

HTML, CSS E Javascript

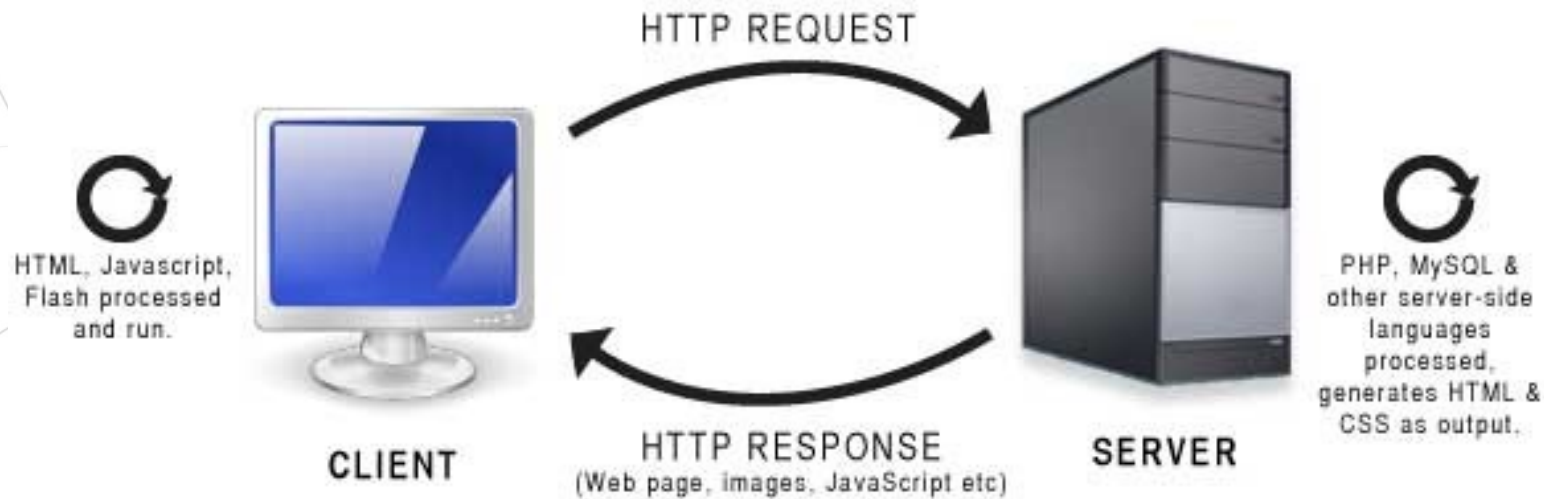
Prof. Dr. João Ricardo Favan

joao.favan@univem.edu.br

Agenda

- Client side x Server side
- HTML
- CSS
- JavaScript

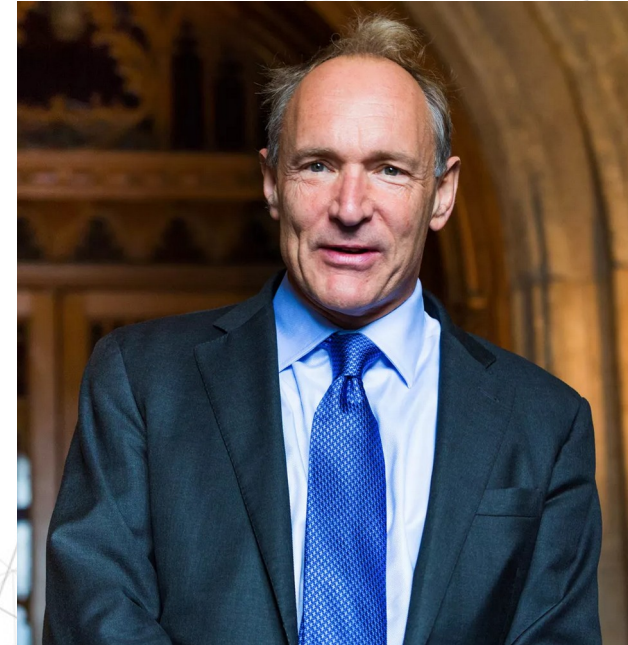
Cliente side vs Server side



HTML

HTML - História

- HyperText Markup Language - 1991
- Tim Bernes-Lee
 - Físico britânico
 - Criador do WWW
 - Criador do HTTP
 - Diretor do W3C
 - Resolver problema de comunicação



HTML – Primeiro Website

World Wide Web

The WorldWideWeb (W3) is a wide-area [hypermedia](#) information retrieval initiative aiming to give universal access to a large universe of documents.

Everything there is online about W3 is linked directly or indirectly to this document, including an [executive summary](#) of the project, [Mailing lists](#) , [Policy](#) , November's [W3 news](#) , [Frequently Asked Questions](#) .

[What's out there?](#)

Pointers to the world's online information, [subjects](#) , [W3 servers](#), etc.

[Help](#)

on the browser you are using

[Software Products](#)

A list of W3 project components and their current state. (e.g. [Line Mode](#) ,X11 [Viola](#) , [NeXTStep](#) , [Servers](#) , [Tools](#) , [Mail robot](#) , [Library](#))

[Technical](#)

Details of protocols, formats, program internals etc

[Bibliography](#)

Paper documentation on W3 and references.

[People](#)

A list of some people involved in the project.

[History](#)

HTML - Versões



HTML - Para que serve?

- Códigos que delimitam conteúdos específicos (tags)
- Desenvolvimento de websites
- Primeira linguagem para criação de sites

HTML – limitações

- Conteúdo estático
- Sem estilos

[Home](#)

[Apostila de HTML](#)

[Apostila de CSS](#)

[Apostila de HTML 5](#)

[Sobre o autor](#)

[Contato](#)

Apostilas para Webmasters

Bem vindos ao nosso site!

Aqui vocês encontrarão apostilas, dicas e tutoriais sobre:

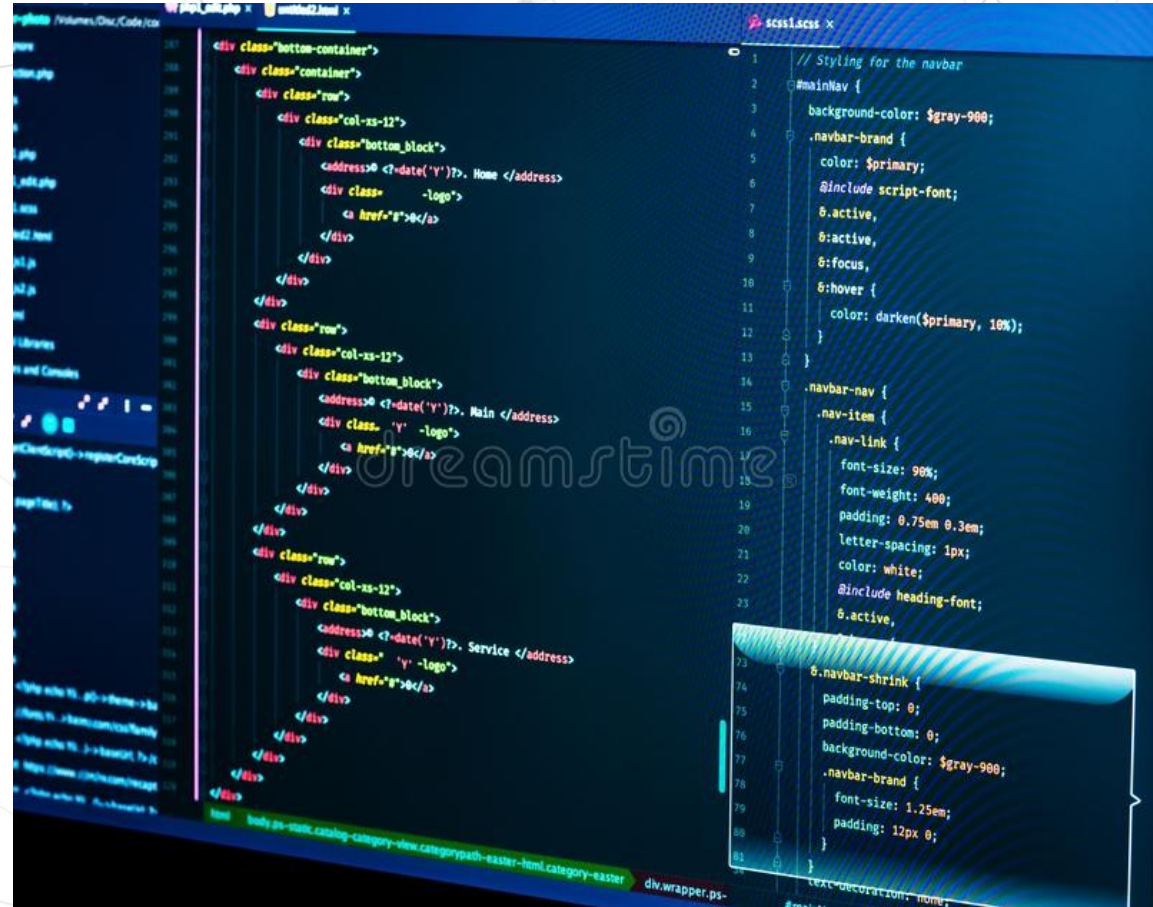
- HTML 4
- HTML 5
- CSS
- Webdesign

Bons estudos!

HTML - O que são tags?

- Estruturas de linguagem de marcação
- Informa ao navegador qual o tipo de conteúdo deve ser apresentado
- Possui início e fim
- Delimitadores de estilo e conteúdo
- Exemplo: `<title>UNIVEM</title>`

HTML - tags



CSS

CSS - História

- Cascading Style Sheet
- Håkon Wium Lie
- Criado em 1994
- Estilos eram aplicados no HTML
- Criados com tabelas

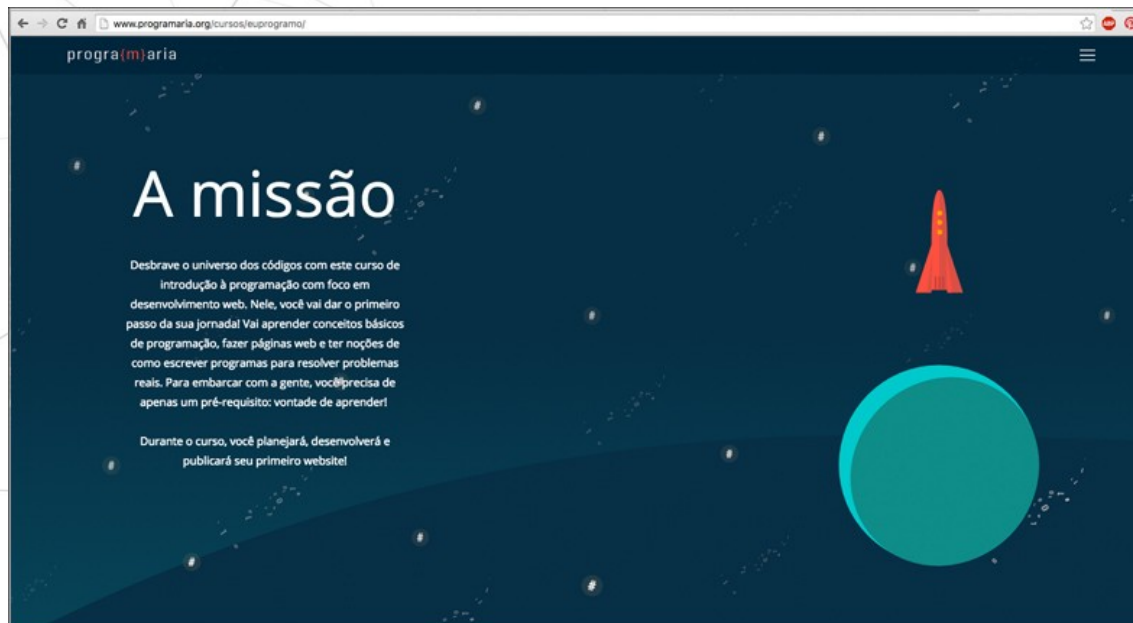


CSS - História

- Demorou muito tempo chegar nos navegadores
- Quirks Mode (*emula comportamento não padrão*)
- Almost standards mode (*Comportamento descrito em html e css*)
- Full standards Mode (*Não implementa peculiaridades não-padrão*)

CSS

- Estilo e formatação a página WEB



CSS – Aplicando estilos

- Aplicando estilos
 - Por ID

```
/* Utilizando ID */  
#idDoInput {  
    margin-top: 50px;  
}
```

CSS – Aplicando estilos

- Aplicando estilos
 - Por classe

```
/* Utilizando Classe */  
.classeDoElemento {  
  margin-top: 50px;  
  background-color: red;  
}
```

CSS – Aplicando estilos

- Aplicando estilos
 - Por tags



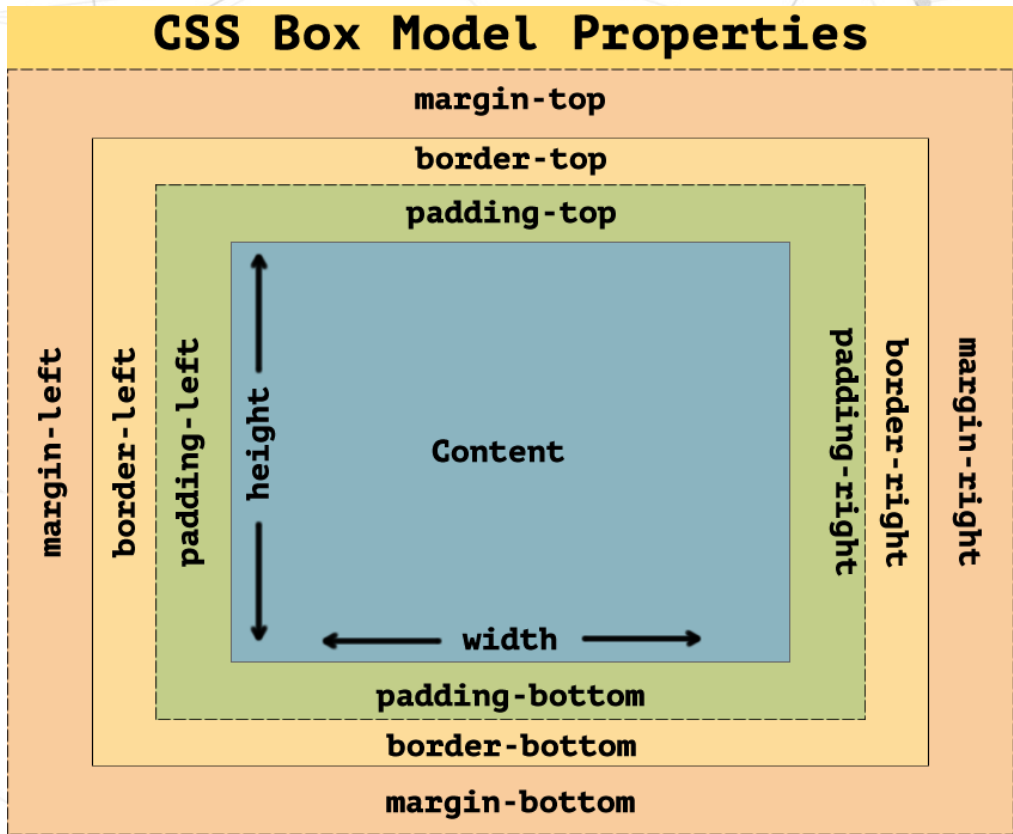
```
/* Utilizando tags */  
div {  
    padding: 0px 10px 0px 10px  
}
```


CSS – Aplicando estilos

- Aplicando estilos
 - Por atributos

```
/* Utilizando atributos */  
div[data="teste"] {  
  color: #0000ff;  
}
```

CSS - Box Model



CSS - Box Model



```
/* Box model */  
div {  
  width: 300px;  
  border: 15px solid green;  
  padding: 50px;  
  margin: 20px;  
}
```

CSS – Grid layout

HTML

```
<div class="wrapper">
  <div class="box a">a</div>
  <div class="box b">b</div>
  <div class="box c">c</div>
  <div class="box d">d</div>
  <div class="box e">e</div>
  <div class="box f">f</div>
</div>
```

CSS

```
.wrapper {
  display: grid;
  grid-template-columns: 100px 100px 100px;
  grid-gap: 10px;
  background-color: #fff;
  color: #444;
}

.box {
  background-color: #444;
  color: #fff;
  border-radius: 5px;
  padding: 20px;
  font-size: 150%;
}
```

CSS – Flexbox

- Guia:

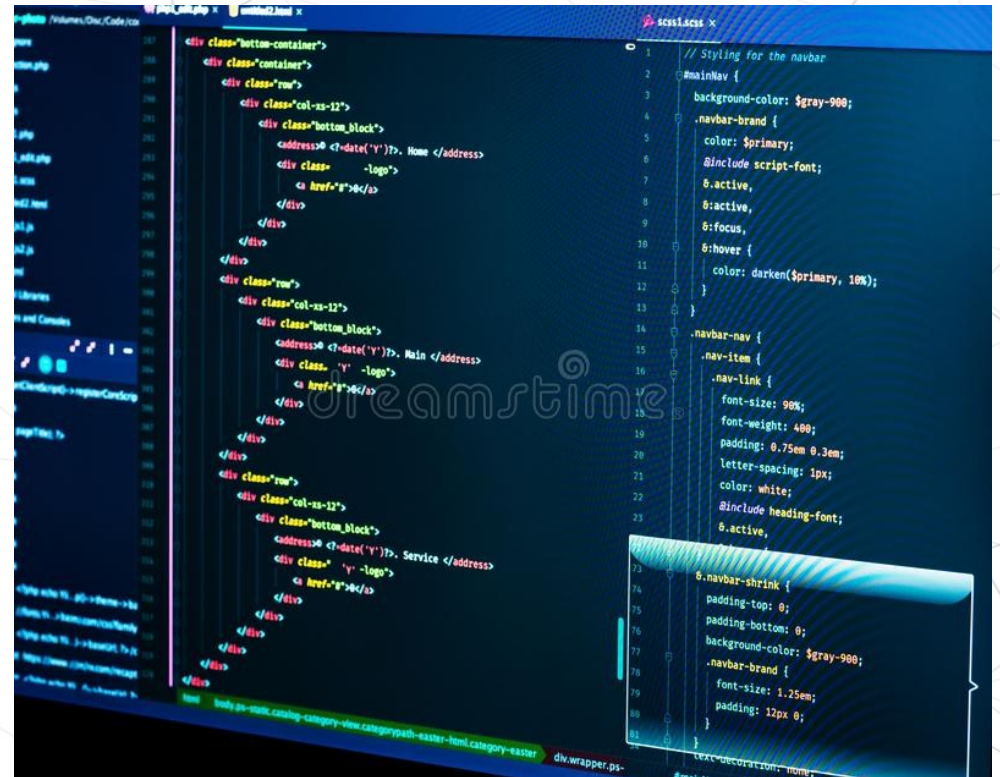
<https://origamid.com/projetos/flexbox-guia-completo/>

```
/* flexbox */  
.flex-container {  
  display: flex;  
  flex-wrap: wrap;  
}  
  
.flex-container > div {  
  border: 2px solid #0e0a5d;  
  background-color: #f1f1f1;  
  text-align: center;  
  line-height: 75px;  
  font-size: 30px;  
  flex: 1 25%;  
}
```


CSS - Flexbox - Exemplo

1	2	3
4	5	6
7		8

CSS - codificando



```
// Styling for the navbar
1 // Styling for the navbar
2 #mainnav {
3   background-color: $gray-900;
4   .navbar-brand {
5     color: $primary;
6     @include script-font;
7     @active,
8     @active,
9     @focus,
10    @hover {
11      color: darken($primary, 10%);
12    }
13  }
14  .navbar-nav {
15    .nav-item {
16      .nav-link {
17        font-size: 90%;
18        font-weight: 400;
19        padding: 0.75em 0.3em;
20        letter-spacing: 1px;
21        color: white;
22        @include heading-font;
23        @active,
24        @active,
25        @focus,
26        @hover {
27          color: $primary;
28          font-weight: bold;
29        }
30      }
31    }
32  }
33  .navbar-shrink {
34    padding-top: 0;
35    padding-bottom: 0;
36    background-color: $gray-900;
37    .navbar-brand {
38      font-size: 1.25em;
39      padding: 12px 0;
40    }
41  }
42 }
```

```
<div class="bottom-container">
1 <div class="container">
2   <div class="row">
3     <div class="col-12">
4       <div class="bottom_block">
5         <address><?date('Y')?>. Home </address>
6         <div class="
7           -logo">
8         <a href="#"></a>
9       </div>
10      </div>
11    </div>
12    <div class="row">
13      <div class="col-12">
14        <div class="bottom_block">
15          <address><?date('Y')?>. Main </address>
16          <div class="
17            'Y' -logo">
18          <a href="#"></a>
19        </div>
20      </div>
21    </div>
22    <div class="row">
23      <div class="col-12">
24        <div class="bottom_block">
25          <address><?date('Y')?>. Service </address>
26          <div class="
27            'Y' -logo">
28          <a href="#"></a>
29        </div>
30      </div>
31    </div>
32  </div>
33 </div>
```

CSS - Frameworks



Atividade

Javascript

Javascript - História

- Primeiro nome: Mocka - 1995
- Brendan Eich
- Java fazia muito sucesso
 - Applet java
- Livescript
- Javascript
 - Applet



Javascript - História

- JotaScript - IE
- ECMAScript - 1997
- Navegadores com motores internos
 - V8 da google 2009
- Nodejs 2010

Javascript – Para que serve?

- TUDO



Javascript e ECMAScript

JS

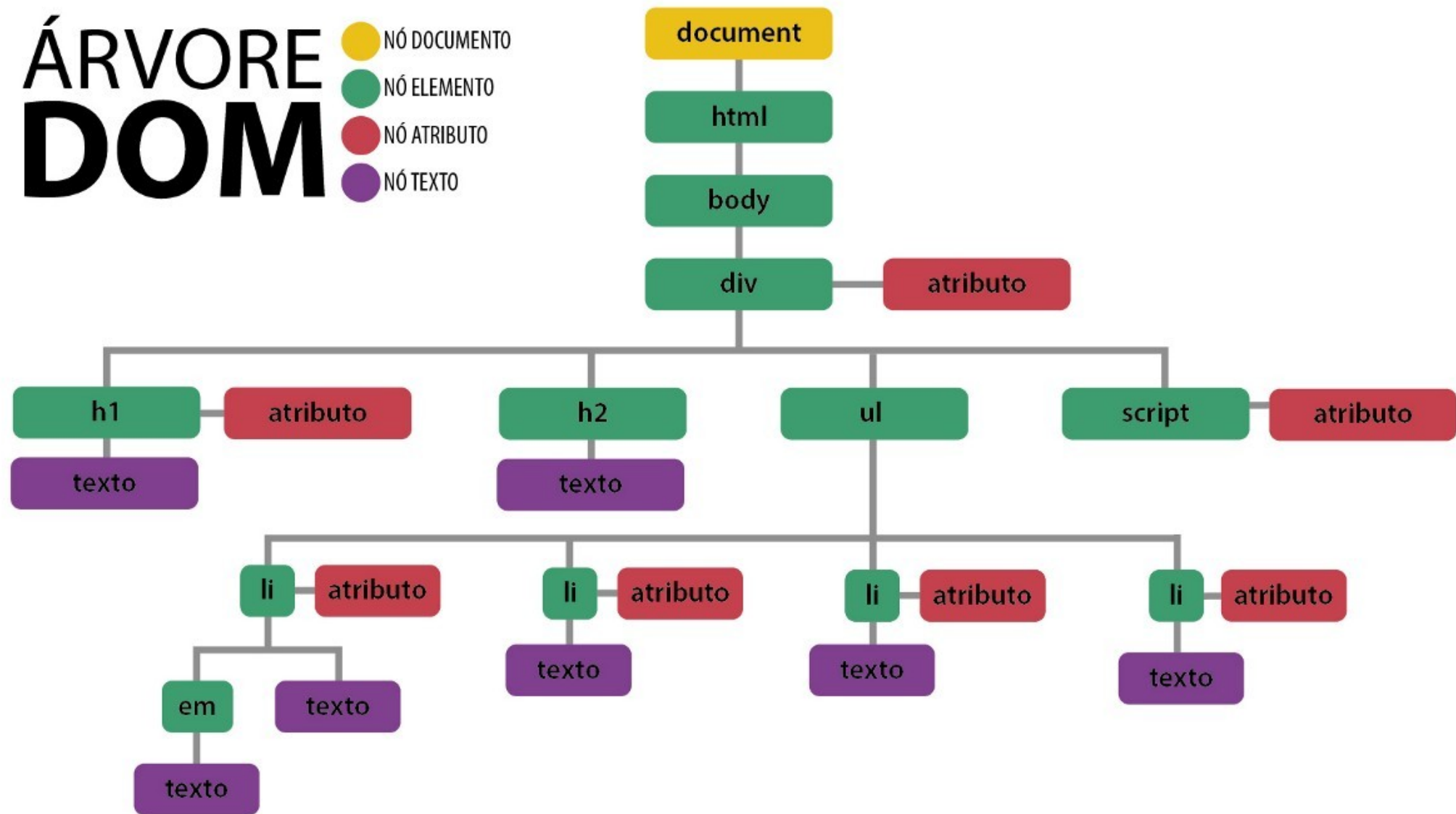
ES

Javascript – O que é DOM?

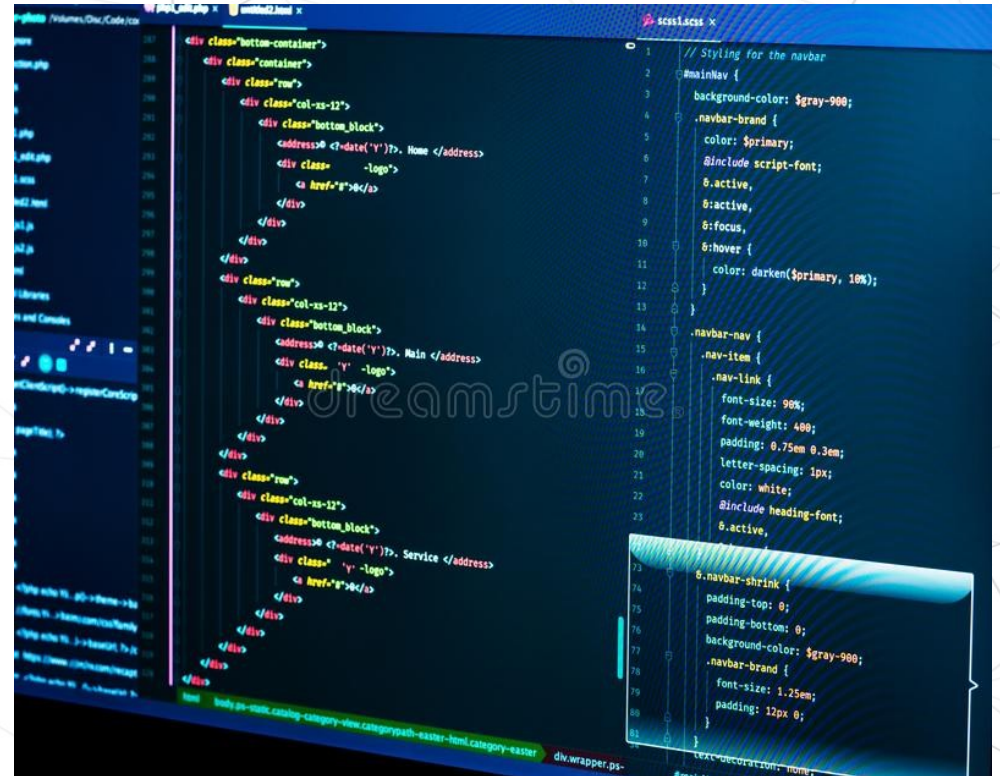
- Modelo de Objeto de Documento
- Interface de programação para documentos HTML, XML e SVG para construir páginas e conteúdos através dos navegadores
- O DOM representa o documento com nós e objetos, dessa forma, as linguagens de programação podem se conectar à página. (Doc Mozilla)

ÁRVORE DOM

- NÓ DOCUMENTO
- NÓ ELEMENTO
- NÓ ATRIBUTO
- NÓ TEXTO

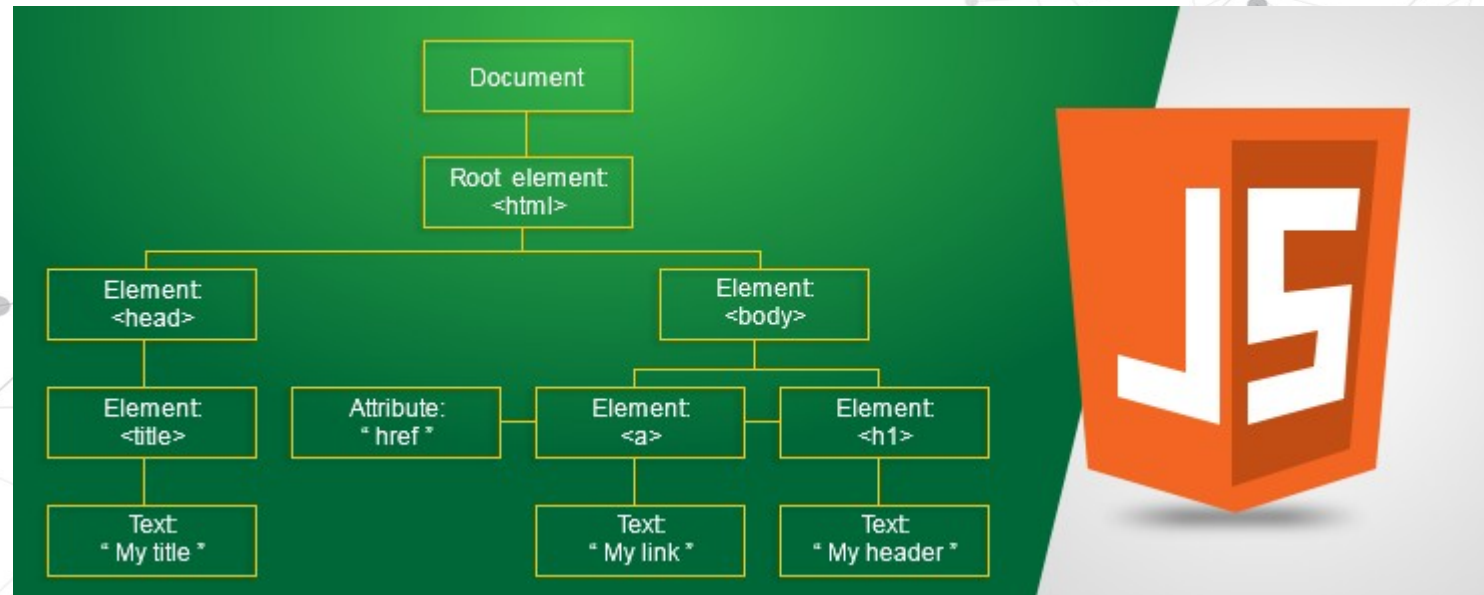


Javascript - Codificando DOM



Javascript – O que são elementos?

- Elemento é um item da árvore DOM que possui seus atributos



Javascript – Manipulando DOM

Métodos de acesso

- ID

```
/* Por Id */  
var elementById = document.getElementById("#idDoElemento");
```

Javascript – Manipulando DOM

- Métodos de acesso
 - Name

```
/* Por Name */  
var elementByName = document.getElementsByName("nomeDoElemento")[0];
```


Javascript – Manipulando DOM

- Métodos de acesso
 - Class



```
/* Por Class */  
var elementByClassName = document.getElementsByClassName("classeDoElemento")[0];
```

Javascript – Manipulando DOM

- Métodos de acesso
 - TagName

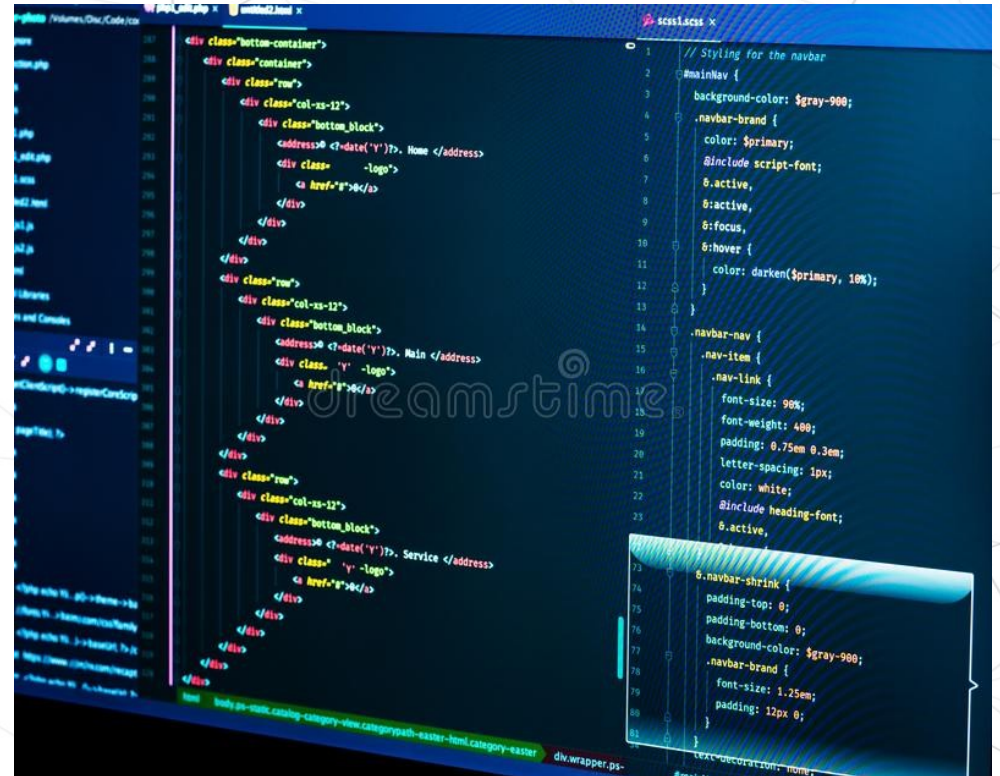
```
/* Por TagName */  
var elementByTagName = document.getElementsByTagName("p")[0];
```

Javascript – Manipulando DOM

- Métodos de acesso
 - Query Selector (css)

```
/* Query Selector */  
var elementByQuerySelector = document.querySelector(".classeDoElemento");  
var elementByQuerySelector = document.querySelector("#idDoElemento");  
var elementByQuerySelector = document.querySelector("input[name='nomeDoInput']");
```

Javascript - Métodos de acesso



```
<div class="bottom-container">
  <div class="container">
    <div class="row">
      <div class="col-12">
        <div class="bottom_block">
          <address><!--date('Y')-->. Home </address>
          <div class="
            -logo">
            <a href="#"></a>
          </div>
        </div>
      </div>
    </div>
    <div class="row">
      <div class="col-12">
        <div class="bottom_block">
          <address><!--date('Y')-->. Main </address>
          <div class="
            'Y' -logo">
            <a href="#"></a>
          </div>
        </div>
      </div>
    </div>
    <div class="row">
      <div class="col-12">
        <div class="bottom_block">
          <address><!--date('Y')-->. Service </address>
          <div class="
            'Y' -logo">
            <a href="#"></a>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
```

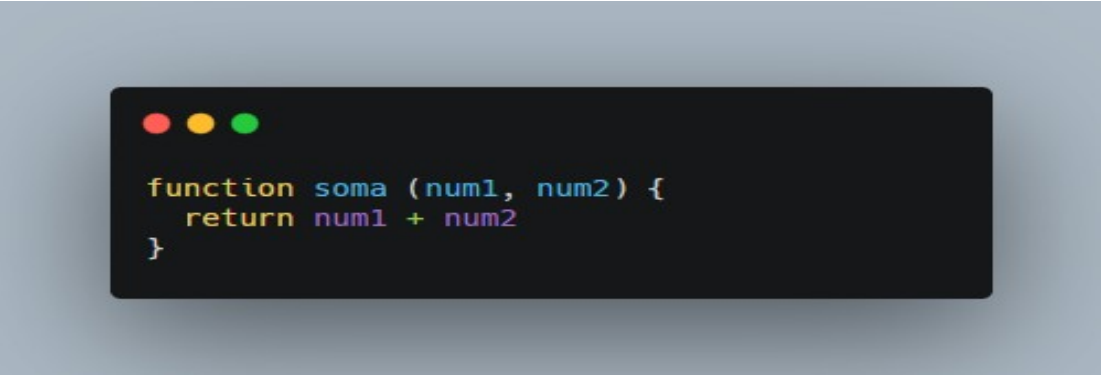
```
// Styling for the navbar
#mainnav {
  background-color: $gray-900;
  .navbar-brand {
    color: $primary;
    @include script-font;
    @active,
    @active,
    @focus,
    @hover {
      color: darken($primary, 10%);
    }
  }
  .navbar-nav {
    .nav-item {
      .nav-link {
        font-size: 90%;
        font-weight: 400;
        padding: 0.75em 0.3em;
        letter-spacing: 1px;
        color: white;
        @include heading-font;
        @active,
        @active,
        @active {
          padding-top: 0;
          padding-bottom: 0;
          background-color: $gray-900;
          .navbar-brand {
            font-size: 1.25em;
            padding: 12px 0;
          }
        }
      }
    }
  }
}
```

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.

Javascript – Funções

- Maneiras de declaração



```
function soma (num1, num2) {  
  return num1 + num2  
}
```

Javascript – Funções

- Maneiras de declaração



```
const soma = (num1, num2) => num1 + num2;
```

Javascript – Funções

- Maneiras de declaração

```
let obj = {};  
obj.func = function() {  
    return "oi";  
};
```

Javascript – Eventos

- Ações que acontecem no sistema
- Capturado e executado alguma ação

- Lista de eventos:

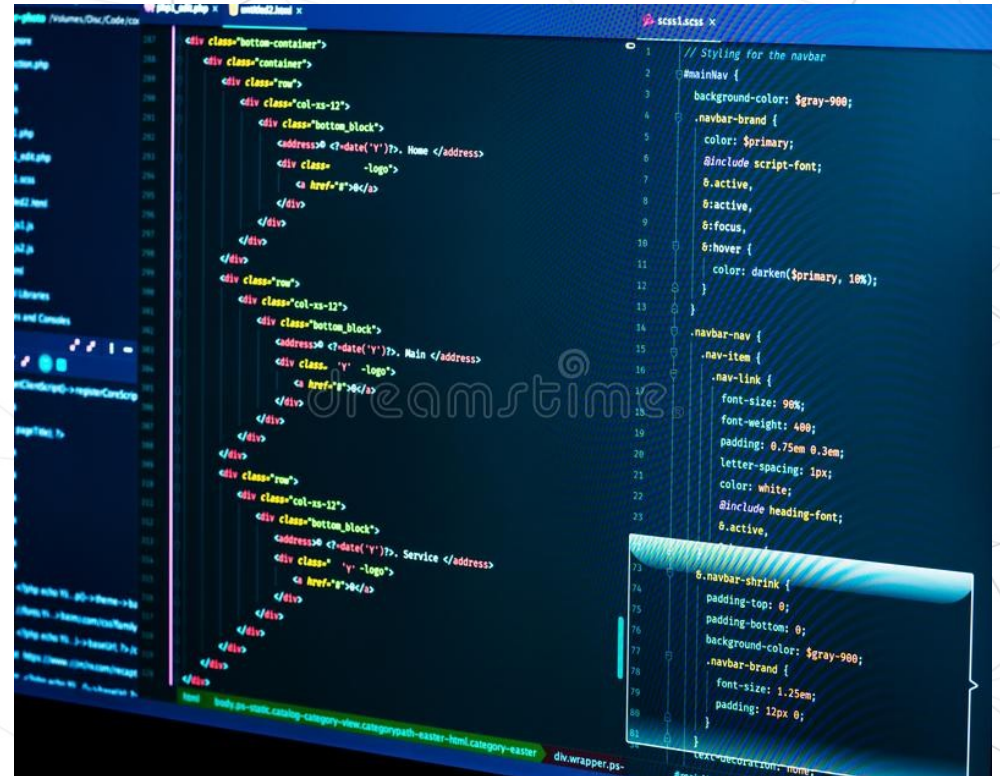
<https://developer.mozilla.org/pt-BR/docs/Web/Events>

Javascript – Evento com HTML



```
<button type="button" onclick="soma()">Clique aqui</button>
```


Javascript - Evento HTML



Javascript – listeners?

- Ouvintes
- EventListeners

var	let	const
It is available right from the beginning when the JavaScript was introduced.	It is a new way to declare variables in JavaScript, starting from ES6.	const is used to store a value that will not be changed throughout the execution of the script. It is also introduced recently in ES6.
It has a global/function scope.	It has block scope.	It also has block scope.
Can be updated or re-declared in its scope.	We can't re-declare them.	const represents a constant value, so it can't be updated or re-declared.

VAR vs LET vs CONST

	var	let	const
Stored in Global Scope	✓	✗	✗
Function Scope	✓	✓	✓
Block Scope	✗	✓	✓
Can Be Reassigned?	✓	✓	✗
Can Be Redeclared?	✓	✗	✗
Can Be Hoisted?	✓	✗	✗

Javascript – AJAX (json e xml assíncrono)

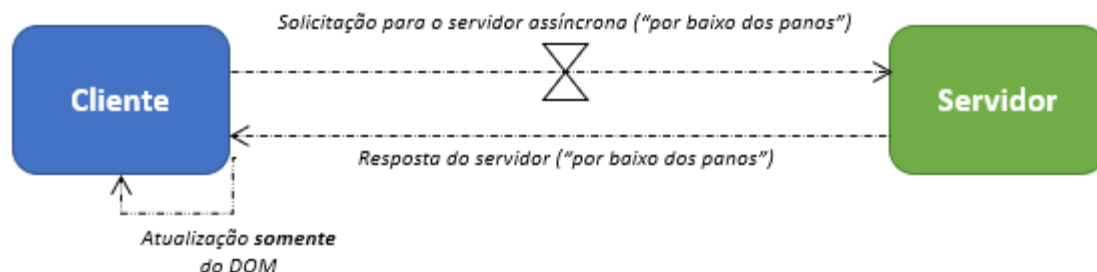
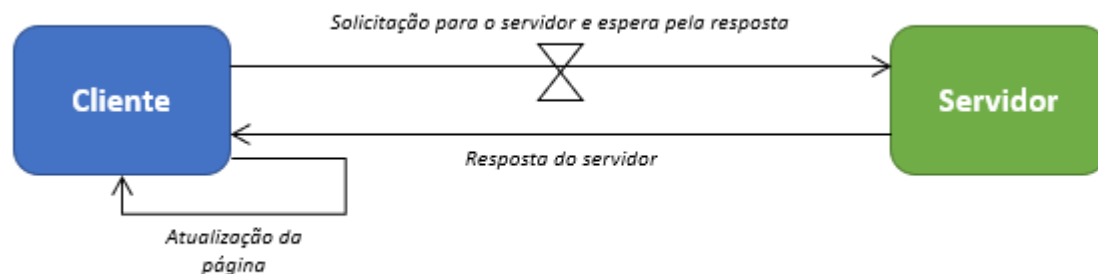
Permitindo o compartilhamento assíncrono de informações entre cliente e servidor.

As informações trocadas entre cliente e servidor geralmente estão no formato de um JSON.

Javascript – AJAX (json e xml assíncrono)

O navegador gera uma chamada do JavaScript que irá ativar o XMLHttpRequest. Com isso, em segundo plano, o navegador cria uma requisição HTTP para o servidor. Este recebe a requisição, busca os dados e envia para o navegador. O navegador, ao receber a resposta de volta, utiliza o DOM para modificar a página atual de maneira que esta reflita a resposta que veio do servidor.

Javascript - AJAX (json e xml assíncrono)



Javascript - AJAX

- Como utilizar com Javascript?

```
/* Método, URL, async */
var xhr = new XMLHttpRequest();
xhr.open("GET", "https://viacep.com.br/ws/01001000/json/", true);
xhr.onload = function (e) {
    if (xhr.readyState === 4) {
        if (xhr.status === 200) {
            var dados = JSON.parse(xhr.responseText);
            console.log("sucesso");
            console.log(dados);
        } else {
            console.error(xhr.statusText);
        }
    }
};
xhr.onerror = function (e) {
    console.error(xhr.statusText);
};
xhr.send();
```

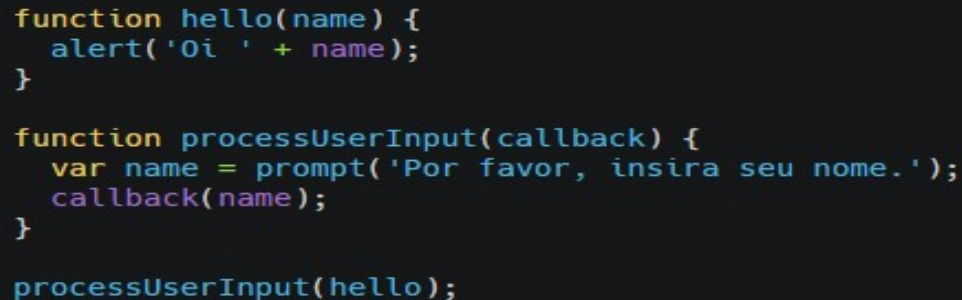
Javascript - Callbacks

- Uma função callback é uma função passada a outra função como argumento, que é então invocado dentro da função externa para completar algum tipo de rotina ou ação.
- Utilizado para executar após as funções assíncronas serem executadas

Javascript - Callbacks

- Exemplo:

```
function hello(name) {  
    alert('Oi ' + name);  
}  
  
function processUserInput(callback) {  
    var name = prompt('Por favor, insira seu nome.');
```



```
    callback(name);  
}  
  
processUserInput(hello);
```


Projeto