

{Frontend}

...

# Apresentação

- Odair Filipe Teixeira
- Graduado em Ciência da Computação
- Técnico em Automação Industrial
- Trabalhando na Trier desde 2019



**Odair Teixeira**  
Desenvolvedor | DSV

📞 (48) 3658-9800

✉️ odair.teixeira@triersistemas.com.br

💻 [www.triersistemas.com.br](http://www.triersistemas.com.br)



# Conteúdo

HTML

CSS

Javascript

Angular

# Ferramentas



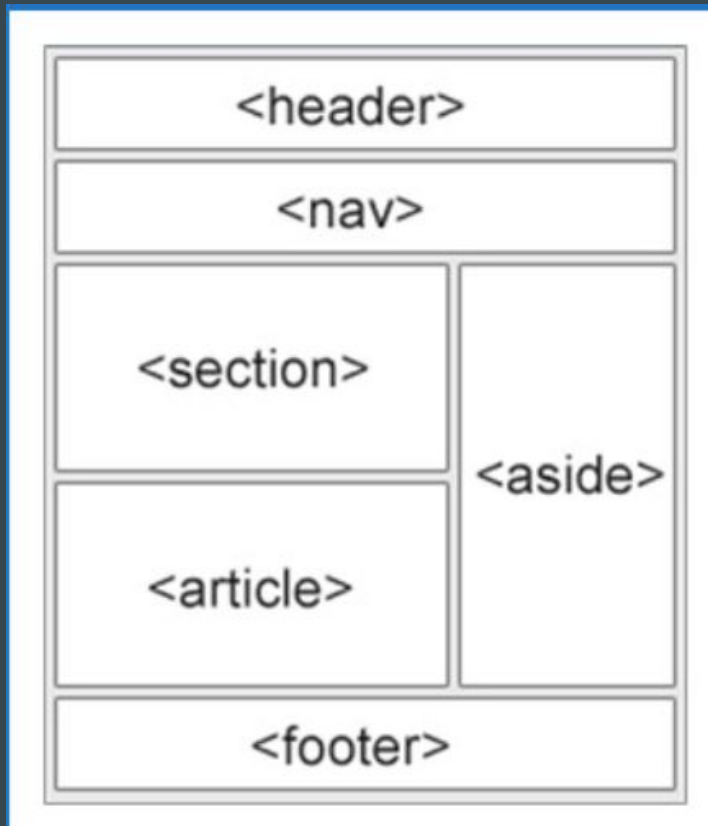
VSCode

<https://code.visualstudio.com/download>



Live Server

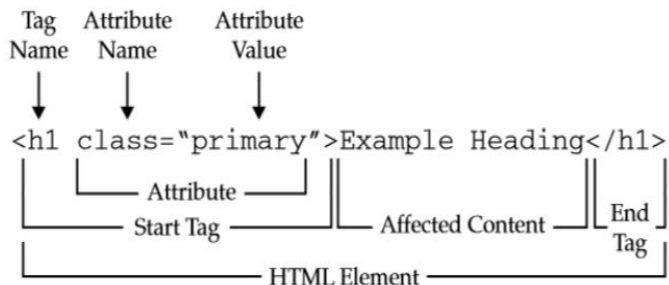
# O que é HTML?



- HTML (HyperText Markup Language) é a linguagem padrão para criar páginas web.
- Define a estrutura e o conteúdo de uma página.
- Navegadores interpretam o HTML e exibem as páginas.

# TAGS


A graphical overview of the HTML markup syntax shown so far is presented here:



- Todo documento HTML apresenta elementos entre `<` e `>`, esses elementos são chamados de tags, que são os comandos de formatação da linguagem.

# Estrutura Básica de uma Página HTML

<> index.html ✕

<> index.html >  html

```
1  <!DOCTYPE html>
2  <html lang="en">
3    <head>
4      <title>Meu primeiro site</title>
5    </head>
6    <body>
7
8    </body>
9    <footer>
10
11    </footer>
12 </html>
```

# Tags HTML Comuns

## Cabeçalho

Lorem ipsum dolor sit amet consectetur adipisicing elit. Alias iure nisi labore incidunt, rem est distinctio illo laudantium repellat commodi quod reprehenderit odit eligendi dolores eaque. Temporibus molestias enim vero.



- Cabeçalhos: `<h1>`, `<h2>`, `<h3>`
- Parágrafos: `<p>`
- Formatação `<strong>`, `<em>`, `<br>`, `<hr>`
- Links: `<a href="URL">Texto do Link</a>`
- Imagens: ``



# Exercício - 01

- Criar uma página html
  - Nome: index.html
  - Estrutura básica
  - Título da página 'Hello world'
  - Título de nível 3 ('Minha primeira página HTML')
  - Um parágrafo com um texto de 10 linhas (coloque palavras em negrito no meio do texto)
  - Link para um site qualquer
  - Adicione uma imagem

# Listas em HTML

- ☐ Levar cachorro para passear
- ☐ Tirar o lixo

1. ☐ Levar cachorro para passear
2. ☐ Tirar o lixo

- Listas ordenadas: `<ol>`
- Listas não ordenadas: `<ul>`

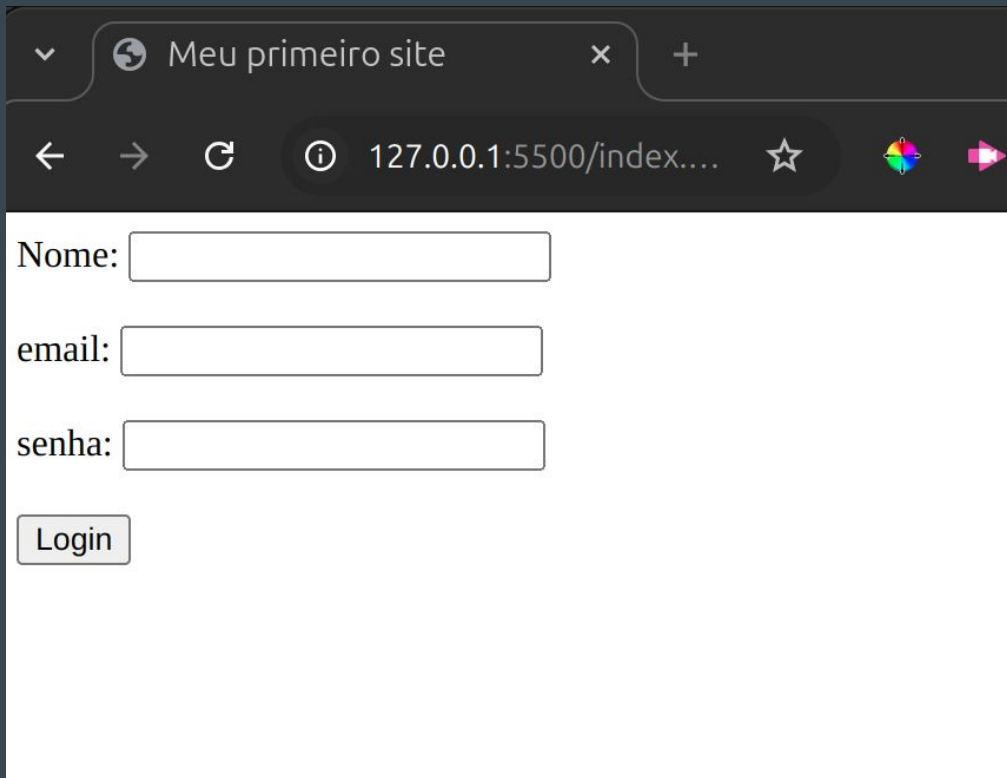
# Formulários

## <form> attributes

Attribute	Value
action	URL to where the form data will be submitted. if blank, it sends to the current page
method	How data is sent: GET or POST
enctype	Encoding Type; the format of the data. Default= application/x-www-form-urlencoded
target	Anchor target (windows / frames) same as <a>'s target
name	Name of form (can be used to refer to form in javascript.)
id	Optional: for CSS or javascript

- Um Formulário é uma seção da página HTML que permitem ao leitor introduzir dados

# Trabalhando com Inputs em HTML



A screenshot of a web browser window. The address bar shows 'Meu primeiro site' and the URL '127.0.0.1:5500/index...'. The page content is a simple login form with three input fields labeled 'Nome:', 'email:', and 'senha:'. Below the fields is a 'Login' button.

Nome:

email:

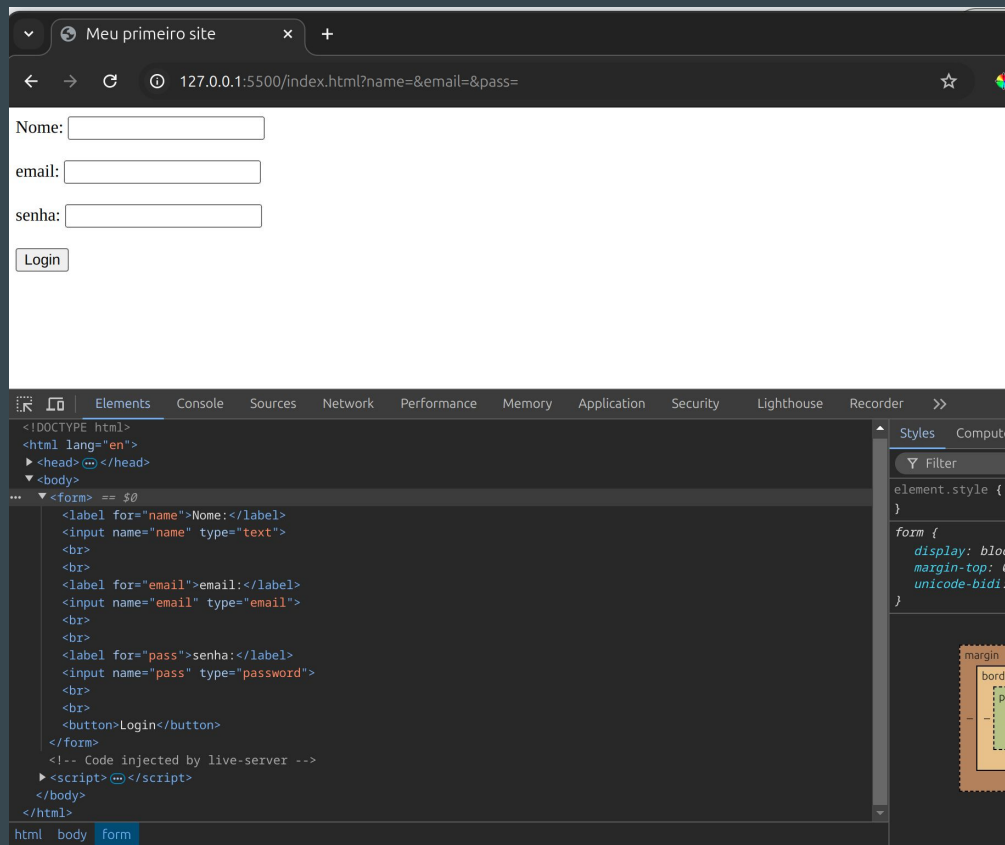
senha:

- Elementos de formulário
  - `<input>`: Campo de entrada de dados
  - `<label>`: Rótulo para o campo de entrada
  - `<textarea>`: Área de texto de várias linhas
  - `<button>`: Botão para enviar ou acionar ações
- Tipos comuns de input
  - texto
  - email
  - senha
  - submit

# Exercício - 02

- Criar uma nova página html
  - Nome: formulario.html
  - Estrutura básica
  - Título da página 'Formulário'
  - Título de nível 3 (Cadastro de Usuário)
  - Adicionar um formulário com os seguintes inputs
    - Nome
    - E-mail
    - Idade
    - Sexo
    - Estado civil
    - Profissão (select com as opções ('Programador', 'Analista', 'Designer'))
    - Aceite de políticas de privacidade
    - Observação (4 linhas e 40 colunas)

# DevTools

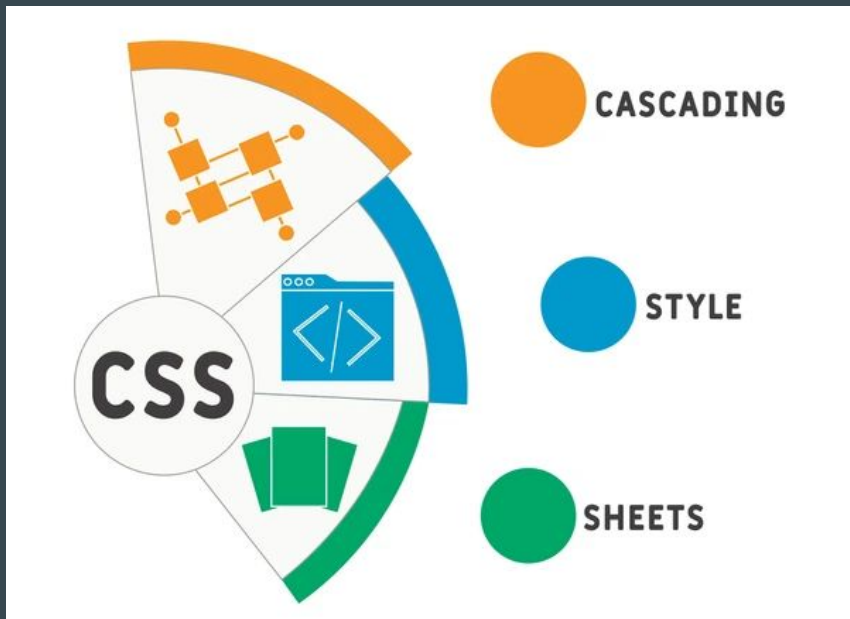


```
.color-transition {  
  background-color: lightblue;  
  transition: background-color 1s  
}  
  
.color-transition:hover {  
  background-color: white;  
}
```

A woman with dark hair, wearing a grey blazer and trousers, is sitting on the large blue letters 'CSS'. She is holding a paintbrush and a blue bucket, and is in the process of painting the word 'CSS' in a vibrant blue color. The background is a light grey gradient with faint, semi-transparent CSS code snippets. One snippet is highlighted with a blue rectangular box around the 'lightblue;' value.

# CSS

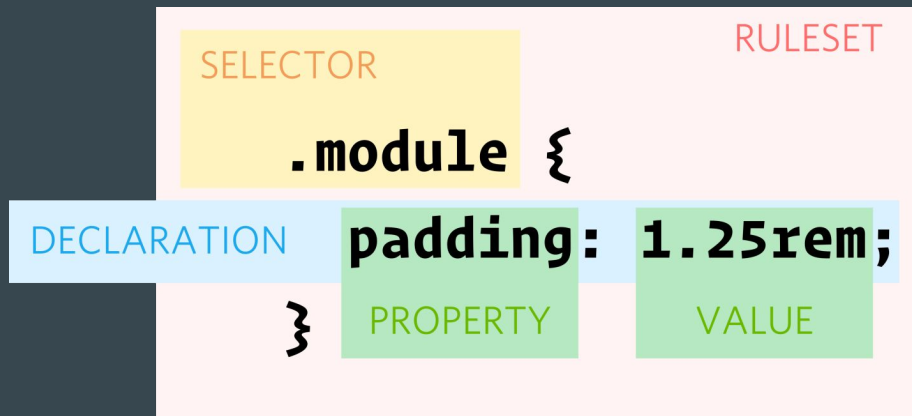
# O que é CSS?



- O termo cascata se refere à sequência ordenada de estilos de formatação do documento.
- As regras das folhas posteriores têm precedência sobre as anteriores.
- A ordem de estilos aplicados é:
  - 1. Uma folha de estilo externa;
  - 2. Uma folha de estilo interna;
  - 3. Estilos aplicados inline (diretamente ao elemento selecionado)



# sintaxe



- A sintaxe é composta por três partes: uma tag e ou seletor e uma declaração

# Adicionando css a uma página html

## Folha externa

```
<head>
|   <link href="estilo.css" rel="stylesheet" />
</head>
```

## Folha Interna

```
<head>
|   <style>
|   |   /* Estilo aqui */
|   </style>
</head>
```

## Estilo inline

```

<!-- Inline CSS Styles -->

<h1 style="font-weight: bold; font-size: 42px;">
  I will be bold and 42 pixels!
</h1>

<p style="color: blue;">I will be blue!</p>
```

# Seletores CSS

## Seletor de elemento

Seleciona elementos HTML

Com base no nome desses elementos

`p { ... }`

## Seletor de id

Seleciona um elemento HTML específico

Com base no atributo id desse elemento

`#id { ... }`

## Seletor de classe

Seleciona elementos HTML

Com base no atributo de classe

`.classe { ... }`

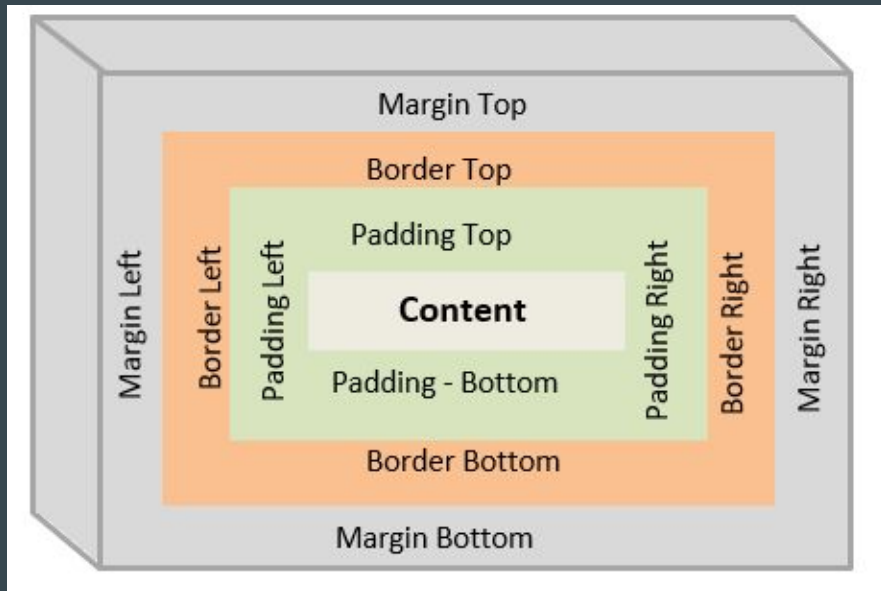
- Seletores são usados para "encontrar" elementos HTML que você deseja estilizar.
- Eles permitem aplicar estilos de forma específica.

# Propriedades básicas de estilo

```
1  .classe-color {
2    color: black;
3  }
4
5  .classe-background {
6    background-color: blueviolet;
7  }
8
9  .classe-font-family-size {
10   font-family: Arial, Helvetica, sans-serif;
11   font-size: 12px;
12 }
13
14 .classe-margin {
15   margin: 10px;
16 }
17
18 .classe-padding {
19   padding: 10px;
20 }
21
22 .classe-border {
23   border: 1px solid black;
24 }
```

- color: Define a cor do texto.
- background: Define o fundo de um elemento.
- font-family: Define a fonte do texto.
- font-size: Define o tamanho da fonte.
- margin: Define o espaçamento externo de um elemento.
- padding: Define o espaçamento interno de um elemento.
- border: Define a borda ao redor do elemento.

# Box Model



- **Content:** O conteúdo real do elemento, como texto ou imagens.
- **Padding:** Espaçamento entre o conteúdo e a borda do elemento.
- **Border:** A linha ao redor do conteúdo e padding.
- **Margin:** Espaço externo entre o elemento e outros elementos.

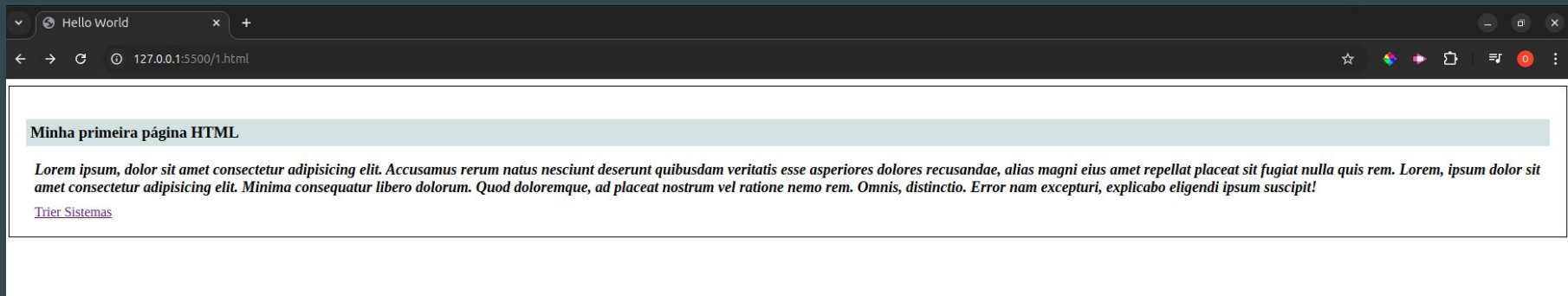
# Unidades de medida



- px: Pixels. Unidade fixa, ideal para elementos com dimensões precisas.
- %: Percentual em relação ao elemento pai.
- em: Relativo ao tamanho da fonte do elemento pai.
- rem: Relativo ao tamanho da fonte raiz (geralmente <html>).
- vh/vw: Unidades relativas à altura e largura da janela.

# Exercício - 03

- Utilizando nossa primeira página html criada
  - Adicione o css externo
  - Crie classes para estilizar o título, o texto e o link
  - Tente reproduzir o exemplo

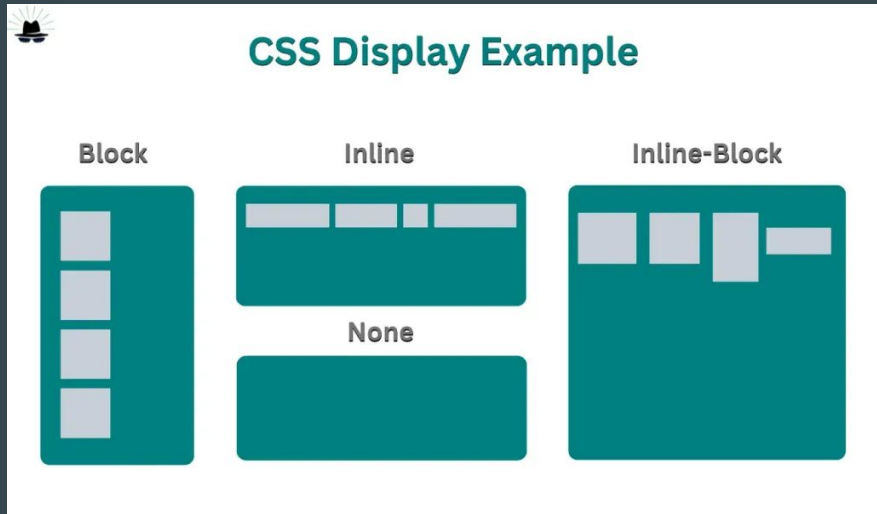


# Layouts responsivos



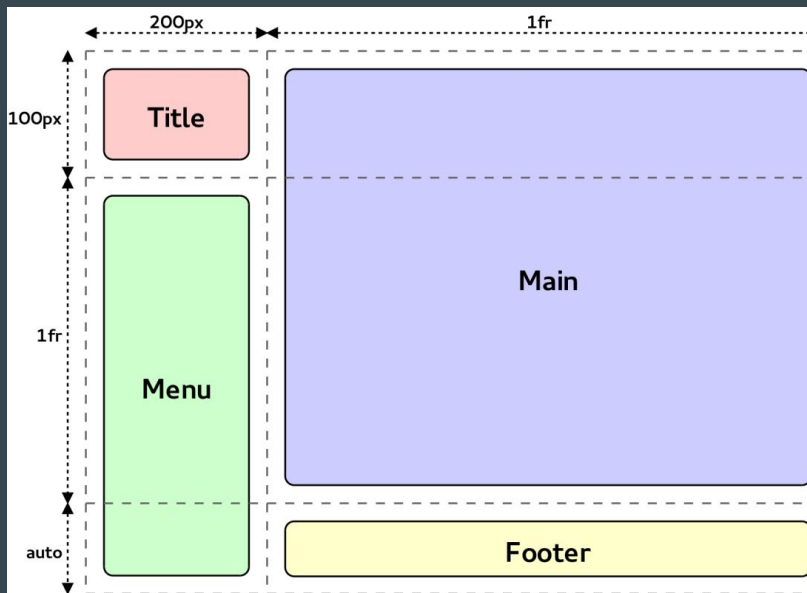


# Layouts



- **Block:** Elementos de bloco ocupam toda a largura disponível e sempre começam em uma nova linha.  
`<div>`, `<h1>`, `<p>`, `<section>`
- **Inline:** Elementos inline ocupam apenas a largura necessária e não iniciam em uma nova linha.  
`<span>`, `<a>`, `<strong>`, `<em>`
- **Inline-block:** Combina características de elementos inline e de bloco. Ocupa apenas o espaço necessário, mas respeita as propriedades de largura e altura.
- **None:** O elemento não é exibido na página e não ocupa espaço algum.

# Grid layout



- `display: grid;`
- `grid-template-columns` – Define o tamanho das minhas colunas;
- `grid-template-rows` – Define o tamanho das minhas linhas;
- `grid-template-areas` – Define as áreas da grid;
- `grid-area` – relaciona o elemento com a área da grid

```

24 .box-a {
25   border: 1px solid black;
26   grid-area: box-a;
27   background-color: rgb(168, 168, 206);
28 }
29
30 .box-b {
31   border: 1px solid black;
32   grid-area: box-b;
33   background-color: rgb(162, 206, 184);
34 }
35
36 .box-c {
37   border: 1px solid black;
38   grid-area: box-c;
39 }
40
41 .grid-layout {
42   display: grid;
43   grid-template-columns: 100px auto;
44   grid-template-rows: 50px auto;
45   grid-template-areas:
46     "box-a box-a"
47     "box-b box-c";
48 }

```

Logo

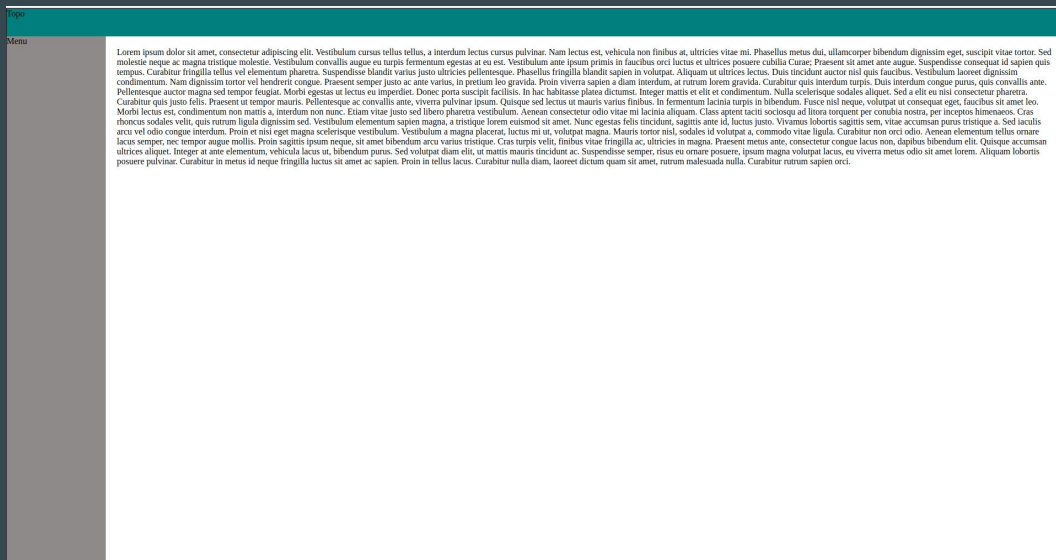
Menu

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum cursus tellus tellus, a interdum lectus cursus pulvinar. Nam lectus est, vehicula non finibus at, ultricies vitae mi. Phasellus metus dai, ullamcorper bibendum dignissim eget, suscipit vitae tortor. Sed molestie neque ac magna tristique molestie. Vestibulum convallis augue eu turpis fermentum egetas at eu est. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Praesent sit amet ante augue. Suspendisse consequat id sapien quis tempus. Curabitur fringilla tellus vel elementum pharetra. Suspendisse blandit varius justo ultricies pellentesque. Phasellus fringilla blandit sapien in volutpat. Aliquam ut ultrices lectus. Duis tincidunt auctor nisl quis faucibus. Vestibulum laoreet dignissim condimentum. Nam dignissim tector vel hendrerit congue. Praesent semper justo ac ante varius, in pretium leo gravida. Proin viverra sapien a diam interdum, a rutrum lorem gravida. Curabitur quis interdum turpis. Duis interdum congue purus, quis convallis ante. Pellentesque auctor magna sed tempor feugiat. Morbi egetas ut lectus eu imperdiet. Donec porta suscipit facilisis. In hac habitasse platea dictumst. Integer mattis et elit et condimentum. Nulla scelerisque sodales aliquet. Sed a elit eu nisl consectetur pharetra. Curabitur quis justo felis. Praesent ut tempor mauris. Pellentesque ac convallis ante, viverra pulvinar ipsum. Quisque sed lectus ut mauris varius finibus. In fermentum lacinia turpis in bibendum. Fusce nisl neque, volutpat ut consequat eget, faucibus sit amet leo. Morbi lectus est, condimentum non mattis a, interdum non nunc. Etiam vitae justo sed libero pharetra vestibulum. Aenean consectetur odio vitae mi lacinia aliquam. Class aptent taciti sociosque ad litora torquent per conubia nostra, per inceptos himenaeos. Cras rhoncus sodales velit, quis rutrum ligula dignissim sed. Vestibulum elementum sapien magna, a tristique lorem euismod sit amet. Nunc egetas felis tincidunt, sagittis ante id, luctus justo. Vivamus lobortis sagittis sem, vitae accumsan purus tristique a. Sed lacinis arcu vel odio congue interdum. Proin et nisl eget magna scelerisque vestibulum. Vestibulum a magna placerat, luctus mi ut, volutpat magna. Mattis tortor nisl, sodales id volutpat a, commodo vitae ligula. Curabitur non orci odio. Aenean elementum tellus ornare lacus semper, nec tempor augue mollis. Proin sagittis ipsum neque, sit amet bibendum arcu varius tristique. Cras turpis velit, finibus vitae fringilla ac, ultricies in magna. Praesent metus ante, consectetur congue lacus non, dapibus bibendum elit. Quisque accumsan ultrices aliquet. Integer at ante elementum, vehicula lacus ut, bibendum purus. Sed volutpat diam elit, ut mattis mauris tincidunt ac. Suspendisse semper, risus eu ornare posuere, ipsum magna volutpat lacus, eu viverra metus odio sit amet lorem. Aliquam lobortis posuere pulvinar. Curabitur in metus id neque fringilla luctus sit amet ac sapien. Proin in tellus lacus. Curabitur nulla diam, laoreet dictum quam sit amet, rutrum malesuada nulla. Curabitur rutrum sapien orci.



# Exercício - 04

- Crie o layout da página anterior
  - Adicione 3 botões na seção de menu (Cadastro de usuário, Listagem de usuário e Menu Principal)
  - Adicione uma logo e um título na seção correspondente
  - No conteúdo adicione apenas um texto



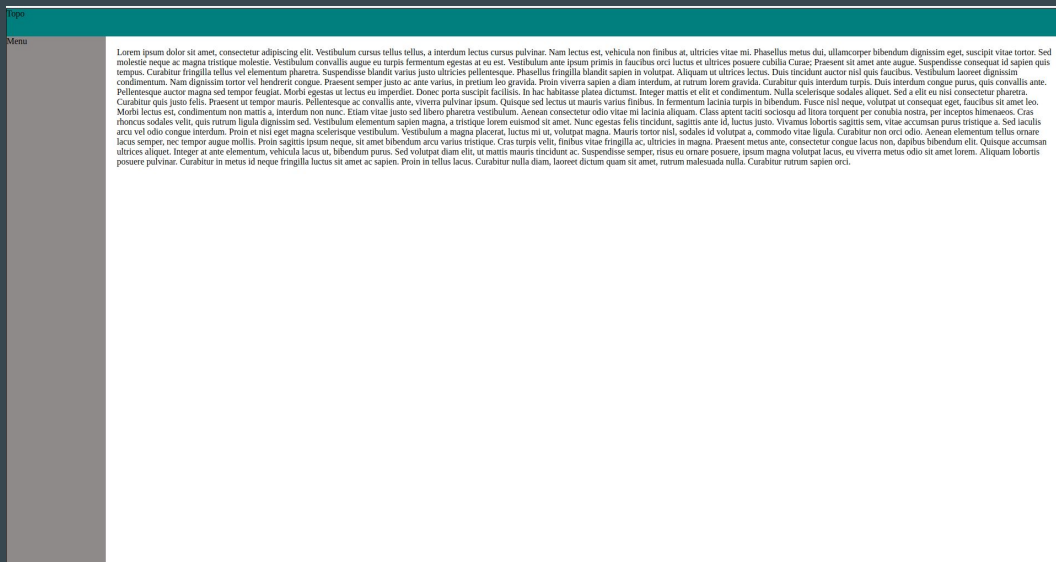
# Flexbox layout



- `display: flex;`
- `flex-direction`: define a direção dos itens (row, column)
- `justify-content`: alinha os itens horizontalmente dentro do container.
- `align-items`: alinha os itens verticalmente dentro do container.
- `flex-wrap`: Controla como os itens flexíveis são distribuídos em várias linhas quando o espaço disponível no container é insuficiente.

# Exercício - 05

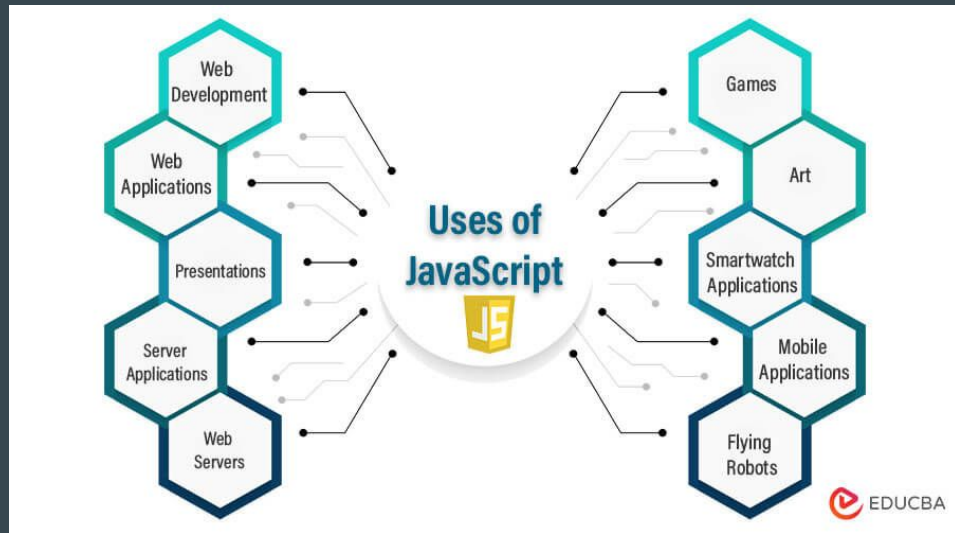
- Refaça o exercício 4 com o Flexbox layout



# Javascript



# O que é Javascript?



- JavaScript é uma linguagem de programação usada para criar interatividade e dinamismo em páginas web.
- Funções Principais:
  - Manipulação do DOM
  - Respostas a eventos do usuário
  - Comunicação com servidores (AJAX)
  - Validação de formulários

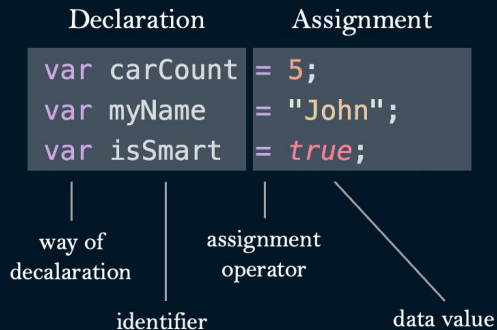


# Organização da linguagem

The image shows the letters 'JS' in a bold, dark blue font, centered on a bright yellow rectangular background.

- Sentença de código, terminada ou não com ";"
- Bloco de códigos:
  - Vazio;
  - Function;
  - Classe;
  - Loop de repetição;
  - Entre outros.

# Declaração de variáveis JS



- `var`
  - Declara uma variável que tem escopo de função ou escopo global (não possui escopo de bloco).
  - Pode ser redeclarada e reatribuída.
- `let`
  - Declara uma variável com escopo de bloco (ou seja, a variável é visível apenas dentro do bloco onde foi declarada).
  - Pode ser redeclarada e reatribuída dentro do mesmo escopo de bloco
- `const`
  - Declara uma variável com escopo de bloco, como `let`.
  - O valor atribuído a uma variável `const` não pode ser reatribuído.

# Tipagem fraca e dinâmica

```
JS script.js > ...
```

```
1 let variavel = "teste"  
2 variavel = 2  
3 variavel = true  
4 variavel = 2.2
```

- JavaScript é uma linguagem com tipos dinâmicos. As variáveis em JavaScript não estão diretamente associadas a nenhum tipo de valor específico, e qualquer variável pode receber (e reatribuir) valores de todos os tipos.

# Tipos em JS

## Tipos de Dado Primitivos

String `"hello world"` `'abacaxi'` ``5 patinhos``

Number `20` `3.1415` `-18` `-9.04` `Infinity` `NaN`

Boolean `true` `false`

Object `{}`

Outros `null` `undefined`

- Valores primitivos
  - string
  - number
  - boolean
  - object
  - array
  - null
  - undefined

# Ferramenta



Runner vscode

## Exercício - 06

- Crie um arquivo javascript e realize a seguinte operação matemática
  - $(a+b)/c$
  - Onde  $a = 40$ ,  $b = 5$  e  $c = 2$
  - O resultado deve ser impresso em duas casas decimais

## Exercício - 07

- Criar um arquivo javascript, onde eu tenha a seguinte operação:
  - Uma constante onde eu possa armazenar o nome "Aula";
  - Uma constante onde eu possa armazenar o nome "Frontend";
  - Imprimir no console utilizando as duas constantes com a seguinte frase: "Aula de Frontend";
  - Imprimir no console as 2 primeiras letras da primeira constante;

# Função

```
js scriptjs > ...
1  function soma(a, b) {
2    |   return a + b;
3  }
4
5  function divide(a, b) {
6    |   return a / b;
7  }
8
9  const resultadoSoma = soma(6, 8)
10 const resultadoDivisao = divide(6, 8)
11
12 console.log(resultadoSoma)
13 console.log([resultadoDivisao])
```

- Função é um trecho de código que executa determinada ação;
- Pode receber parâmetros;
- Pode ter um retorno;
- Pode ser anônima ou com nome;

# Exercício - 08

- Converta os exercícios 06 e 07 para utilizar função



# Operadores

```
JS script.js > ...
1 // Operadores Lógicos Simples
2 let a = true;
3 let b = false;
4 console.log(a && b); // false, ambos devem ser true
5 console.log(a || b); // true, ao menos um deve ser true
6 console.log(!a); // false, inverte o valor de a
7
8 // Operadores Unários
9 let numero = 10;
10 let negativo = -numero;
11 console.log(`Negativo de ${numero} é ${negativo}`);
12
13 let verdadeiro = true;
14 let falso = !verdadeiro;
15 console.log(`Inverso de true é ${falso}`);
16
17 let contador = 0;
18 contador++;
19 console.log(`Contador incrementado: ${contador}`);
20
21 contador--;
22 console.log(`Contador decrementado: ${contador}`);
23
24 let texto = "Olá";
25 console.log(`O tipo de "${texto}" é ${typeof texto}`);
26
27 console.log(void 0);
28
29 // Operador Ternário
30 let idade = 18;
31 let podeDirigir = (idade >= 18) ? 'Sim, pode dirigir' : 'Não, ainda é menor';
32 console.log(podeDirigir);
33
34 let saldo = 50;
35 let statusSaldo = saldo > 0 ? 'Saldo positivo' : 'Saldo insuficiente';
36 console.log(statusSaldo);
37
38 let statusNegativo = negativo < 0 ? 'Número negativo' : 'Número positivo';
39 console.log(statusNegativo);
40
```

- Lógicos
- Unários
- Ternários

# Listas em JS

## Searches in current Array:

[▲●■●].find(□===●) → ●

[▲●■●].findIndex(□===■) → 2

[▲●■●].indexOf(●) → 1

[▲●■●].some(□===■) → true

[▲●■●].every(□===●) → false

[▲●■●].includes(■) → true

<https://igorgo.nl>

- Declaração, Pode se utilizar colchetes ou o construtor Array
  - `let numeros = [1, 2, 3, 4, 5];`
  - `let frutas = new Array("maçã", "banana", "laranja");`
- `push()`: Adiciona elementos no final do array.
- `unshift()`: Adiciona elementos no início do array.
- `pop()`: Remove o último elemento do array.
- `shift()`: Remove o primeiro elemento do array.

# Loops

js script.js > ...

```
1  for (let index = 0; index < 10; index++) {  
2    console.log(index);  
3  }  
4  
5  let j = 10;  
6  while (j > 0) {  
7    j = --j;  
8    console.log(`J: `, j)  
9  }
```

- For
  - for (let I = 0; I > 10; I++) {}
- While
  - while (I < 10) {I++;}

# Objetos em JS

**{Object}**

**JS**

# Exercício - 09

- Gerenciamento de Alunos
  - Perguntar quantos alunos tem na sala
  - Adicionar n alunos (Perguntando o nome via prompt).
  - Perguntar quantas notas o aluno possui
  - Registrar notas para o aluno.
  - Calcular a média das notas (Utilizando arrow function) de cada aluno.
  - Mostrar quais alunos estão aprovados (média  $\geq 7$ ) e reprovados (média  $< 7$ ). (utilizar filter e arrow function)

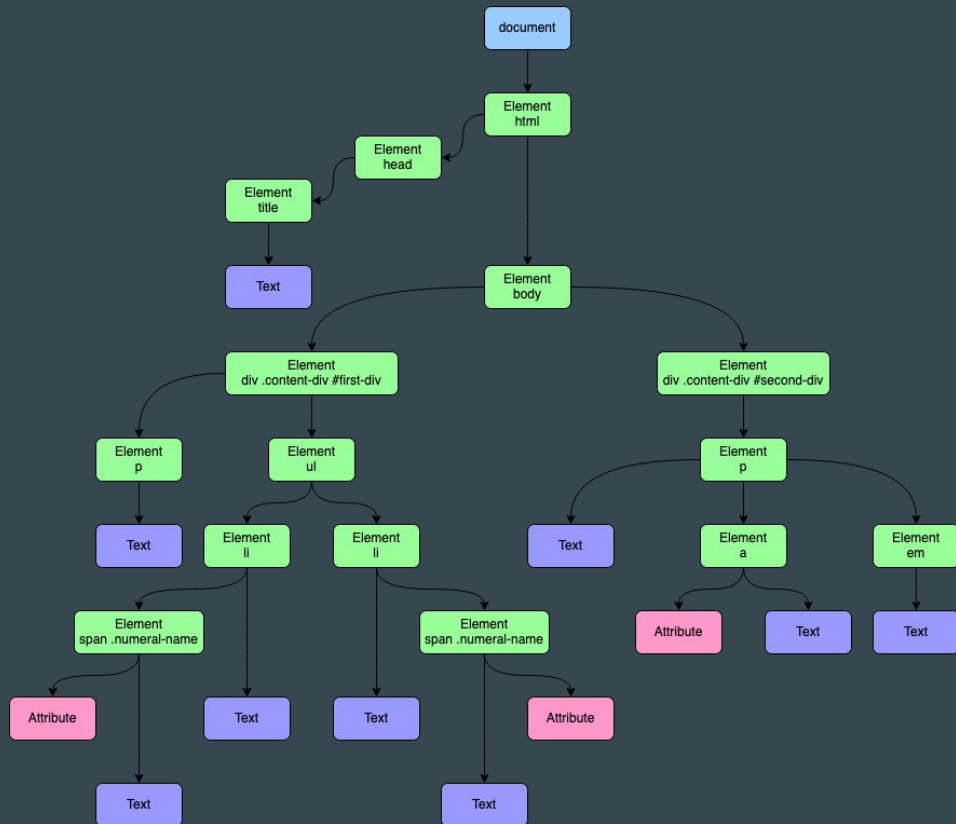
# Eventos

- Eventos de Mouse:
  - click: Disparado quando o usuário clica em um elemento.
  - dblclick: Disparado quando o usuário dá um duplo clique.
  - mouseover: Disparado quando o usuário passa o mouse sobre um elemento.
  - mouseout: Disparado quando o usuário tira o mouse de cima de um elemento.
  - mousedown: Disparado quando o botão do mouse é pressionado.
  - mouseup: Disparado quando o botão do mouse é solto.
- Eventos de Teclado:
  - keydown: Disparado quando uma tecla é pressionada.
  - keyup: Disparado quando uma tecla é solta.
  - keypress: Disparado enquanto uma tecla está sendo pressionada (depreciado, use keydown e keyup).
- Eventos de Formulário:
  - submit: Disparado quando um formulário é enviado.
  - change: Disparado quando o valor de um campo de formulário é alterado.
  - focus: Disparado quando um elemento recebe foco.
  - blur: Disparado quando um elemento perde o foco.



# The DOM Manipulation

# DOM



- Document Object Model (DOM) é a representação estruturada de um documento HTML.
- O DOM é uma árvore de nós que pode ser manipulada via JavaScript.
- Cada elemento HTML é um objeto no DOM.



# Acessando o DOM

```
5  
6  💡  
7  const element = document.getElementById('grid');  
8  const allLabels = document.querySelectorAll('label');
```

- document.getElementById('id')
- document.getElementsByClassName('class')
- document.getElementsByTagName('tag')
- document.querySelector('selector')
- document.querySelectorAll('selector')

# Manipulando a DOM



- Alterando o Conteúdo de Elementos
  - Propriedade `innerHTML` para alterar o conteúdo.
  - Propriedade `textContent` para alterar o texto.
- Modificando Estilos no DOM
  - `document.getElementById('label').style.color = 'pink';`
- Adicionando e Removendo classes
  - `classList.add('nova-classe')`
  - `classList.remove('classe')`
  - `classList.toggle('classe')`
- Criando e Removendo Elementos no DOM
  - `createElement()` para criar novos elementos.
  - `appendChild()` para adicionar.
  - `removeChild()` para remover.

# Exercício - 10

- TODO LIST
  - O usuário deve ser capaz de adicionar novos itens à lista de tarefas.
  - Cada item adicionado deve ter um botão de "remover" e um botão de "marcar como concluído".
  - Ao clicar em "marcar como concluído", o item deve mudar de cor e ser tachado, indicando que foi concluído.
  - O botão "remover" deve remover o item da lista.
  - Adicione validação para impedir que o usuário adicione tarefas vazias.
  - Adicione um botão "Limpar Tarefas Concluídas", que remove todos os itens que foram marcados como concluídos.
  - Crie uma contagem dinâmica de quantas tarefas ainda estão pendentes e quantas já foram concluídas.