



**Universidade Federal de Santa Catarina**  
**Curso de Tecnologias da Informação e Comunicação**  
**Desenvolvimento de Sistemas Web**  
**Professor: Adriano de Oliveira**

---

# **Desenvolvimento de Sistemas Web**

**Prof.: Adriano de Oliveira**  
**email: [adriano.inovar@gmail.com](mailto:adriano.inovar@gmail.com)**  
**Turma: 05652 - 3-2020-2 e 5-1830-2**

# CSS

## CSS - CASCADING STYLE SHEETS

- ✓ **CSS** é o abreviação de *Cascading Style Sheets*.
- ✓ **CSS** é uma linguagem que permite definir o *estilo visual* de documentos web, definindo como deve ser apresentado o conteúdo e estrutura definidos em **HTML**.

## CSS – POR QUE USAR?

- ✓ A separação ao nível das tecnologias facilita a separação e organização do trabalho nas equipes.
- ✓ A separação entre **HTML** e **CSS** permite definir o conteúdo e estrutura apenas uma vez e ter diferentes folhas de estilos em função do dispositivo de destino.
- ✓ Com **CSS** é possível definir a apresentação em função do destino.



# VISÃO GERAL



+ CSS =



+ CSS =



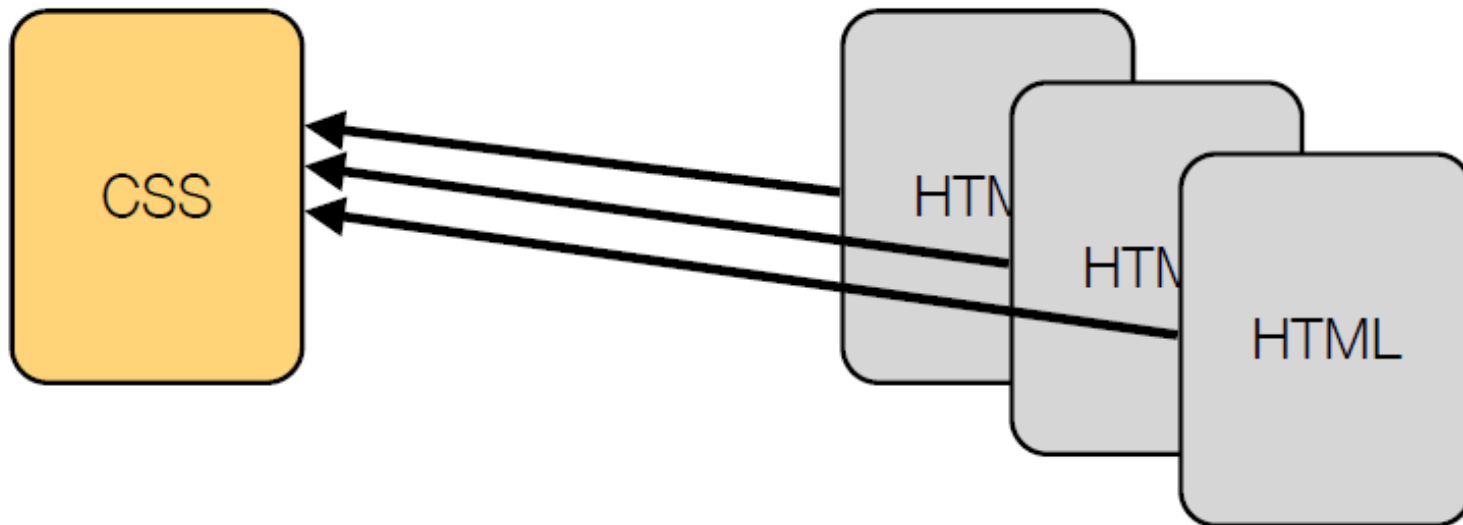
+ CSS =



## USO DO CSS

- 1) Criar um novo documento de texto do tipo **CSS** (extensão **.css**).
- 2) Indicar no documento **HTML** a associação ao documento **CSS**.
- 3) Vários documentos **HTML** podem ser associados ao mesmo documento **CSS**.

## USO DO CSS



## LIGAÇÃO DO HTML COM O CSS

- ✓ O uso de regras CSS externas é feito incluindo no cabeçalho de um documento **HTML** uma ligação para o arquivo com as regras **CSS**.

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



## REGRAS CSS

- ✓ Um documento *CSS* é composto por um conjunto de regras *CSS*.

seletor



h1 {

valores



color: rgb(100,100,100);

margin-top: 10px;

font-family: Arial, sans-serif;

}



propriedades

## SELETORES CSS

- ✓ Um seletor identifica elementos num documento *HTML*.

```
h1 {  
    /* Identifica os elementos h1 */  
}  
  
p {  
    /* Identifica os elementos p */  
}  
  
div {  
    /* Identifica os elementos div */  
}
```

## SELETORES CSS

- ✓ Um seletor identifica elementos num documento *HTML*.

```
h1 {  
  background-color: ■red;  
  color: ■black;  
  font-family: Arial, sans-serif;  
}  
  
p {  
  margin-top: 10px;  
  text-align: justify;  
}  
  
div {  
  width: 80%;  
}
```

## PROPRIEDADES E VALORES

- ✓ Cada elemento *HTML* tem um conjunto de propriedades.
- ✓ Cada propriedades aceita um conjunto de valores.
- ✓ As propriedades que não são alteradas, mantêm o valor original.



## PROPRIEDADES E VALORES

```
h1 {  
    color: ■ red;  
    font-style: italic;  
}  
  
p {  
    letter-spacing: 5px;  
    font-size: 18px;  
}
```

## PROPAGAÇÃO DE ESTILOS

- ✓ Algumas propriedades **CSS** são propagadas para os elementos filhos de um elemento **HTML**.

```
<body>
  <h1>Título</h1>
  <p>Parágrafo</p>
</body>
```

HTML

```
body {
  font-family: Verdana, Arial;
  font-size: 14px;
}
```

CSS

# **PROPRIEDADES BÁSICAS**

## FORMATAÇÃO DE TEXTO

- ✓ ***font-family:*** Os tipos de letra usado.
- ✓ ***font-size:*** Tamanho das letras. Unidades possíveis: px, %.
- ✓ ***font-style:*** Estilo das letras. Opções: normal ou italic.
- ✓ ***font-weight:*** Espessura das letras. Opções: lighter, normal, bold ou bolder.
- ✓ ***text-decoration:*** Efeitos que será aplicado ao texto. Opções: none, underline, overline, line-through.



## TIPOS DE LETRAS

- ✓ Para usar um determinado tipo de *letra* num documento web, é necessário que esse tipo esteja instalado no computador do utilizador que acessa o documento.
- Os tipos de letra mais comuns nos computadores são: *Arial*, *Courier New*, *Times New Roman*, *Verdana*, etc.

## TIPOS DE LETRAS

- ✓ A propriedade *font-family* permite indicar uma lista de tipos de letra para ser usada na formatação do elemento.
- ✓ O navegador web tenta usar o primeiro tipo de letra indicado. Se o tipo não estiver instalado no computador é tentado o tipo seguinte, e assim sucessivamente.

*font-family: Verdana, Arial, Courier New;*

## EXEMPLO DE FORMATAÇÃO DE TEXTO

- ✓ Um parágrafo com alterações na formatação.

```
p {  
  font-family: Verdana;  
  font-size: 18px;  
  font-style: italic;  
  font-weight: bold;  
  text-decoration: underline;  
}
```



## APRESENTAÇÃO DE 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.



## EXEMPLO DE APRESENTAÇÃO DE TEXTO

- ✓ Um parágrafo com alterações ao nível de apresentação.

```
p {  
  text-align: justify;  
  text-indent: 3em;  
  line-height: 2em;  
  letter-spacing: 0.50em;  
  word-spacing: 2.50em;  
}
```



127.0.0.1:5500/aula

T e s t e            d e            p a r á g r a f o

## PRINCIPAIS UNIDADES



- ✓ As principais unidades são as duas seguintes:
  - px (píxel) — Os píxeis representam unidades absolutas, um pixel corresponde a um pixel na tela. 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%.

## DEFINIÇÃO DE CORES

- ✓ Existem várias alternativas para definir cores em CSS:
  - 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`.*

## DEFINIÇÃO DE CORES

- ✓ ***color***: Cor do texto do elemento.
- ✓ ***background-color***: Cor do fundo do elemento.

```
p {  
  color:  rgb(100%, 0%, 0%);  
  background-color:  yellow;  
}
```

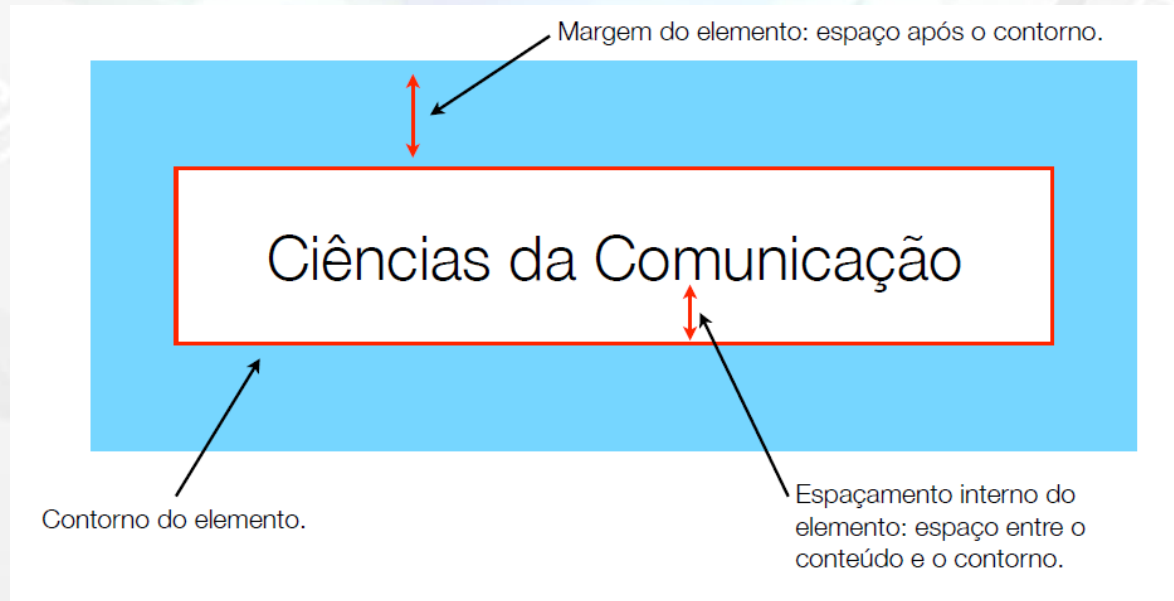


Um texto colorido



## MARGENS E ESPAÇAMENTO

- ✓ Cada elemento HTML é representado visualmente como uma caixa retangular composta por três partes: a margem, o contorno e o espaçamento interno.



## CONTORNO

- ✓ • É possível definir o contorno de todo o elemento ou apenas de parte:
  - ***border:*** alteração de todo o contorno.
  - ***border-top:*** alteração apenas do contorno de topo.
  - ***border-bottom:*** contorno inferior.
  - ***border-left:*** contorno esquerdo.
  - ***border-right:*** contorno direito

## CONTORNO

- ✓ O contorno define-se especificando três valores: a espessura, o estilo e a cor.
  - ***Espessura:*** definida em píxels.
  - ***Estilo:*** none, dotted, dashed, solid, double, groove, ridge.
  - ***Cor:*** valor da cor.

# CONTORNO

```
h1 {  
  border: 1px solid black;  
}  
  
h2 {  
  border-top: 2px dotted red; rgb(255, 0, 0);  
  border-bottom: 2px dashed blueviolet;  
}
```

← → ↻ 🏠 🛡️ 📄 127.0.0.1:5

**Titulo h1**

**Titulo h2**



## MARGEM

- ✓ É possível definir a margem de todo o elemento ou apenas de parte. Na definição da margem indica-se a distância em pixels.
  - ***margin***: alteração da margem de todo o elemento.
  - ***margin-top***: alteração apenas da margem de topo.
  - ***margin-bottom***: margem inferior.
  - ***margin-left***: margem esquerda.
  - ***margin-right***: margem direita.



## ESPAÇAMENTO INTERNO

- ✓ É possível definir o espaçamento interno de todo o elemento ou apenas de parte. Na definição do espaçamento indica-se a distância em píxels.
  - ***padding***: alteração do espaçamento interno de todo o elemento.
  - ***padding-top***: alteração apenas do espaçamento interno de topo.
  - ***padding-bottom***: espaçamento interno inferior.
  - ***padding-left***: espaçamento interno esquerdo.
  - ***padding-right***: espaçamento interno direito.

# MARGEM E ESPAÇAMENTO

```
img {  
    margin: 15px;  
}  
  
a {  
    margin-left: 5px;  
    margin-right: 5px;  
}
```

Margens

```
h1 {  
    padding: 10px;  
}  
  
p {  
    padding-top: 10px;  
    padding-bottom: 15px;  
}
```

Espaçamento interno

## LISTAS

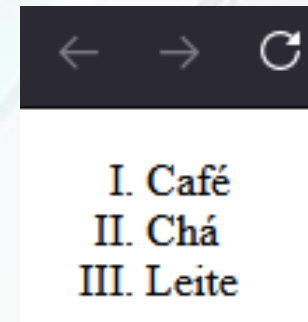
- ✓ É possível definir a aparência do marcador de cada item com a propriedade ***list-style-type***.
  - ***Marcadores não numéricos:*** none, disc, circle, square.
  - ***Marcadores numéricos:*** none, decimal, decimal-leading-zero, lowerroman, upperroman, lower-latin, upper-latin.

# LISTAS

```
ul {  
  list-style-type: square;  
}
```



```
ul {  
  list-style-type: upper-roman;  
}
```





## ATRIBUTO CLASS

- ✓ É possível distinguir os elementos usando classes. Todos os elementos **HTML** têm um atributo optativo class.
  - Os nomes das classes podem ser compostos por letras (não acentuadas) ou algarismos, e devem começar com uma letra.
  - **Exemplos:** principal, menu, submenu, rodape, cabeçalho, publicidade.
  - Cada elemento pode pertencer a várias classes. Para atribuir um elemento a várias classes, basta separar os nomes das diferentes classes com espaços.



## ATRIBUTO CLASS

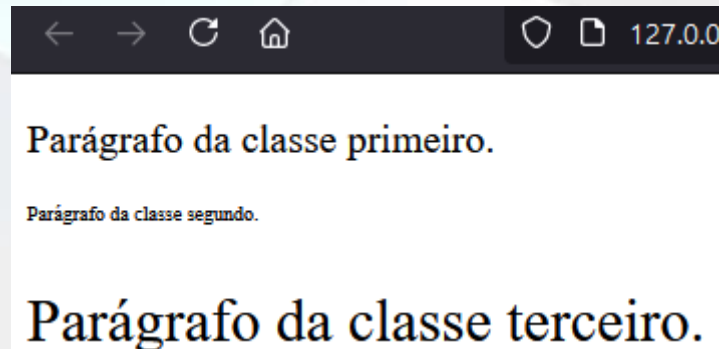
- ✓ Para identificar uma determinada classe em **CSS** usa-se um ponto seguido do *nome da classe*. Exemplos: .menu, .rodape, .destaque.

```
p {  
  font-size: 10px;  
}
```

```
p.primeiro {  
  font-size: 20px;  
}
```

```
p.terceiro {  
  font-size: 30px;  
}
```

```
<p class="primeiro">Parágrafo da classe primeiro.</p>  
<p class="segundo">Parágrafo da classe segundo.</p>  
<p class="terceiro">Parágrafo da classe terceiro.</p>
```



## SELEÇÃO CONJUNTA DE ELEMENTOS

- ✓ É possível aplicar a mesma regra **CSS** a elementos diferentes, para tal basta separar os vários elementos por vírgulas.
- ✓ A principal vantagem desta opção é evitar a duplicação de código idêntico.

# SELEÇÃO CONJUNTA DE ELEMENTOS

```
h1, h2, h3 {  
  /* Regra comum aos elementos h1, h2, e h3 */  
  background-color: brown;  
}  
  
p, h4 {  
  /* Regra comum aos elementos p e h3 */  
  background-color: goldenrod;  
}
```

```
<h1>Título 1</h1>  
<h2>Título 2</h2>  
<h3>Título 3</h3>  
<h4>Título 4</h4>  
<p>Parágrafo</p>
```

Título 1

Título 2

Título 3

Título 4

Parágrafo

## SELEÇÃO DE ELEMENTOS ENCAIXADOS

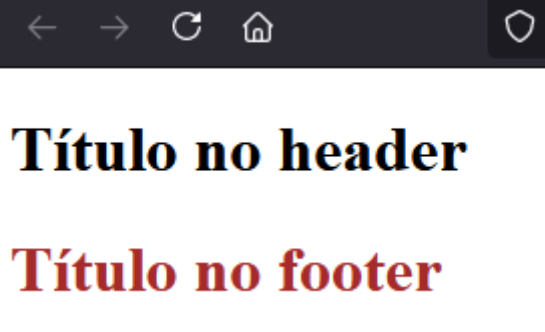
- ✓ É possível selecionar elementos em **CSS** com base na estrutura do **HTML**. Por exemplo, selecionar os elementos do tipo A que estão dentro de elementos do tipo B.
- ✓ Para selecionar elementos encaixados em outro elemento, usa-se o espaço.



# SELEÇÃO DE ELEMENTOS ENCAIXADOS

```
header h1 {  
    /* elementos h1 dentro de elementos header */  
    color: black;  
}  
  
footer h1 {  
    /* elementos h1 dentro de elementos footer */  
    color: brown;  
}
```

```
<header>  
    <h1>Título no header</h1>  
</header>  
  
<footer>  
    <h1>Título no footer</h1>  
</footer>
```





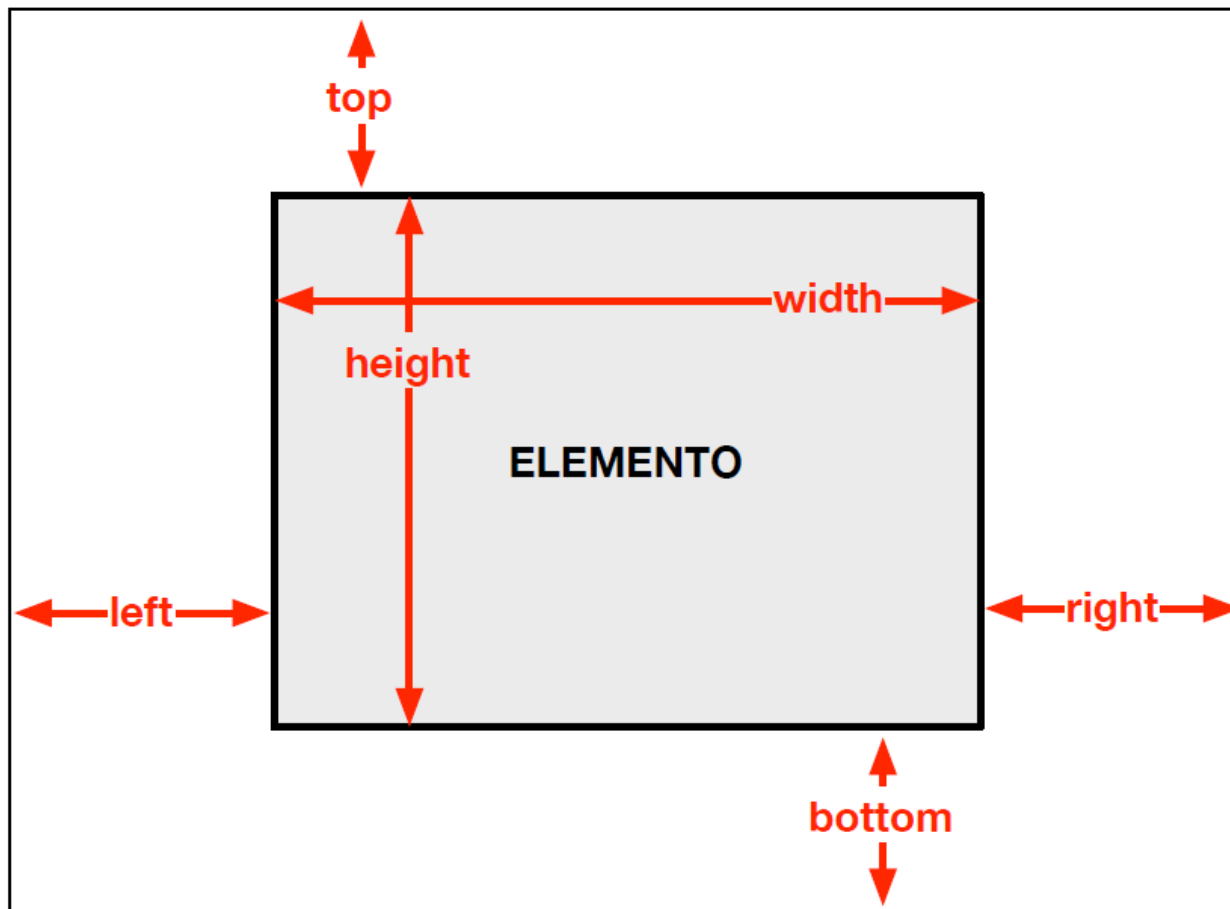
## PROPRIEDADE POSITION

- ✓ Todos os elementos têm a propriedade *position* que estabelece a forma de cálculo da *posição* de um elemento. Existem 4 alternativas para o posicionamento de um elemento:
  - ***static***: O elemento é posicionado segundo o fluxo normal. Este é o valor definido por omissão.
  - ***relative***: A posição do elemento é calculada em relação à posição original segundo o modelo estático.

## PROPRIEDADE POSITION

- ✓ Todos os elementos têm a propriedade *position* que estabelece a forma de cálculo da *posição* de um elemento. Existem 4 alternativas para o posicionamento de um elemento:
  - ***absolute***: A posição do elemento é especificada com as propriedades top, right, bottom e left relativamente ao elemento que o contém.
  - ***fixed***: O cálculo da posição é igual ao método absoluto mas o elemento pai é a "janela de visualização" (o navegador).

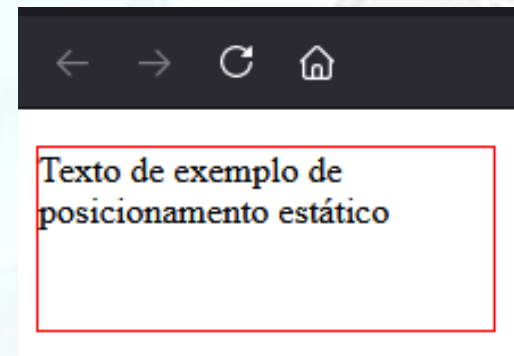
## PROPRIEDADE POSITION



## POSICIONAMENTO ESTÁTICO

- ✓ O posicionamento estático corresponde ao posicionamento normal, definido por omissão. Segue a sequência definida no documento *HTML*, de cima para baixo e da esquerda para a direita
- ✓ As propriedades *top*, *left*, *bottom*, *right* não se aplicam.

```
p {  
  position: static;  
  width: 200px;  
  height: 5em;  
  border: 1px solid red;  
}
```

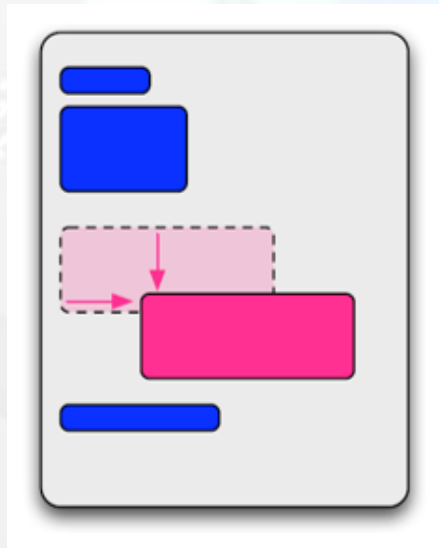


```
<p>Texto de exemplo de posicionamento estático</p>
```



## POSICIONAMENTO RELATIVO

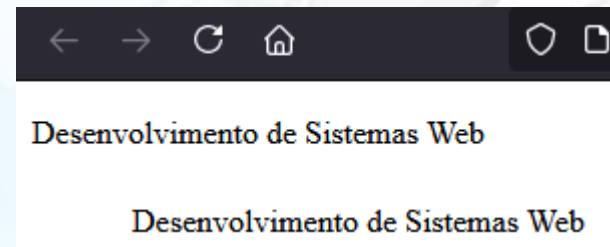
- ✓ Com o *posicionamento* relativo, a *posição* é calculada em relação à localização natural do elemento. O elemento pode ser deslocado na vertical ou na horizontal face à posição original.



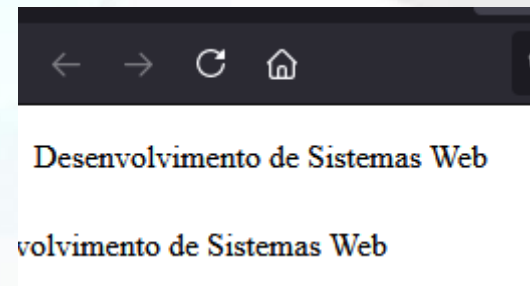
```
p {  
  position: relative;  
  top: 50px;  
  left: 50px;  
}
```

## EXEMPLO DE POSICIONAMENTO RELATIVO

```
p.paragrafo2 {  
  position: relative;  
  top: 10px;  
  left: 50px;  
}
```

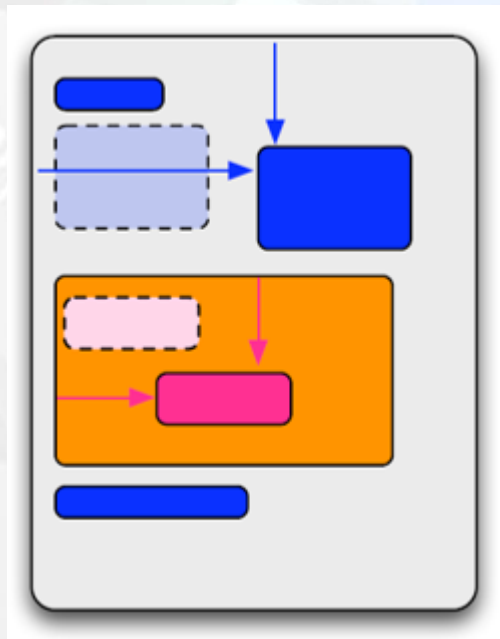


```
p.paragrafo2 {  
  position: relative;  
  top: 10px;  
  right: 50px;  
}
```



## POSICIONAMENTO ABSOLUTO

- ✓ Com o *posicionamento absoluto*, a posição do elemento é calculada em relação ao elemento 'pai' (o elemento que engloba este).



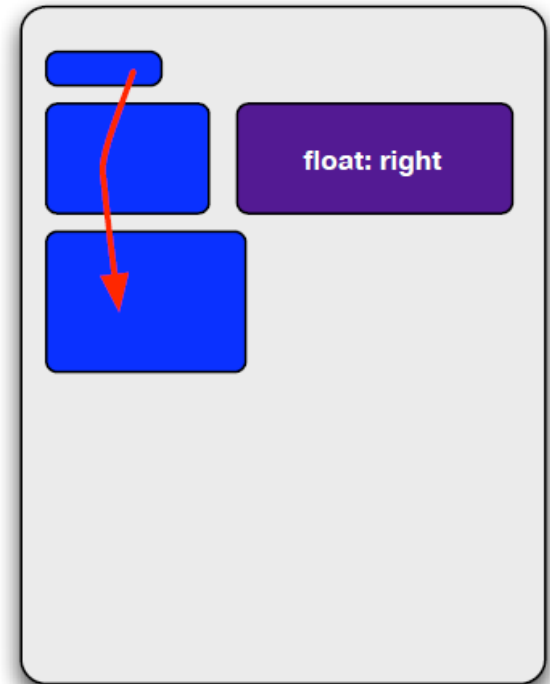
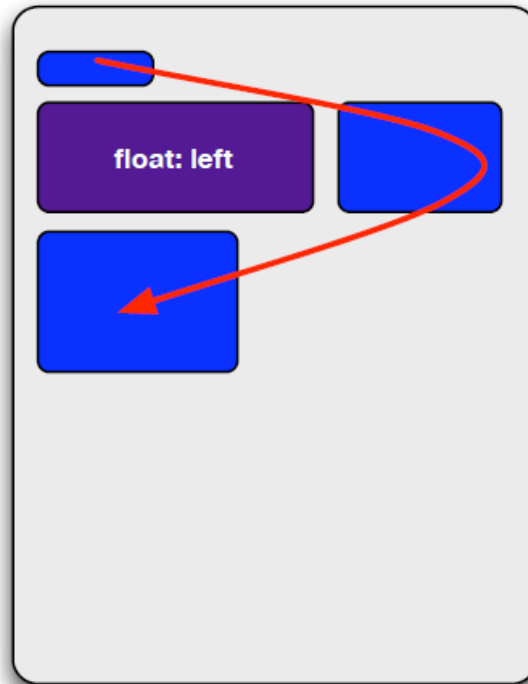
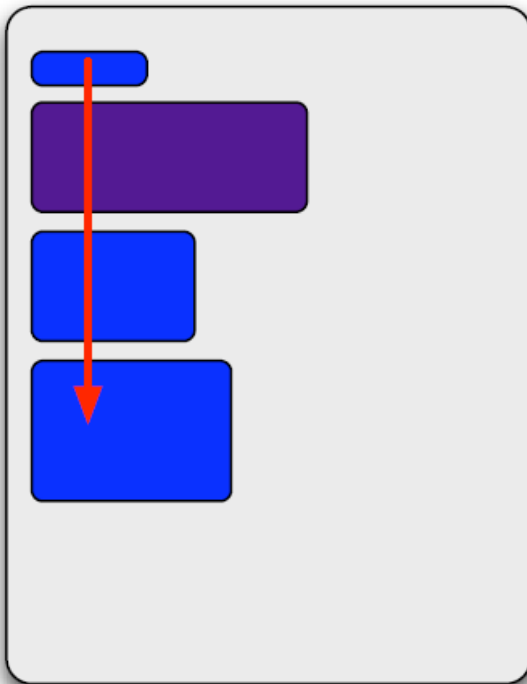
```
p {  
    position: absolute;  
    top: 50px;  
    left: 50px;  
}
```

## PROPRIEDADE FLOAT

- ✓ A propriedade *float* permite desassociar o elemento do fluxo normal e deslocá-lo para a esquerda ou direita da linha, permitindo que o restante conteúdo circule paralelamente ao elemento.
- ✓ Valores possíveis:
  - *none*: Valor por omissão.
  - *left*: O elemento é deslocado para a esquerda com o conteúdo envolvente a flutuar à direita.
  - *right*: O elemento é deslocado para a direita com o conteúdo envolvente a flutuar à esquerda.



## EXEMPLO DA PROPRIEDADE FLOAT





**Universidade Federal de Santa Catarina**  
**Curso de Tecnologias da Informação e Comunicação**  
**Desenvolvimento de Sistemas Web**  
**Professor: Adriano de Oliveira**

---

**FONTE: W3schools**

[CSS Tutorial \(w3schools.com\)](http://w3schools.com)