

# DM121 – Introdução ao desenvolvimento Web: HTML5, CSS3 e Javascript

Elton Barbosa Rodrigues  
elton.barbosa@inatel.br

# SOBRE A DISCIPLINA

- ✓ Teórica e prática
- ✓ Material de apoio: Apostila, slides, exemplos em sala de aula
- ✓ Pré-requisitos: N/A
- ✓ Trabalho único que irá compor a nota e conceito.

# AGENDA

**Aula I:** Introdução ao desenvolvimento Web, HTML (Estrutura de conteúdos, semântica, principais tags), CSS (Sintaxe, posicionamento, grids), Layout Responsivo, Javascript (Sintaxe básica, funções, objeto literal, controle de seleção, estruturas de repetição), DOM API (Criação de elementos, manipulação de estilos, eventos), ES6, ES7, ES8 e além, atividades práticas.

# DURANTE A AULA

- ✓ Pergunte
- ✓ Troque experiências
- ✓ Participe
- ✓ Ajude

# INTRODUÇÃO

- ✓ Cliente X Servidor
- ✓ Baseado em HTTP
- ✓ Camada cliente, o navegador

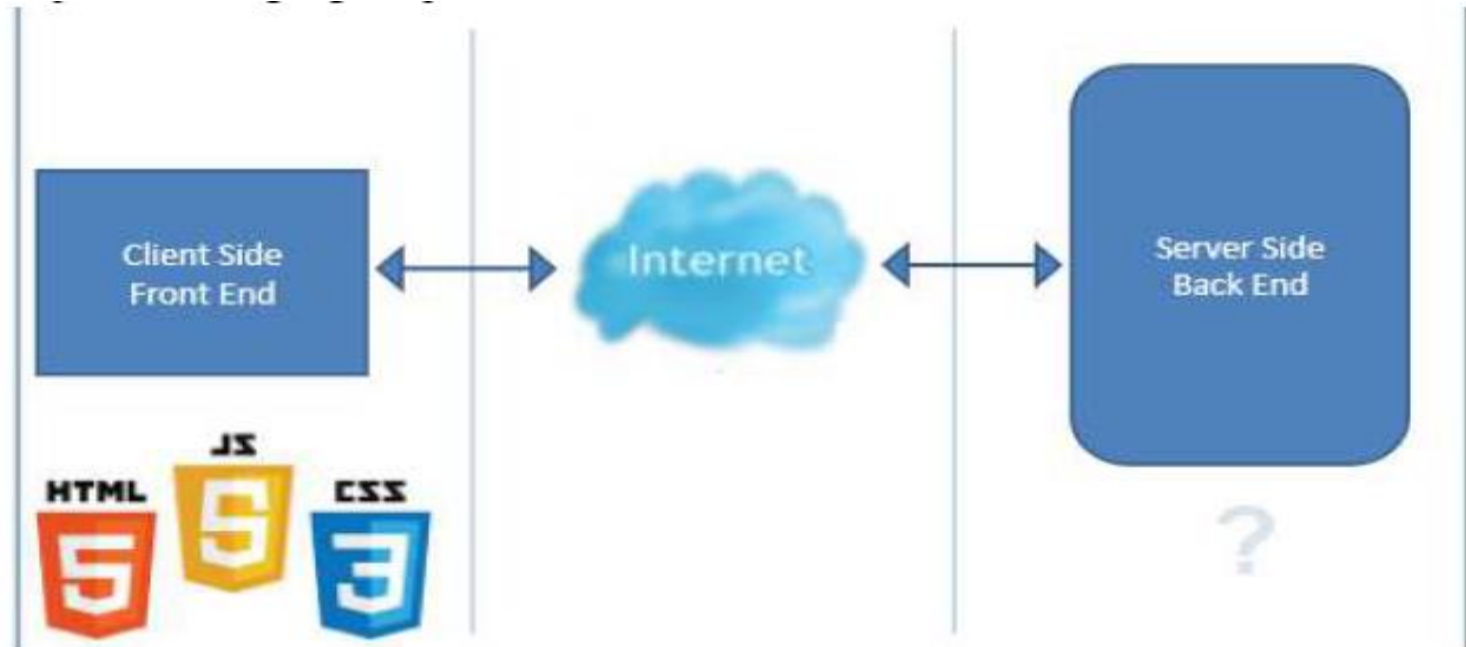
# INTRODUÇÃO



# DESENVOLVIMENTO WEB

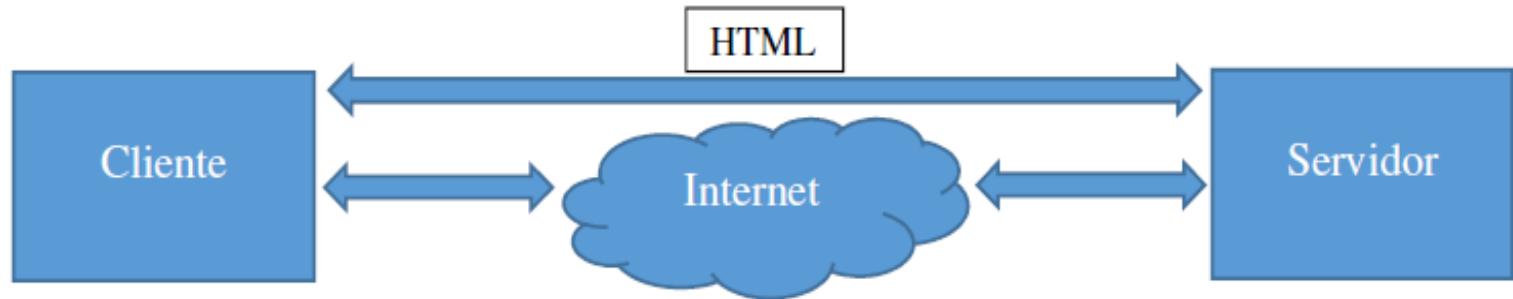
- ✓ Front-end
- ✓ Back-end
- ✓ Múltiplos dispositivos
- ✓ Motores de busca
- ✓ Frameworks, bibliotecas são lançadas a todo momento

# DESENVOLVIMENTO WEB





# HTML



# HTML

- ✓ Baseado em tags
- ✓ Versões: HTML 1.0, HTML 2.0, HTML 3.0, HTML 3.2, HTML 4.01, XHTML 1.0, **HTML 5**

# HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>

</body>
</html>
```

# HTML

- Metadata content - **Base Command Link Meta noscript Script, style, title**
- Flow content - **article aside audio b bdo blockquote br button**
- Sectioning content - **header footer article aside nav**
- Heading content - **h1 até h6 hgroup header**
- Phrasing content - **br button canvas code command datalist em embed**
- Embedded content - **audio canvas embed iframe img math object svg vídeo**
- Interactive content - **a audio button details embed iframe img video**

# HTML

## Tags estruturais com significado semântico

```
<section id="rock">  
  <h2>Rock</h2>  
</section>
```

```
<section id="blues">  
  <h2>Blues</h2>  
</section>
```

```
<section id="jazz">  
  <h2>Jazz</h2>  
</section>
```

# HTML

## Tags estruturais com significado semântico

```
<nav>
  <h3>Esportes</h3>
  <ul>
    <li><a href="#">Futebol</a></li>
    <li><a href="#">Natação</a></li>
    <li><a href="#">Vôlei</a></li>
    <li><a href="#">Basket</a></li>
  </ul>
  <h3>Educação</h3>
  <ul>
    <li><a href="#">Português</a></li>
    <li><a href="#">Matemática</a></li>
  </ul>
  <h3>Notícias</h3>
  <ul>
    <li><a href="#">Esporte</a></li>
    <li><a href="#">Política</a></li>
    <li><a href="#">Tecnologia</a></li>
  </ul>
</nav>
```

# HTML

## Tags estruturais com significado semântico

```
<article>
  <h1>HTML 5</h1>
  <p>
    Lorem ipsum dolor sit amet consectetur adipisicing elit.
    Ea architecto totam ex recusandae laboriosam facere quod ducimus
  </p>
</article>
```

# HTML

## Tags estruturais com significado semântico

```
<aside id="menu-lateral">
  <nav>
    <h3>Esportes</h3>
    <ul>
      <li><a href="#">Futebol</a></li>
      <li><a href="#">Natação</a></li>
      <li><a href="#">Vôlei</a></li>
      <li><a href="#">Basket</a></li>
    </ul>
    <h3>Educação</h3>
    <ul>
      <li><a href="#">Português</a></li>
      <li><a href="#">Matemática</a></li>
    </ul>
    <h3>Notícias</h3>
    <ul>
      <li><a href="#">Esporte</a></li>
      <li><a href="#">Política</a></li>
      <li><a href="#">Tecnologia</a></li>
    </ul>
  </nav>
</aside>
```



# HTML

## Tags estruturais com significado semântico

```
<footer>  
    <p>Inatel DM121, Copyright &copy; 2018</p>  
</footer>
```

# ESTUDO DIRIGIDO 1 – ESTRUTURANDO PÁGINAS COM HTML

Inatel DM121	Home	Serviços	Sobre
--------------	------	----------	-------

## Affordable Inatel DM121

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam eu luctus ipsum, rhoncus semper magna.

Nulla nec magna sit amet sem interdum condimentum.

### Inscriva-se em nossa newsletter

Inscriva

#### HTML5 Markup

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus mi augue, viverra sit amet ultricies

#### CSS3 Styling

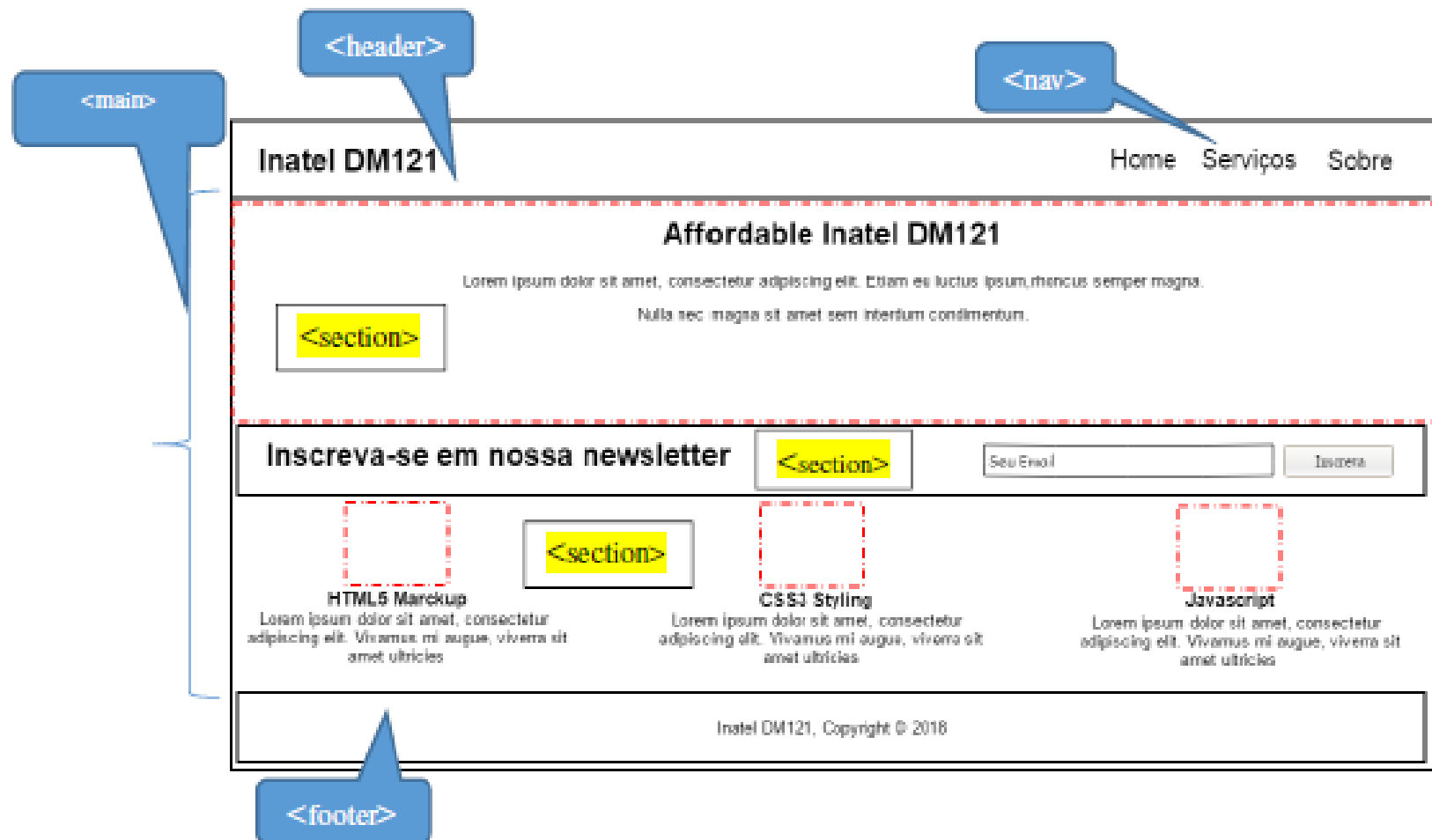
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus mi augue, viverra sit amet ultricies

#### Javascript

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus mi augue, viverra sit amet ultricies

Inatel DM121, Copyright © 2018

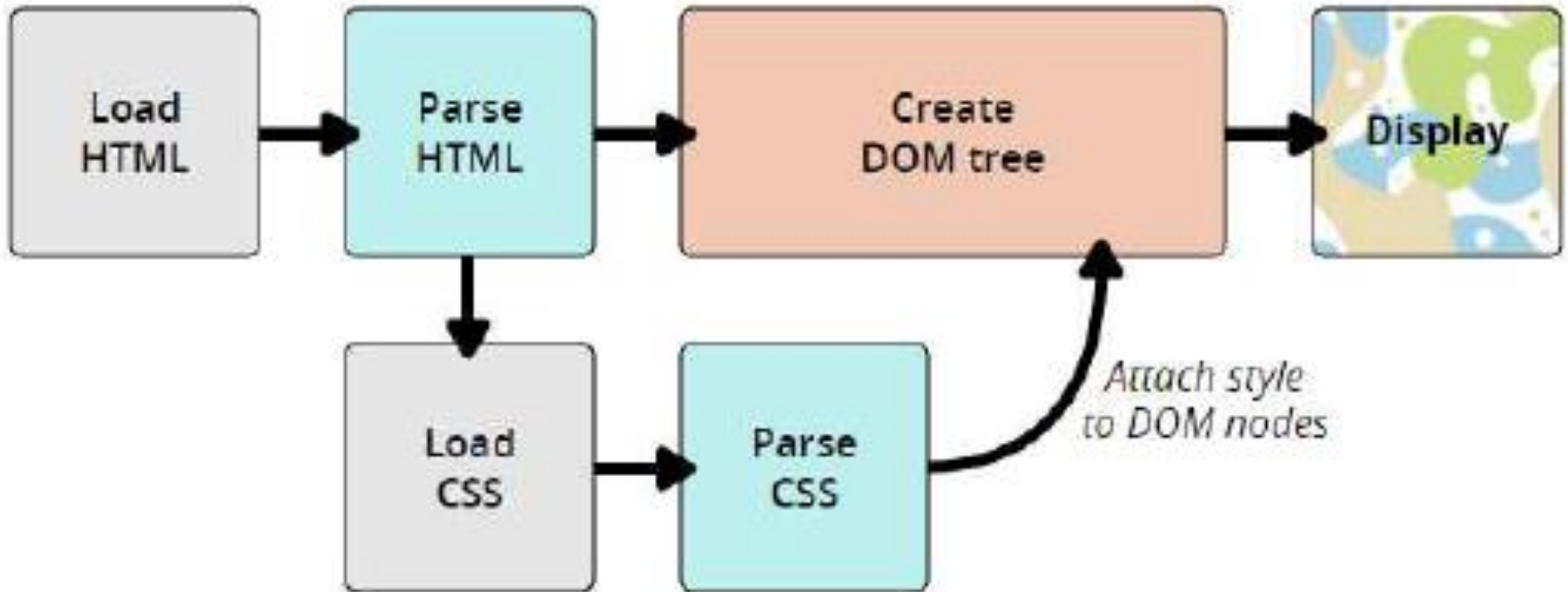
# ESTUDO DIRIGIDO 1 – ESTRUTURANDO PÁGINAS COM HTML



# CSS

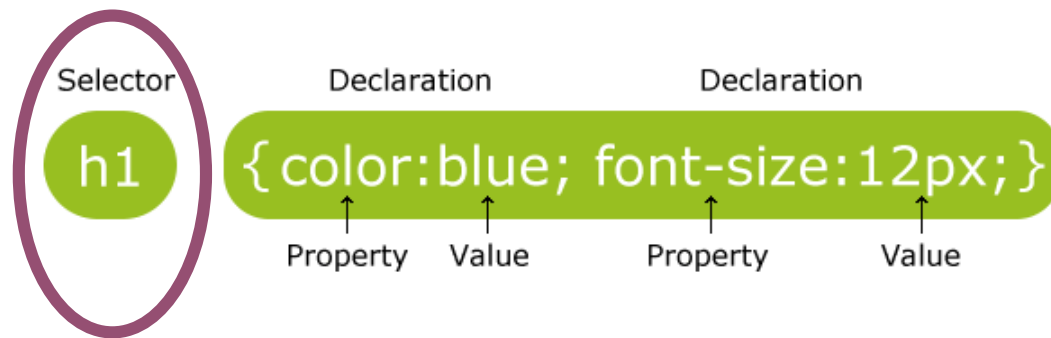
- ✓ CSS - Cascading Style Sheets
- ✓ Funciona em conjunto com HTML
- ✓ Baseada em declarações
- ✓ CSS 1, CSS 2, **CSS 3**, CSS 4 ([Última atualização Novembro 2018](#))

# CSS

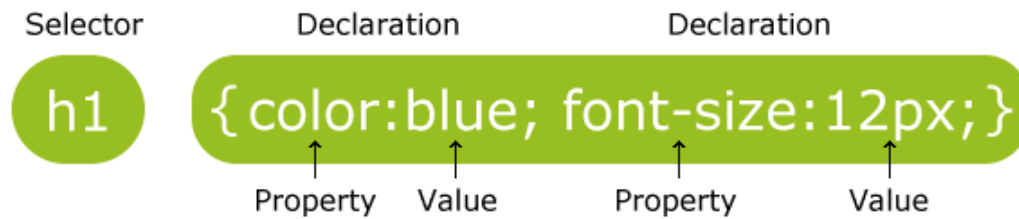


Fonte: <https://developer.mozilla.org>

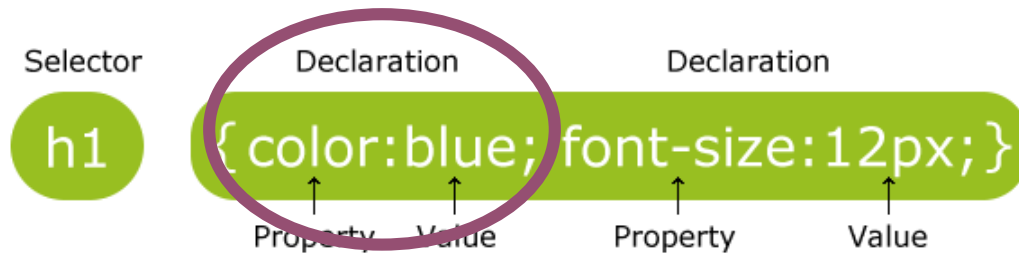
# Seletor



# Propriedade/Valor

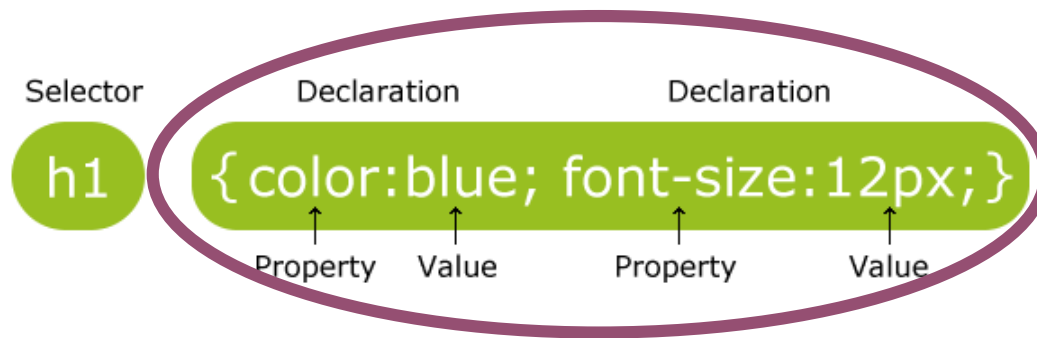


# Declaração

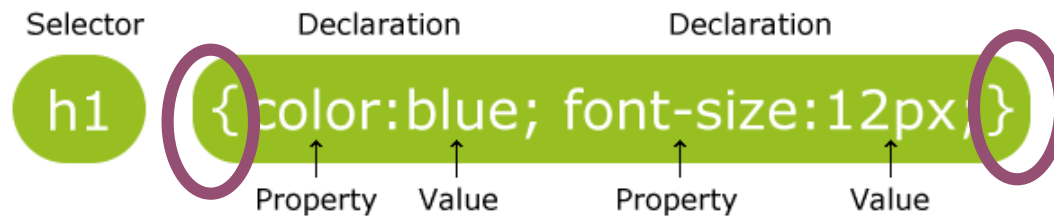




# Bloco de declaração



# Chaves



# Seletores de classe e id

id - Especifica um unico elemento

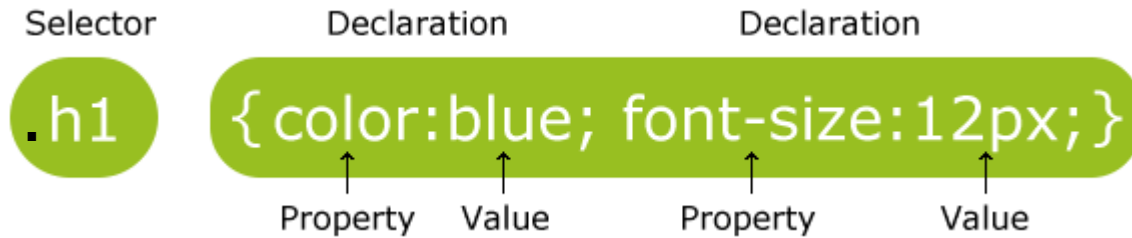
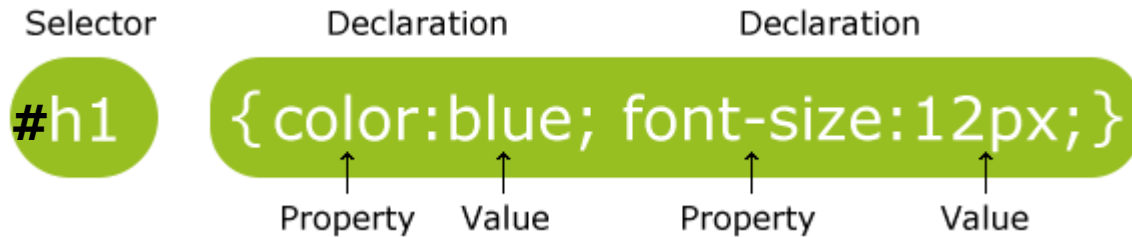
- É um atributo html e é definido no css com símbolo "#".
- O estilo abaixo será aplicado para a tag html cujo o ID é id="campo1":
- **#campo1 {text-align:center;color:red;}**

# Seletores de classe e id

**Class** - Utilizado para especificar um grupo de elementos

- É um atributo html e é definido no css com símbolo “ . ”.
- O estilo abaixo será aplicado para a tag html cujo a class é id=“campo”:
- **.campo {text-align:center;}**

# Seletores de classe e id



# Como aplicar CSS na página

- Inline
- Internal
- External style sheet

# Como aplicar CSS na página

## **Inline CSS**

Aplicado diretamente na tag HTML.

Exemplo:

```
<p style="color: red;"> Este parágrafo ficou vermelho.</p>
```

# Como aplicar CSS na página

## Interno

Aplicado diretamente na página HTML por uma tag chamada

`<style></style>`, entre essas tags ficam as declarações css.

Exemplo:

```
<head>
<title>my page</title>
<style type="text/css">
  p{color:red}</style>
</head>
<body>
<p>
  Isto é um parágrafo.
</p>
</body>
```



# Como aplicar CSS na página

## Externo

Arquivo externo com extensão .css. O arquivo é “linkado” à página HTML.

Exemplo:

```
<head>
```

```
<title>Linkando folha de estilo</title>
```

```
<link rel=“stylesheet” type=“text/css” href=“estilo.css”>
```

```
<body>
```

```
<Isto é um parágrafo.
```

```
</p>
```

```
</body>
```

# Formatando Textos

- **font-family**: Lista ordenada dos tipos de letra a usar.
- **font-size**: Tamanho das letras. Unidades possíveis: px, % ou em.
- **font-style**: Estilo das letras. Opções: normal ou italic.
- **font-weight**: Espessura das letras. Opções: lighter, normal, bold ou bolder.
- **text-decoration**: Efeitos a aplicar ao texto. Opções: none, underline, overline, line-through.

# Texto

- `text-align`: Alinhamento do texto. Opções: left, right, justify, center.
- `text-indent`: Alinhamento da primeira linha de um bloco de texto.
- `line-height`: Altura de cada linha num bloco de texto.
- `letter-spacing`: Espaçamento entre letras.
- `word-spacing`: Espaçamento entre palavras.

# Unidades

As principais unidades são as duas seguintes.

- **px** ( píxel ) — Os píxeis representam unidades absolutas, um pixel, corresponde a um pixel no ecrã do utilizador. Exemplos: 10px, 20px, 1px.
- **%** ( percentagem ) — As unidades percentuais representam unidades relativas, e são calculadas em relação ao contexto do elemento. O contexto varia em função do elemento e da propriedade. Exemplos: 1%, 100%, 5.5%.

## Unidades Relativas

em - Relativo ao font-size corrente

rem - Relativo ao font-size do elemento root

vw - Relativo a 1% do width do viewport

vh - Relativo a 1% do height do viewport

# Cores

Existem várias alternativas para definição de cores

Palavra-chave: black, yellow, red, blue, etc.

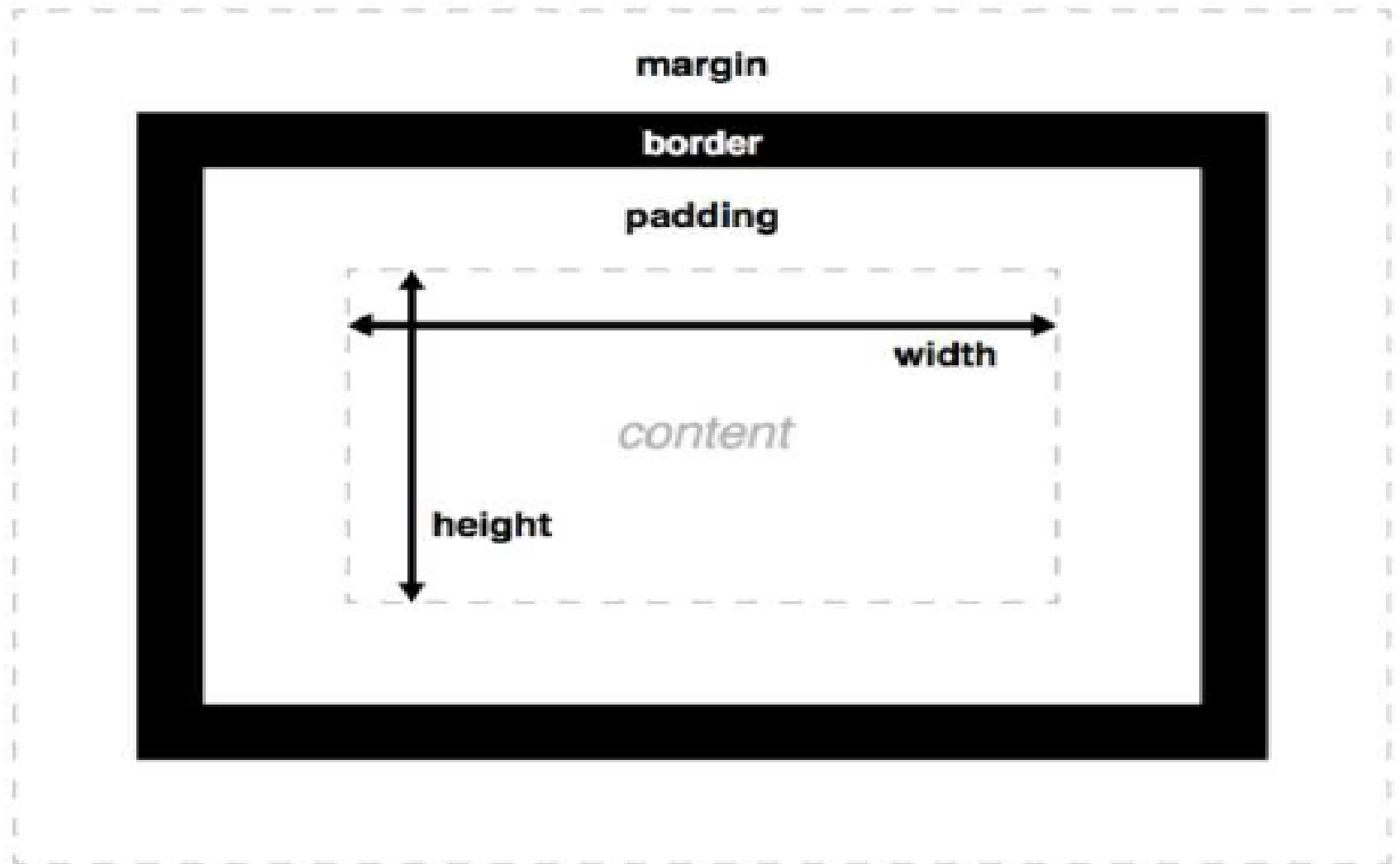
Código RGB: rgb(vermelho, verde, azul)

Exemplos: rgb(100%, 40%, 0%); rgb(255, 102, 0).

Código Hexadecimal: #código hexadecimal

Exemplos: #326432, #000000, #0088ff.

# Box Model



# Posicionamento em CSS

## FLOAT

A propriedade float aceita quatro valores:

**right:** Permitir flutar o elemento para direita

**left:** Permitir flutar o elemento para esquerda

**inherit:** Permitir flutar o elemento baseado no valor do float do elemento pai

**none:** Não fluta o elemento (Esse é o comportamento default nos navegadores)

*et adipisicing elit. Laboriosam quos voluptas autem quisquam quia ea optio rem quasi inventore mollitia eos libero unde, sed Dolonibus eius triciendis in voluptate fugiat saepe voluptatem, enim, porro ducimus dederunt repellat repellendus quidem maxime, se neque? Voluptatibus nulla neque corporis quae. Nobis, iure. Blanditiis, nisi quam? Vitae, aspernatur aperiam qui assumenda in beatae, corporis voluptas incidunt nisi quo tenetur quod maiores iure. Accusantium repellat ex est id illo accusamus ipsa soluta, perspiciatis voluptatum nostrum, autem excepturi?*

Div float

# Posicionamento em CSS

## POSITION

A propriedade position aceita quatro valores:

**static:** É o posicionamento padrão, posiciona o elemento no fluxo normal dos elementos na página.

**relative:** Tem comportamento semelhante ao static a diferença é que permite alterar a posição final do elemento, para isso é preciso usar as propriedades top, bottom, left, and right.

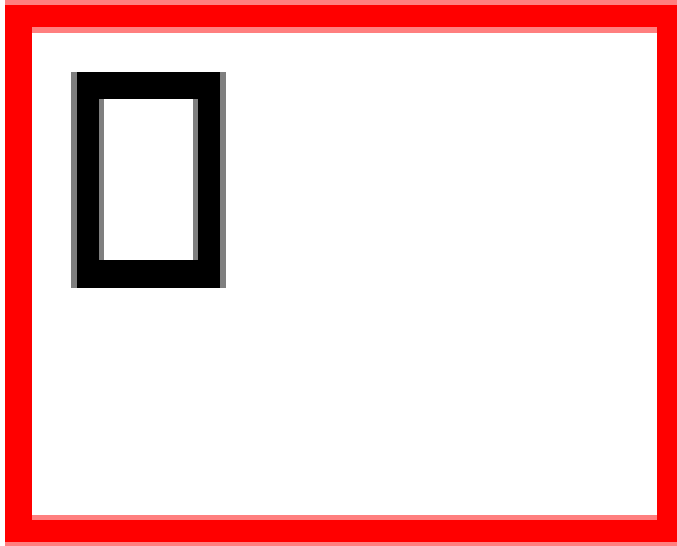
**absolute:** No posicionamento absolute a referência é a página e também é possível especificar a distancia dos lados do elemento utilizando as propriedades **top**, **bottom**, **left** e **right** em relação a página.

**fixed:** Possui comportamento semelhante ao absolute a diferença é que neste o elemento é fixado em sua posição e sua referência é a tag **html** ou seu ancestral mais próximo.



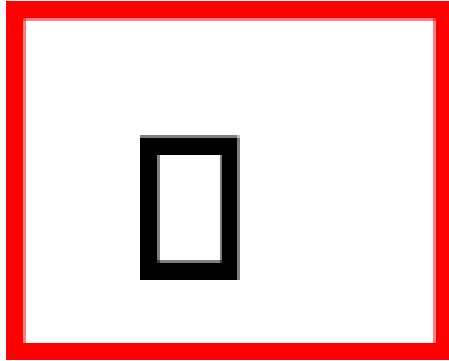
# Posicionamento em CSS

**Position:static**



# Posicionamento em CSS

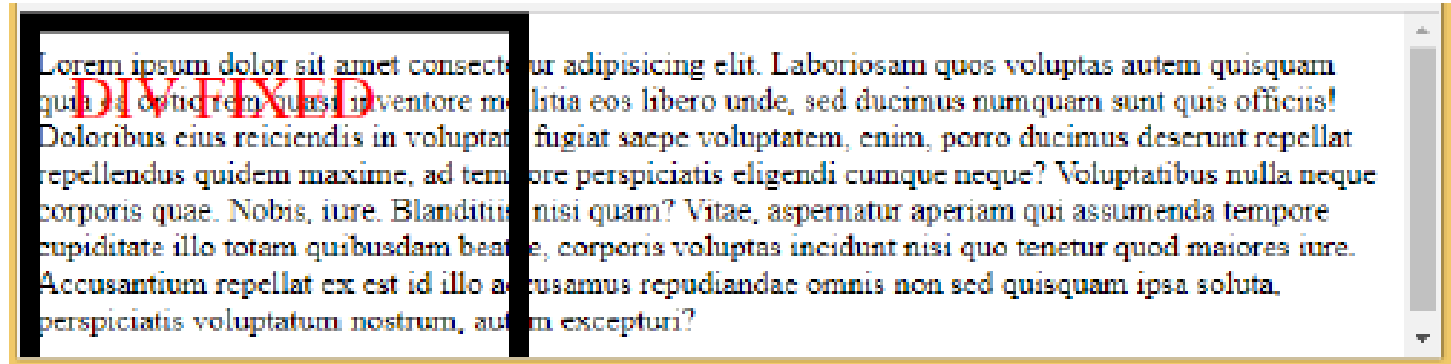
**Position:relative**





# Posicionamento em CSS

## Position:fixed



# FORMULÁRIOS

# SELETORES AVANÇADOS

ELEMENTO1 ELEMENTO2  
.portfolio P{  
    text-transform: uppercase  
}

ELEMENTO1, ELEMENTO2  
h1,p{  
    color: red  
}

Seletores filho  
ELEMENTO 1 > ELEMENTO 2  
article > p{  
    border: 5px double green  
}

# SELETORES AVANÇADOS

Seletores irmãos adjacentes  
ELEMENTO1 + ELEMENTO2  
header + p{  
    font-style: italic  
}

Seletores irmãos  
ELEMENTO1 ~ ELEMENTO2  
header ~ p{  
    color: blueviolet  
}

# PSEUDO-CLASSES

<code>:active</code>	<code>:hover</code>	<code>:only-child</code>
<code>:any</code>	<code>:indeterminate</code>	<code>:only-of-type</code>
<code>:checked</code>	<code>:in-range</code>	<code>:optional</code>
<code>:default</code>	<code>:invalid</code>	<code>:out-of-range</code>
<code>:dir()</code>	<code>:lang()</code>	<code>:read-only</code>
<code>:disabled</code>	<code>:last-child</code>	<code>:read-write</code>
<code>:empty</code>	<code>:last-of-type</code>	<code>:required</code>
<code>:enabled</code>	<code>:left</code>	<code>:right</code>
<code>:first</code>	<code>:link</code>	<code>:root</code>
<code>:first-child</code>	<code>:not()</code>	<code>:scope</code>
<code>:first-of-type</code>	<code>:nth-child()</code>	<code>:target</code>
<code>:fullscreen</code>	<code>:nth-last-child()</code>	<code>:valid</code>
<code>:focus</code>	<code>:nth-last-of-type()</code>	<code>:visited</code>
<code>:focus-within</code>	<code>:nth-of-type()</code>	



# PSEUDO-ELEMENTOS

`::after`

`::before`

`::first-letter`

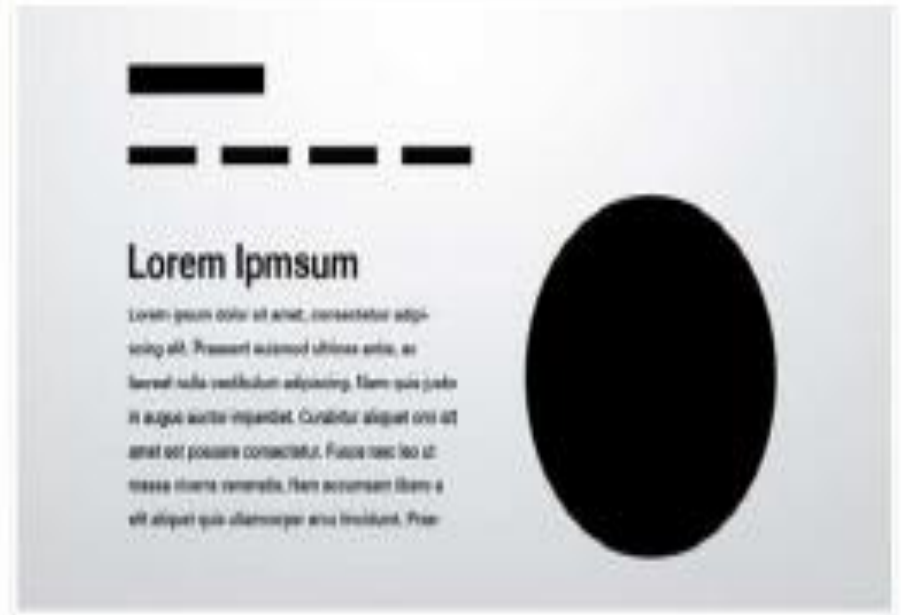
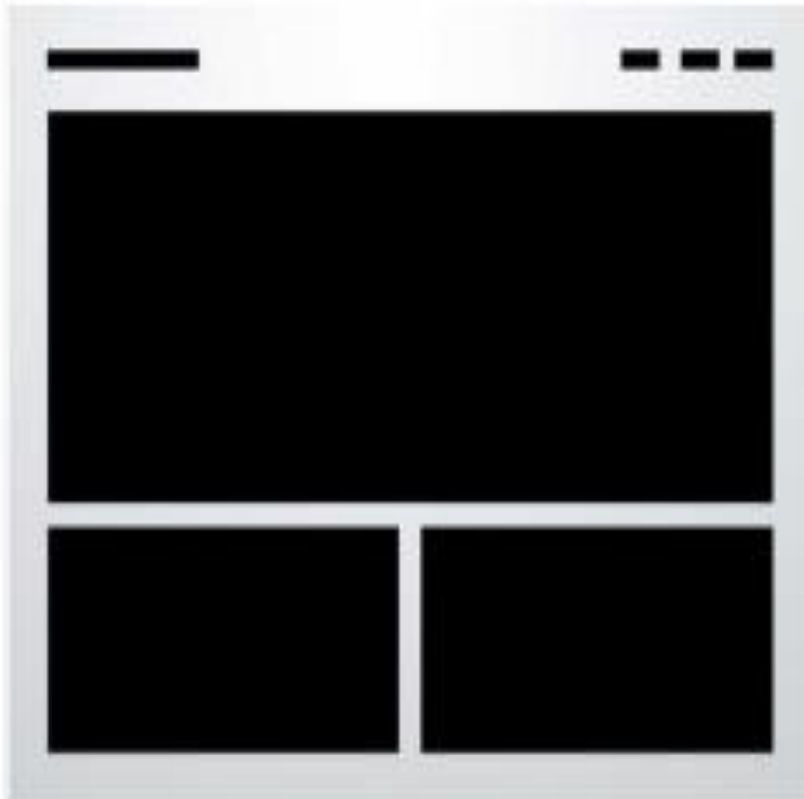
`::first-line`

`::selection`

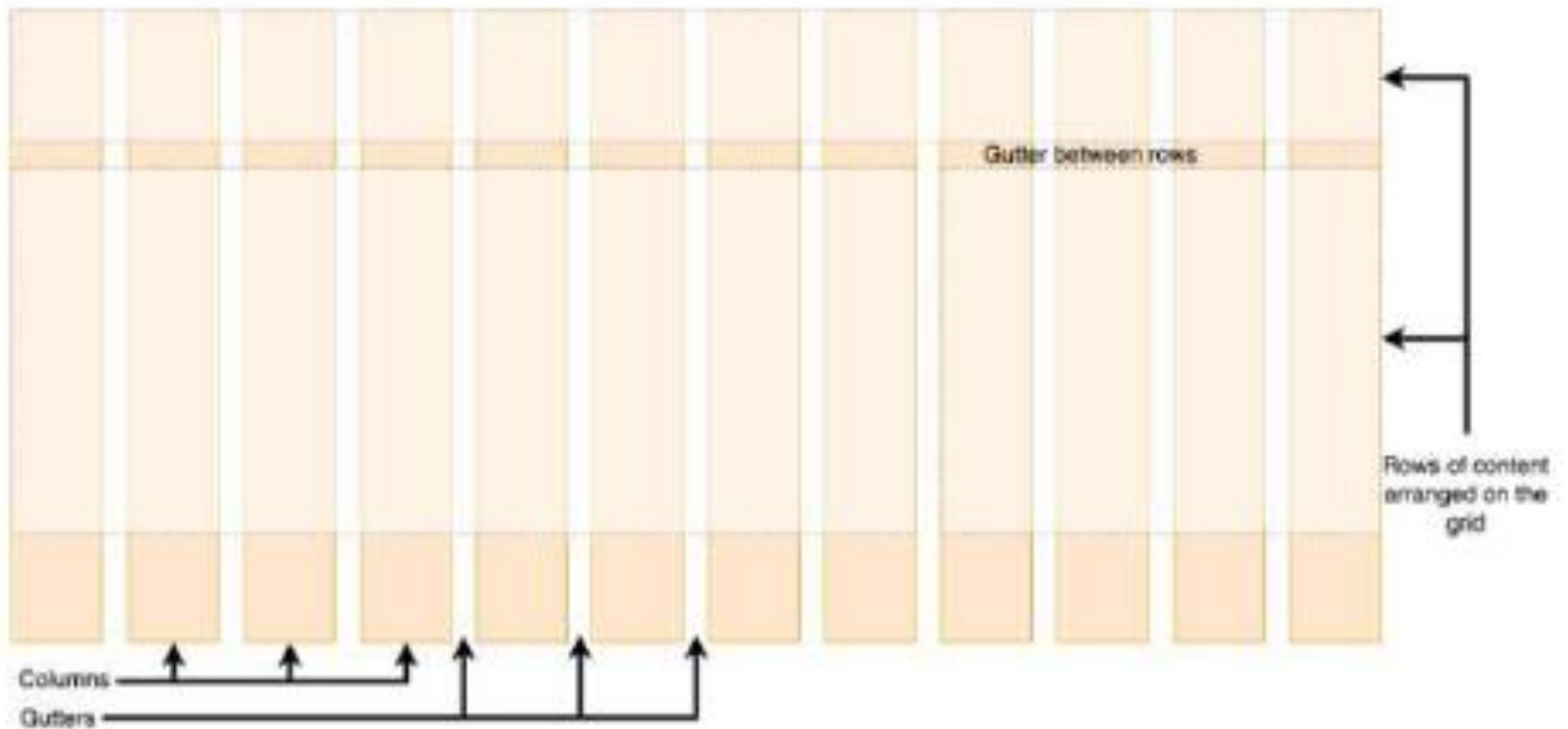
`::backdrop`

# Estudo Dirigido 2 – Seletores avançados

# Layout CSS

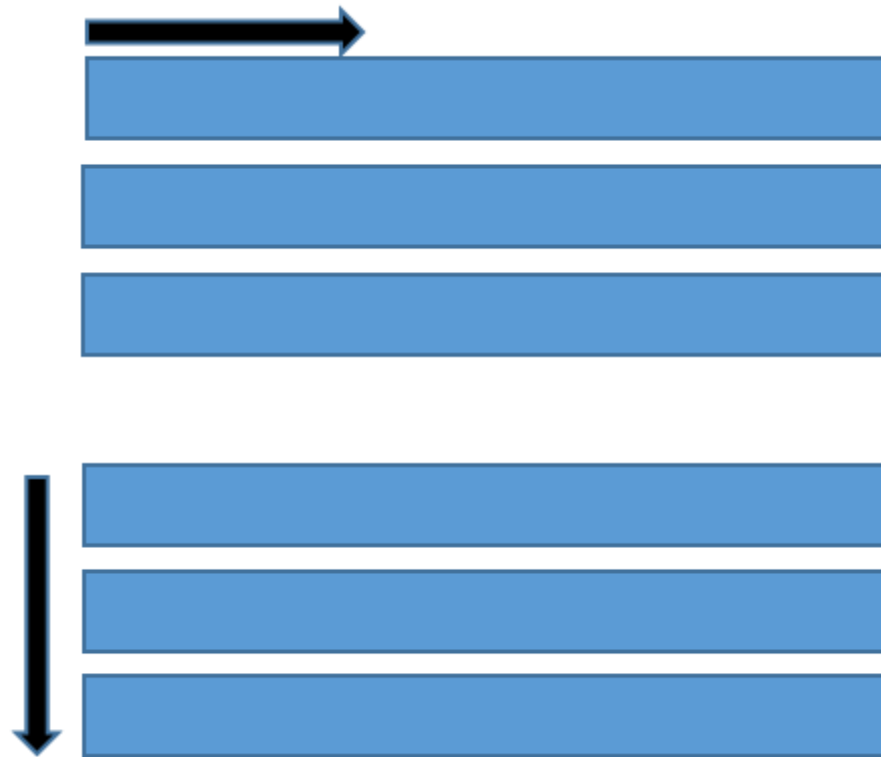


# Grid em CSS

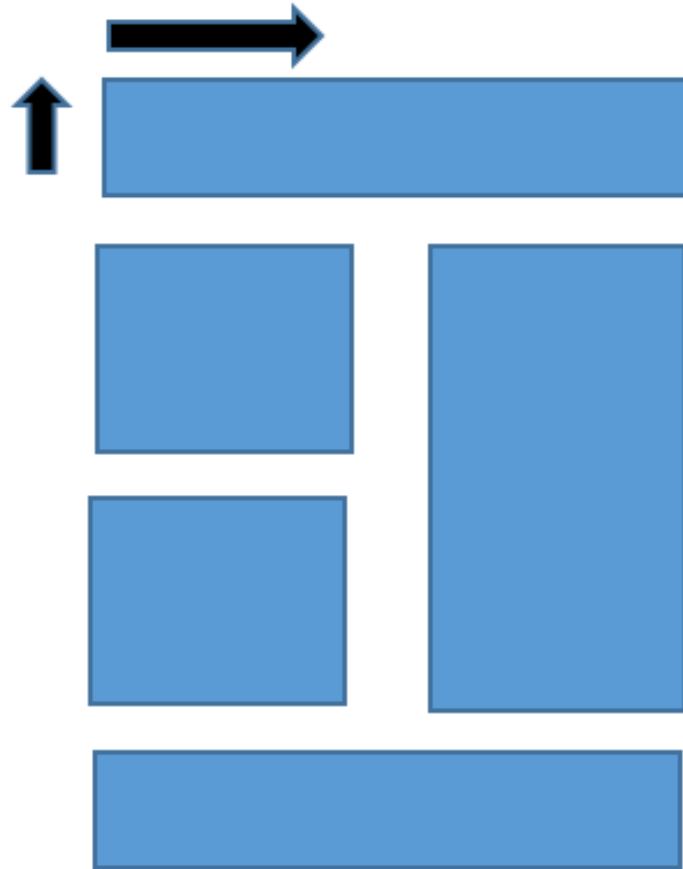


# Flexbox e Grid

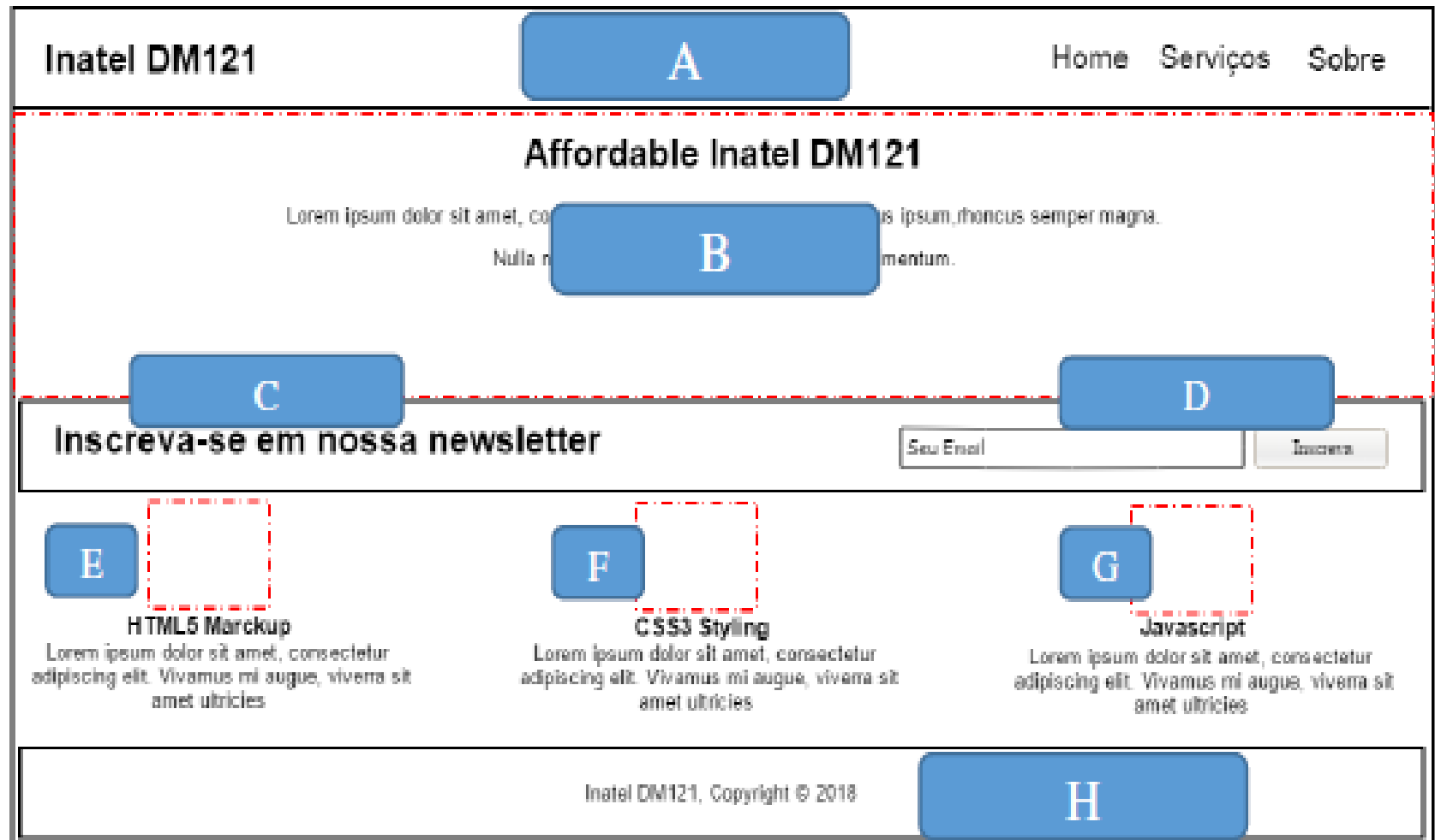
# Flexbox



# Grid



## ESTUDO DIRIGIDO 3 – APLICANDO ESTILOS COM CSS





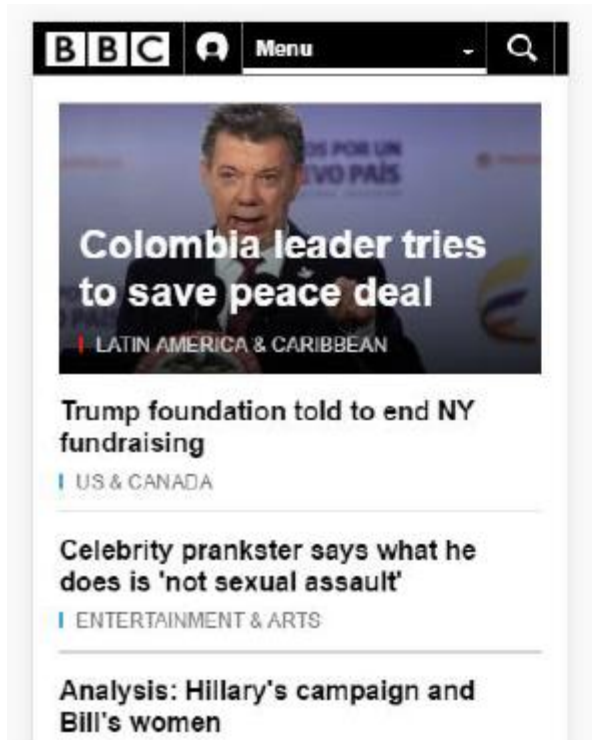
# LABORATÓRIO 1 – Adicionando a página de serviços

# LABORATÓRIO 2 – Adicionando a página sobre

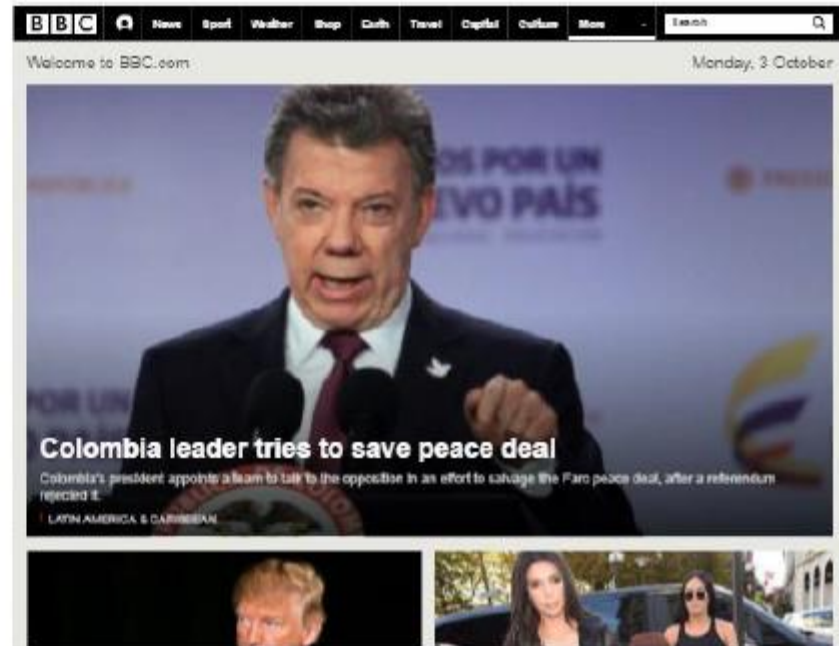
# LAYOUT RESPONSIVO



# LAYOUT RESPONSIVO



[bbc.com](http://bbc.com)



# LAYOUT RESPONSIVO

➤ **Layout Fluído**

➤ **Recursos Flexíveis**

➤ **Media Queries**

# LAYOUT RESPONSIVO

## ➤ Layout Fluído

É a criação de layout sem utilização de medidas fixas (pixels,pontos,centímetros, etc), ou seja, é utilizar medidas **flexíveis**

# LAYOUT RESPONSIVO

## ➤ Layout Fluído

### PORCENTAGEM

Especifica uma medida em relação ao elemento “pai”.

```
body{  
    width: 100% /*Ocupará a largura da página toda*/  
}  
section{  
    width: 50% /*Ocupa 50% da página em relação ao  
                elemento pai*/  
}
```

# LAYOUT RESPONSIVO

## ➤ Layout Fluído

### EM

Especifica uma medida em relação ao font-size do elemento pai. Por padrão, caso não exista nenhuma regra CSS o font-size adotado pelos navegadores é de 16px.

```
p{  
    font-size: 2em /*Parágrafos serão 16px * 2 = 32px */  
}
```



# LAYOUT RESPONSIVO

## ➤ Layout Fluído

### REM

Pode ser utilizado em qualquer propriedade, porém, temos que ter em mente que o mesmo faz relação com o font-size do elemento root.

```
html{  
    font-size: 16px;  
}  
body{  
    font-size: 20px;  
}  
p{  
    font-size: 2rem /*Parágrafos serão 16px * 2 = 32px */  
}
```

# LAYOUT RESPONSIVO

## ➤ Layout Fluído

```
<body>  
  <section class="main-content">  
    <p>HTML</p>  
    <p>CSS</p>  
    <p>Javascript</p>  
  </section>  
</body>
```

## CSS

section {font-size: 1.125em} p {font-size: 0.7em}

A tag **html** e **body** ainda terão 16px de font-size como referência (valor default do navegador).

A tag section terá:  $16\text{px} * 1.125 = 18\text{px}$

A tag p terá:  $18\text{px} * 0.7 = 12.6\text{px}$

# LAYOUT RESPONSIVO

## ➤ Layout Fluído

**Resultado = Elemento alvo / Contexto**

# LAYOUT RESPONSIVO

## ➤ Layout Fluído

Qual a equivalência em **em** de **18px** em um contexto que está configurado com **16px**?

Resultado = Elemento alvo / Contexto

Resultado = 18px / 16px

Resultado = **1.125em**

```
html{  
    font-size: 16px  
}  
body{  
    width: 980px;font-size: 1.125em  
}
```

# LAYOUT RESPONSIVO

## ➤ Recursos Flexíveis

A solução em CSS `img{max-width: 100%}`

# LAYOUT RESPONSIVO

## ➤ **Media Queries**

São consultas de mídias que com base em expressões permitem aplicar um conjunto de regras CSS específicas

# LAYOUT RESPONSIVO

## ➤ Media Types

*Relationship between media groups and media types*

Media Types	Media Groups			
	continuous/paged	visual/audio/speech/tactile	grid/bitmap	interactive/static
<b>braille</b>	continuous	tactile	grid	both
<b>embossed</b>	paged	tactile	grid	static
<b>handheld</b>	both	visual, audio, speech	both	both
<b>print</b>	paged	visual	bitmap	static
<b>projection</b>	paged	visual	bitmap	interactive
<b>screen</b>	continuous	visual, audio	bitmap	both
<b>speech</b>	continuous	speech	N/A	both
<b>tty</b>	continuous	visual	grid	both
<b>tv</b>	both	visual, audio	bitmap	both

# LAYOUT RESPONSIVO

## ➤ Media Types

Exemplo de Media Types sendo aplicadas dentro do arquivo CSS:

```
@media screen, print{  
  body{  
    width: 100%;  
  }  
}
```

Neste caso, o estilo CSS seria aplicado apenas se o tipo de dispositivo fosse screen ou se a página for impressa



# LAYOUT RESPONSIVO

## ➤ Media Queries

```
@media screen and(max-width: 800px) {  
body{  
    font-size: 70%;  
}  
}
```

# LAYOUT RESPONSIVO

## ➤ Media Queries

### Características

- ✓ Possuem quantidade extensa de parâmetros: width, height, device-width, device-height, orientation, aspect-ratio, device-aspect-ratio, color, color-index, monochrome, resolution, scan, grid
- ✓ Permitem utilização de unidades de medidas padrão do CSS
- ✓ Permitem operadores lógicos: and, or, not, only

# LAYOUT RESPONSIVO

## ➤ Media Queries

Quais pontos de interrupção usar?

- ✓ Baseado no mínimo (min-width) ou máximo (max-width)
- ✓ Baseado no conteúdo

# Estudo Dirigido 4 – Aplicando técnicas de layout responsivo

# Laboratório 3 – Aplicando Media queries página serviços

# Laboratório 4 – Aplicando Media queries página sobre

# Javascript

Criado por Brendan Eich da Netscape o Javascript é uma linguagem leve, interpretada e com recursos de orientação a objetos e funções de primeira classe. Inicialmente o contexto de execução do javascript era o navegador web onde a principal finalidade era adicionar comportamento em páginas web.

# Javascript

Javascript padrão ECMA-262

Javascript standard ECMAScript 2011 (ES5.1)

Ciclo de lançamento

- ECMAScript 2015 (ES6)
- ECMAScript 2016 (ES7)
- ECMAScript 2017 (ES8).



# Javascript

## Palavras chaves

break  
case  
catch  
continue  
debugger  
default  
delete

do  
else  
finally  
for  
function  
if  
in

instanceof  
new  
return  
switch  
this  
throw  
try

typeof  
var  
void  
while  
with

# Javascript

## Operadores

{	}	(	)	[	]
.	,	,	<	>	<=
>=	==	!=	===	!==	
+	-	*	%	++	--
<<	>>	>>>	&		^
!	~	&&		?	:
=	+=	-=	*=	/=	<<=
>>=	>>>=	&=	=	^=	

# Javascript

## **Declaração de variáveis**

# Javascript

## **Manipulação de string**

# Javascript

**If, else, do,  
while, for**

# Javascript

## **Array**

# Javascript

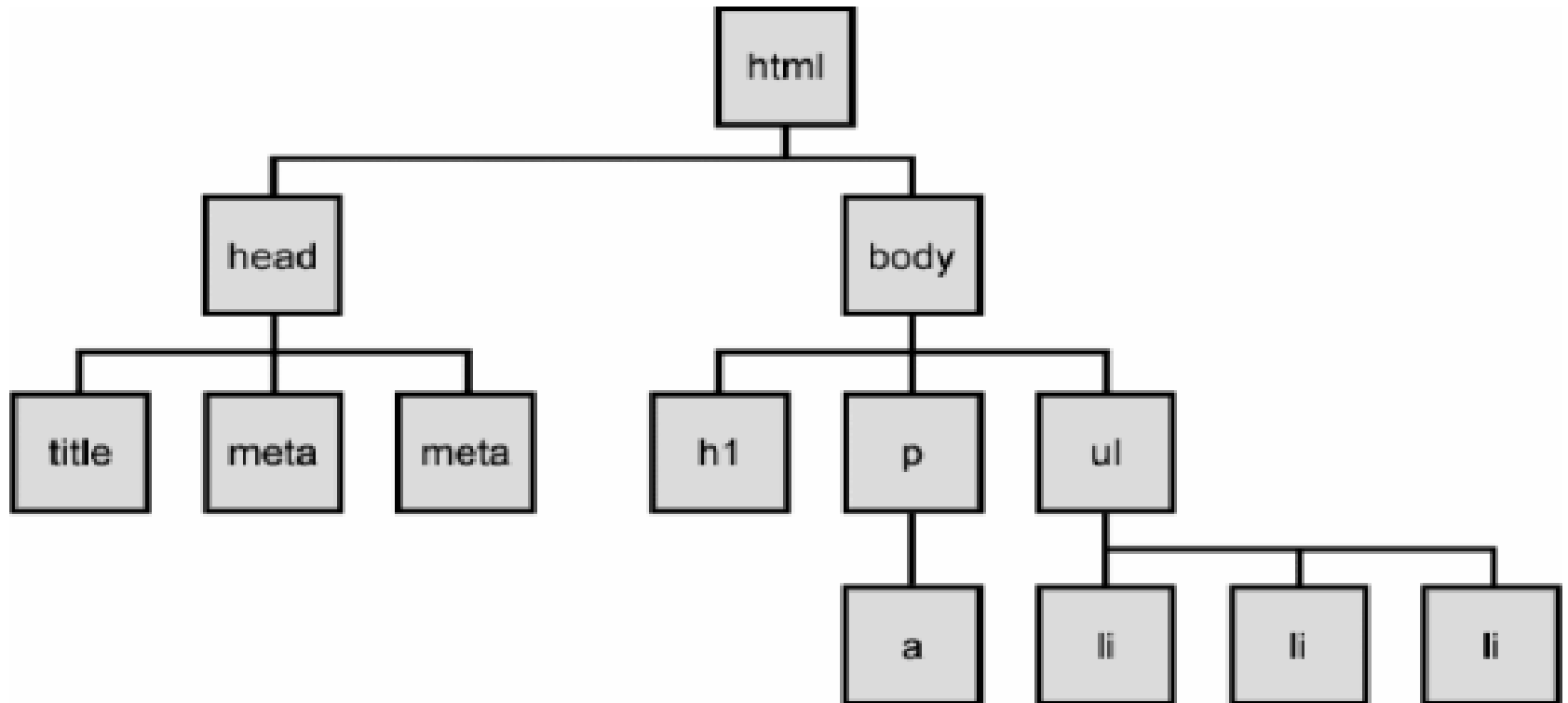
## **Escopo de variáveis**

# Javascript

## **Closure Javascript**



# DOM API



# DOM API

**Objetos document e window**

# Javascript

**ES6, ES7, ES8 e além**

# ESTUDO DIRIGIDO 5 – Adicionando Javascript página serviços

# LABORATÓRIO 5 – Aplicando Javascript na página sobre