></script>
  </head>
  <body>
    <form name="form" method="post" action="">
      <input type="text" id="texto" name="texto">
      <input type="button" value="Enviar" onclick="ver_preenchimento(this.form)">
    </form>
  </body>
</html>
```

```
function ver_preenchimento(form){
  document.write("Dados enviados com sucesso")
}
```

# Formulário

Assim como no CSS, os comentários em JavaScript são feitos com `/* */`.

Apresenta sintaxe semelhante à linguagens como Java, Perl e Python.

Os tipos de declaração são:

`var` – declara uma variável.

`let` – Declara variável local de escopo do bloco.

`const` – declara uma constante de escopo de bloco.

# JSON

É um formato padrão de texto muito utilizado na representação de dados estruturados com base na sintaxe do objeto JavaScript. No entanto pode ser utilizado sem que seja necessário o uso do JavaScript. Uma de suas utilizações mais comuns é a transmissão de dados por aplicativos web.

Exemplo simples: O envio de dados do lado do cliente para o servidor.

# Estrutura JSON

Como já mencionado, se assemelha muito ao formato literal do objeto JavaScript sendo possível incluir os mesmos tipos de dados básico de objetos JavaScript em seu interior, como: strings, números, booleanos e outros literais de objetos.

```
{  
  "nome": "Guerreiro errante",  
  "age": 1000000,  
  "secretIdentity": "Unknown",  
  "powers": [  
    "Immortality",  
    "Heat Immunity",  
    "Teleportation",  
    "Interdimensional travel"  
  ]  
}
```

Fonte: <https://mzl.la/2QwZBJz>

# Exemplo JSON

Exemplo contido em um único arquivo:

```
<script type="text/javascript">
  function testeAtributos()
  {
    alert("O Primeiro valor é 2");
    this.Valor = 2; //atributo público
    alert("O segundo valor é desconhecido");
    var Valor2 = 4; //atributo privado
    this.Total = this.Valor + Valor2; //atributo publico;
  }

  function TestarJSON()
  {
    var meuObj = new testeAtributos(); // momento de criação do JSON
    alert("Valor "+meuObj.Valor+" + o segundo Valor");
    alert("É igual a "+meuObj.Total);
  }
</script>
</HEAD>
<BODY>
  <button onclick="TestarJSON();">Teste</button>
</BODY>
</HTML>
```

Fonte: Adaptado de <https://bit.ly/3IC2m7L>

# O que é AJAX

Sigla para *Asynchronous JavaScript and XML (AJAX)* é uma técnica que serve para carregar conteúdos é uma página web com uso de JavaScript e XML. Possibilita realizar requisições assíncronas.

Essa técnica de carregamento de conteúdos pode ser realizada com uso de JavaScript e XML, HTML, TXT, PHP, ASP, JSON ou qualquer linguagem de marcação ou programação que pode ser recuperada de um servidor.

# AJAX e JSON

JSON é muito utilizado juntamente com AJAX. Isso porque as chamadas em AJAX, por exemplo, podem obter certas execuções do lado do servidor que obedecem aos padrões adotados em objetos JSON. Dessa forma, por exemplo, os dados retornados podem ser lidos e, eventualmente, tratados em JavaScript com o uso da função `Eval()`.

---

# Canvas

Este é um novo elemento do HTML5 que possibilita manipular com maior grau de controle a criação de desenhos em uma página web. Para isso, utiliza de linguagens de script (JavaScript) para criar coisas como: animação, gráficos de jogos, apresentação de dados e muito mais.

---



# Canvas (Exemplo)

O método `Document.getElementById()` referencia o elemento HTML `<canvas>`. Em seguida, o método `HTMLCanvasElement.getContext()` pega o contexto daquele elemento - a coisa sobre a qual o desenho será renderizado.

```
const canvas = document.getElementById('canvas');  
const ctx = canvas.getContext('2d');  
  
ctx.fillStyle = 'blue';  
ctx.fillRect(100, 100, 150, 150);
```

Desenho criado com auxílio do Canvas



```
<h2>Desenho criado com auxílio do Canvas</h2>  
<canvas id="canvas"></canvas>
```

Fonte: Adaptado de <https://mzl.la/3cZNpbJ>

# Geolocalização

*Geolocation* permite fornecer a localização do usuário para aplicações web que ele consentir. Este, é um objeto filho que está dentro do objeto pai *navigator*. A partir dessa localização, scripts feitos em JavaScript podem fazer seu trabalho e explorar essa informação. Um exemplo comum é publicidade de alimentos nas proximidades de sua localização.

O método `getCurrentPosition()` permite obter a localização atual do usuário.

---

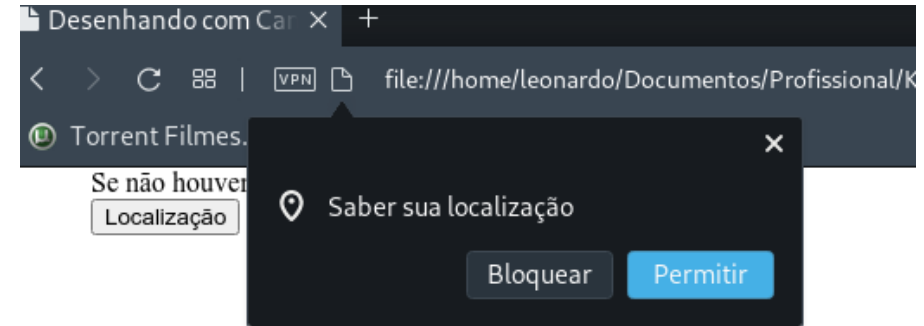
# Geolocalização

```
<body>
<p id="demo">Se não houver permissão, a posição não será mostrada.</p>
<p>
<button class="w3-btn w3-blue" onclick="getLocation()">Localização</button>
</p>
<div id="mapholder"></div>
<script src="https://maps.google.com/maps/api/js?key=AIzaSyBu-916DdpKAjTmJNigngS6HL_kDIKU0aU"></script>
<script>
var x=document.getElementById("demo");
function getLocation()
{
  if (navigator.geolocation)
  {
    navigator.geolocation.getCurrentPosition(showPosition,showError);
  }
  else{x.innerHTML="Geolocalização não é suportada por esse navegador.";}
}

function showPosition(position)
{
  var lat=position.coords.latitude;
  var lon=position.coords.longitude;
  var latlon=new google.maps.LatLng(lat, lon)
  var mapholder=document.getElementById('mapholder')
  mapholder.style.height='250px';
  mapholder.style.width='100%';

  var myOptions={
    center:latlon,zoom:14,
    mapTypeId:google.maps.MapTypeId.ROADMAP,
    mapTypeControl:false,
    navigationControlOptions:{style:google.maps.NavigationControlStyle.SMALL}
  };
  var map=new google.maps.Map(document.getElementById("mapholder"),myOptions);
  var marker=new google.maps.Marker({position:latlon,map:map,title:"Você está aqui!"});
}
</script>
</body>
```

Fonte: Adaptado de <https://bit.ly/3cTijT3>



# Áudio e Vídeo

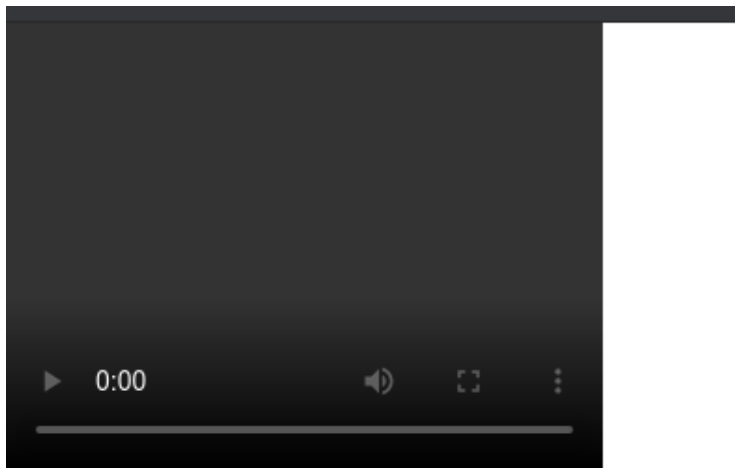
Controlar áudio e vídeo na web com auxílio do JavaScript é algo relativamente simples. No HTML5 foram criados dois novos elementos para auxiliar nessa tarefa. `<video>` e `<audio>` são os elementos que permitem um controle maior sobre estes recursos na web. Além disso, surgiram APIs JavaScript que foram criados para ampliar controle sobre essas mídias.

---

# Áudio e Vídeo (Exemplo)

```
<!DOCTYPE html>
<html lang="pt-br">

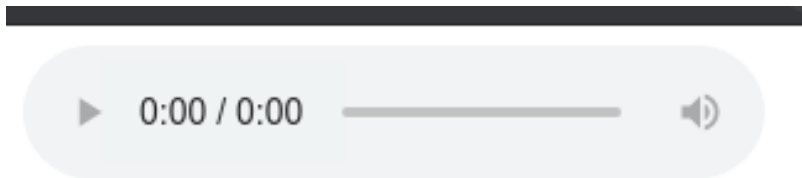
  <head>
    <title>Audio e vídeo</title>
    <meta charset="UTF-8"/>
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/bootstrap.min.css">
  </head>
  <body>
    <video width="320" height="240" controls>
      <source src="https://www.youtube.com/watch?v=XUJGajpWW2U" type="">
    </video>
    <p>Se seu navegador não suportar HTML5, clique <a href="https://www.youtube.com/watch?v=XUJGajpWW2U">aqui</a></p>
  </body>
</html>
```



Se seu navegador não suportar HTML5, clique [aqui](https://www.youtube.com/watch?v=XUJGajpWW2U)

# Áudio e Vídeo (Exemplo)

```
<head>
  <title>Audio e vídeo</title>
  <meta charset="UTF-8"/>
</head>
<body>
  <audio autoplay="autoplay" controls="controls">
    <source src="musica.mp3" type="audio/mp3">
  </audio>
</body>
</html>
```



# Animação e transição em CSS

```
<div>
```

O processo de animação é relativamente simples. Pode ser utilizado no tratamento de imagens, por exemplo.

```
</div>|
```

```
div{
  animation-duration: 1s;
  animation-name: direita;
}
@keyframes direita{
  from{
    margin-right: 100%;
    width: 0%;
  }
  to {
    margin-left: 30%;
    width: 80%;
  }
}
```

Executar

# Bibliotecas

As bibliotecas, assim como os frameworks, são conjuntos de funcionalidades criados para colaborar com o o processo de desenvolvimento. Um bom exemplo disso são funcionalidades semelhantes existentes em sites na internet. Por exemplo, a exposição de álbuns de fotos. Geralmente programadores reutilizam código de programação. E isso é possível por conta das bibliotecas. Em termos simples, uma biblioteca contém um conjunto de funções prontas que podem ser utilizadas em seu site.

---



# Bibliotecas conhecidas

Temos algumas bibliotecas conhecidas a saber:

Node.js – Utilizado para desenvolvimento em back-end

Jquery – Simplifica desenvolvimento em JavaScript

React – Criação de interfaces (Em alto no mercado)

D3.js – Visualização de dados web

AngularJS – Construção de documentos Web Single page

E outros

---

# JQuery

De inicialização fácil e boa documentação disponível na web. Tem vários efeitos de animação disponível, como, por exemplo, slide.

```
<body>
  <script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
  <audio autoplay="autoplay" controls="controls">
    <source src="musica.mp3" type="audio/mp3">
  </audio>
```

# React

Biblioteca que está em alta no momento sendo utilizada por várias empresas de tecnologia por sua facilidade de uso e versatilidade. Voltada para construção de interfaces com usuário, não é framework e tão pouco exclusivo para internet. Por exemplo, o React Native é utilizado para construção de aplicativos móveis.

---

# React

Biblioteca que está em alta no momento sendo utilizada por várias empresas de tecnologia por sua facilidade de uso e versatilidade. Voltada para construção de interfaces com usuário, não é framework e tão pouco exclusivo para internet. Por exemplo, o React Native é utilizado para construção de aplicativos móveis.

---

# Bootstrap Vue (Javascript)

Para utilizar o Bootstrap Vue é instalar o Node.js pois será necessário utilizar o npm (gerenciador de pacotes do Node) para instalar o bootstrap e o bootstrap-vue. A seguir um exemplo de arquivo main.js com as importações necessárias para o seu bom funcionamento.

```
import App from './App.vue'
import Vue from 'vue'
import { BootstrapVue, IconsPlugin } from 'bootstrap-vue'
import 'bootstrap/dist/css/bootstrap.css'
import 'bootstrap-vue/dist/bootstrap-vue.css'
Vue.use(BootstrapVue)
Vue.use(IconsPlugin)
Vue.config.productionTip = false
new Vue({
  render: h => h(App)
}).$mount('#app')
```

Resolução da SP

---

**SP2**

---

- Você deve criar um script para validação de e-mail. Isso se torna fácil uma vez que você pode encontrar vários disponíveis na internet. Basta adaptá-lo.

```
<script language="Javascript">
function validarEmail(field) {
  usuario = field.value.substring(0, field.value.indexOf("@"));
  dominio = field.value.substring(field.value.indexOf("@")+ 1, field.value.length);

  if ((usuario.length >=1) &&
      (dominio.length >=3) &&
      (usuario.search("@")==-1) &&
      (dominio.search("@")==-1) &&
      (usuario.search(" ")==-1) &&
      (dominio.search(" ")==-1) &&
      (dominio.search(".")!=-1) &&
      (dominio.indexOf(".") >=1)&&
      (dominio.lastIndexOf(".") < dominio.length - 1)) {
    document.getElementById("msgemail").innerHTML="E-mail válido";
    alert("E-mail valido");
  }
  else{
    document.getElementById("msgemail").innerHTML="<font color='red'>E-mail inválido </font>";
    alert("E-mail invalido");
  }
}
</script>
```

Fonte: <https://bit.ly/2Pm5ngi>

**A casadinha  
HTML5, CSS e  
Javascript é  
realmente útil?**

---



Conceitos

---

# Recapitulando

---

# Recapitulando

- JavaScript, sintaxe, variáveis e estrutura
  - Funções e métodos
  - Utilização com CSS e HTML5
-

