

**GEX613 – Programação II**

**CSS**



1100/1101 – CIÊNCIA DA COMPUTAÇÃO

Giancarlo Salton & Edimar Junior

O que é CSS?

Cascading Style Sheets

Modelo de Caixa

Estilizando a Página HTML

Express Server

# O que é CSS?

- **CSS** (Cascading Style Sheets) é uma linguagem declarativa que controla a apresentação visual de páginas web em um navegador.
- O navegador aplica as declarações de estilo CSS aos elementos selecionados para exibí-los apropriadamente.
- Uma declaração de estilo contém as propriedades e seus valores, que determinam a aparência de uma página web.

- CSS é uma das três principais tecnologias Web, junto com HTML e JavaScript.

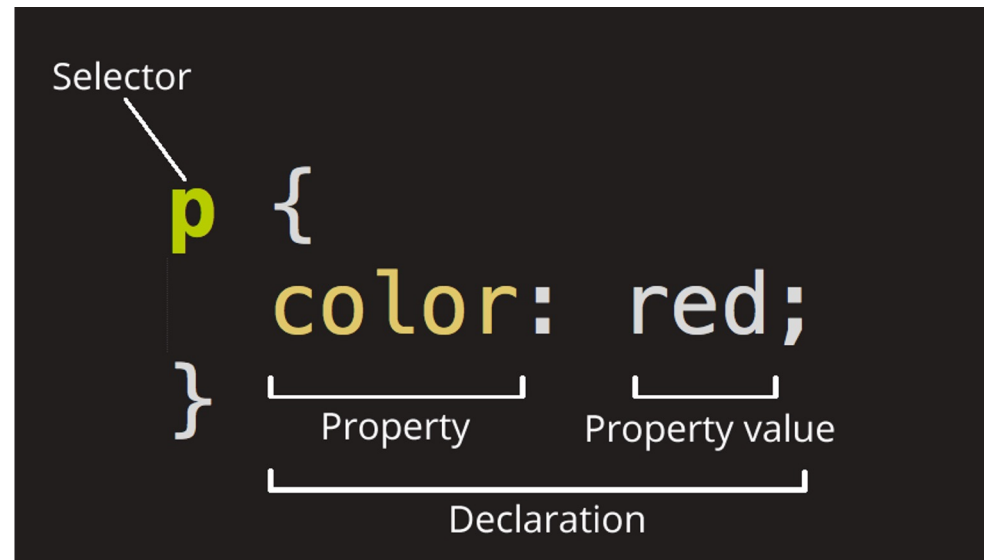
CSS normalmente estiliza Elementos HTML

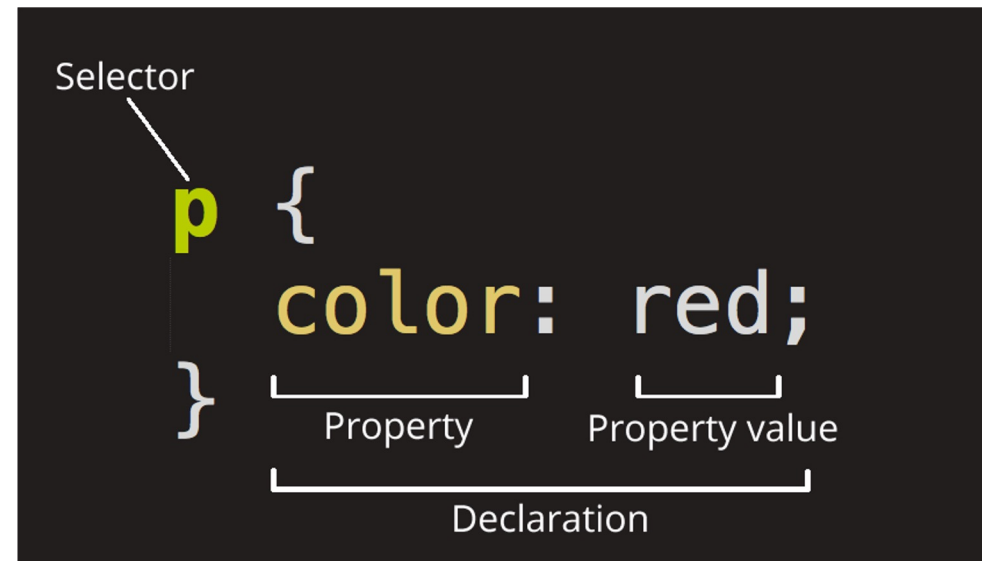
- Uma regra CSS é um conjunto de propriedades associados a um seletor.

Aqui está um exemplo que faz com que todos os parágrafos HTML fiquem amarelos num fundo preto:

```
p { /* O seletor "p" indica que todos os paragrafos no documento serão afetados por essa regra */  
    /* A propriedade "color" define a cor do texto, neste caso amarelo. */  
    color: yellow;  
  
    /* A propriedade "background-color" define a cor ao fundo, neste caso preto. */  
    background-color: black;  
}
```

# Cascading Style Sheets

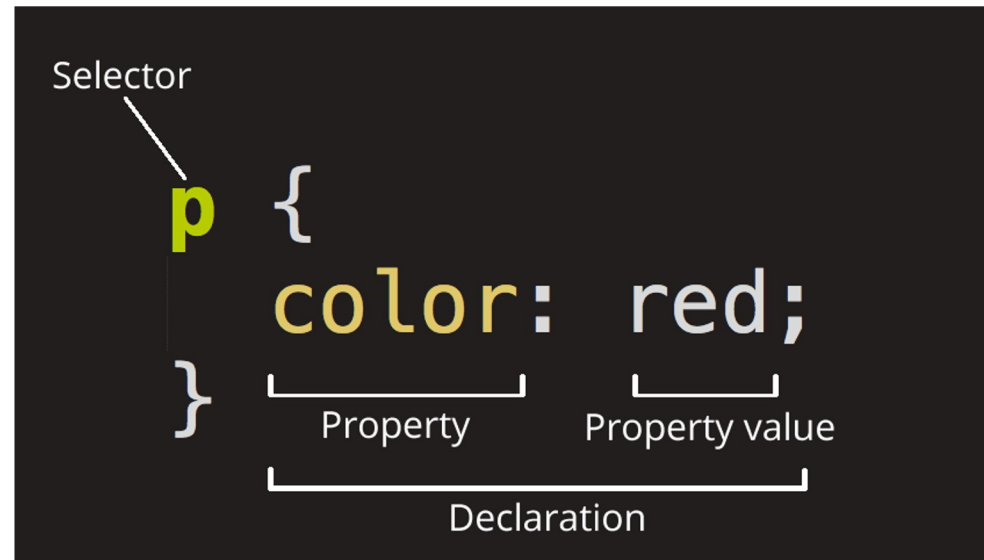




### Seletor (Selector)

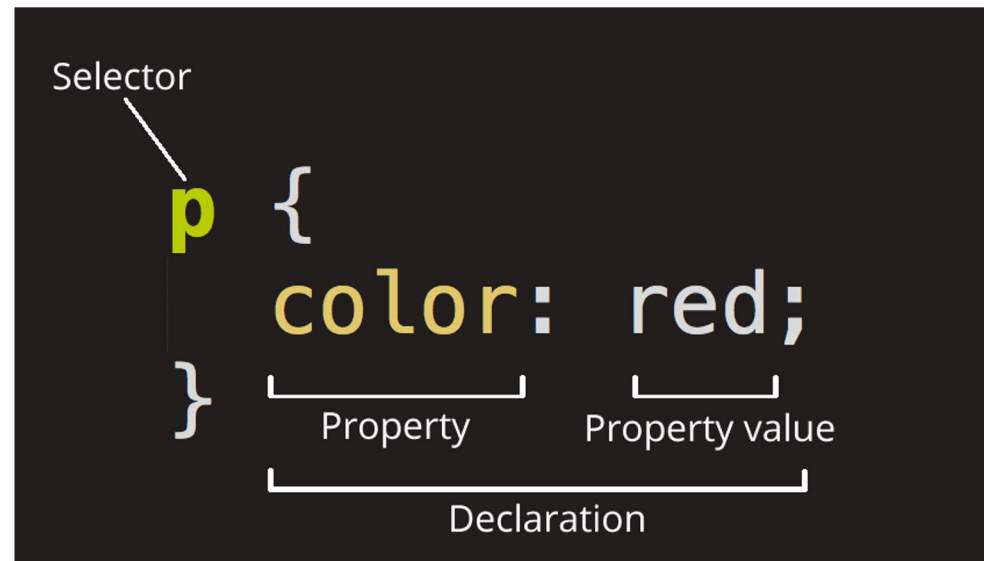
O nome do elemento HTML no começo do conjunto de regras. Ele seleciona o(s) elemento(s) a serem estilizados (nesse caso, elementos `<p>`). Para dar estilo a um outro elemento, é só mudar o seletor.





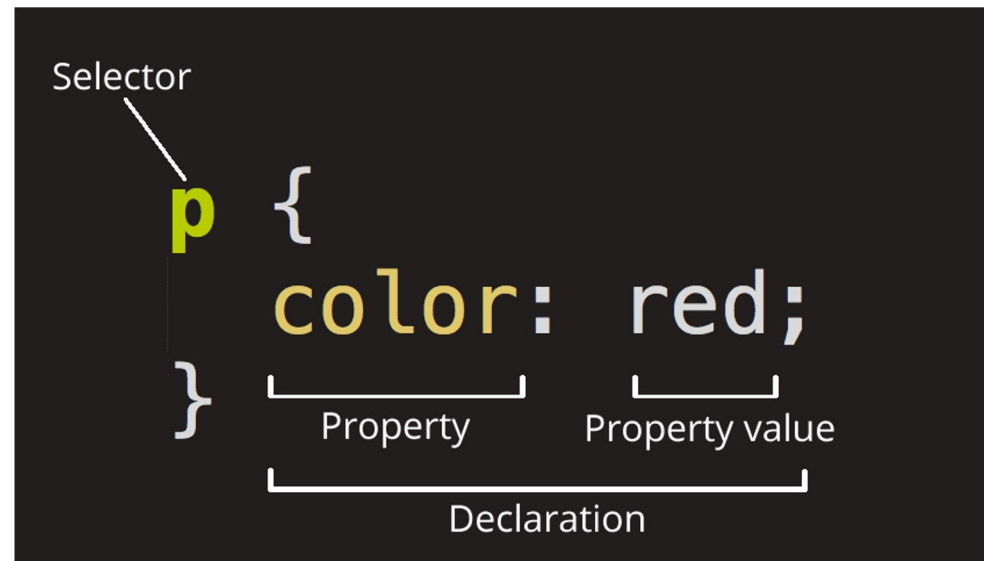
### Declaração (Declaration)

Uma regra simples como `color: red;` especificando quais das **propriedades** do elemento você quer estilizar.



### Propriedades (Property)

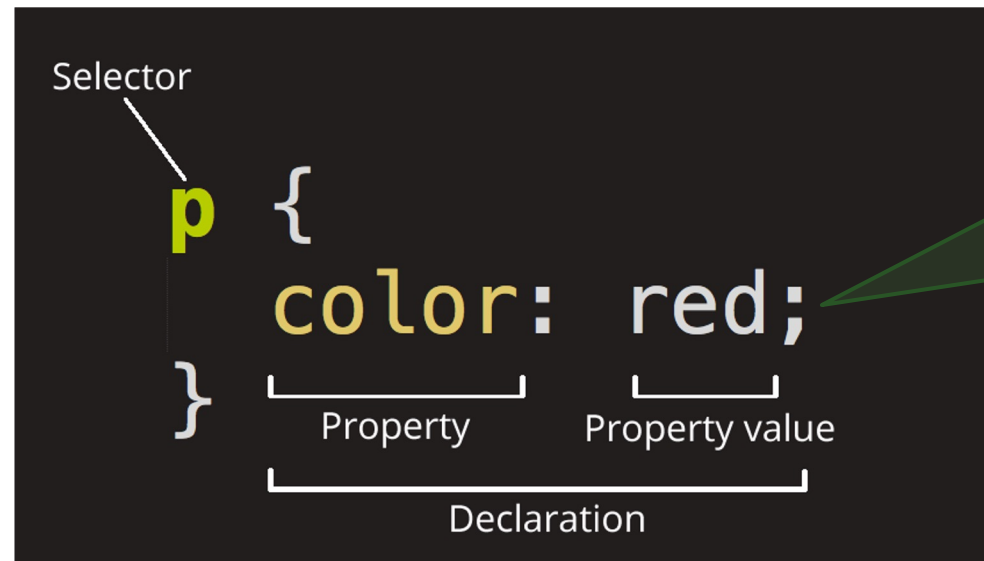
Forma pela qual você estiliza um elemento HTML. (Nesse caso, `color` é uma propriedade dos elementos `<p>`.) Em CSS, você escolhe quais propriedades você deseja afetar com sua regra.



Valor da propriedade (Property value)

À direita da propriedade, depois dos dois pontos, nós temos o **valor de propriedade**, que escolhe uma dentre muitas aparências possíveis para uma determinada propriedade (há muitos valores color(cor) além do red(vermelho)).

## Anatomia de um elemento



Dentro de cada conjunto de regras, você deve usar um ponto e vírgula (;) para separar cada declaração da próxima

## Seletor de ID

O elemento na página com o ID especificado. Em uma determinada página HTML, é uma boa prática usar um elemento por ID (e claro, um ID por elemento) mesmo que seja permitido usar o mesmo ID para vários elementos.

#my-id Selecciona

```
<p id='my-id'>
```

ou

```
<a id='my-id'>
```

## Seletor de classe

O(s) elemento(s) na página com a classe especificada (várias instâncias de classe podem aparecer em uma página).

.my-class Seleciona

```
<p class='my-class'>
```

ou

```
<a class='my-class'>
```

`img[src]` Seleciona

Seletor de atributo

O(s) elemento(s) na página com  
o atributo especificado.

```
<img src='myimage.png'>
```

mas não seleciona

```
<img />
```

Seletor de *pseudo-classe*

O(s) elemento(s)  
especificado(s), mas somente  
quando estiver no estado  
especificado. Ex.: com o mouse  
sobre ele.

`img[src]` Selecciona

`a:hover`

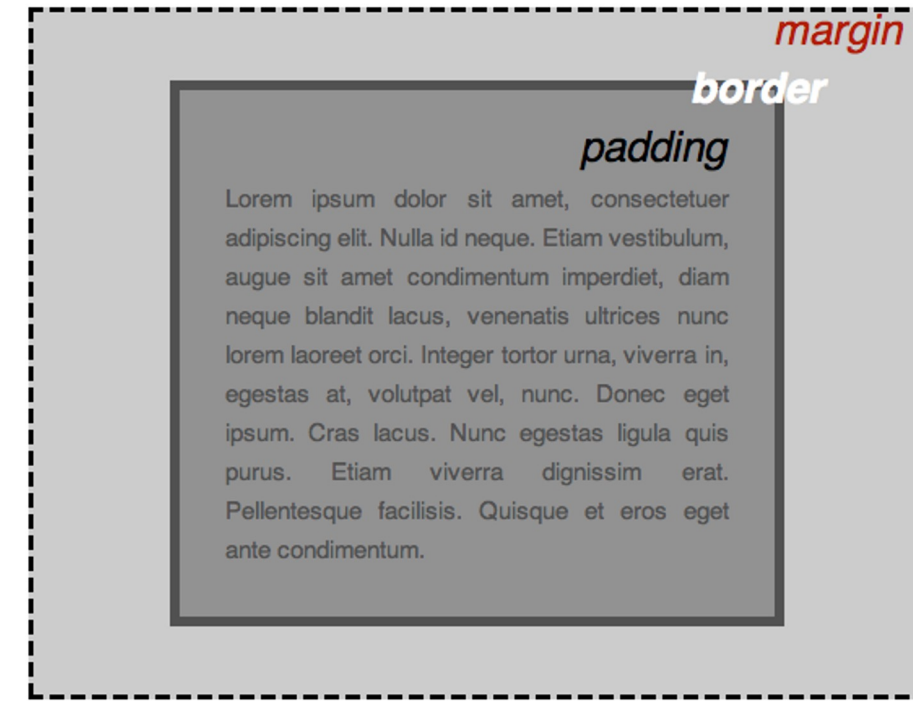
`<a>`

A selecciona mas somente  
quando o mouse está em  
cima do link



# Modelo de Caixa

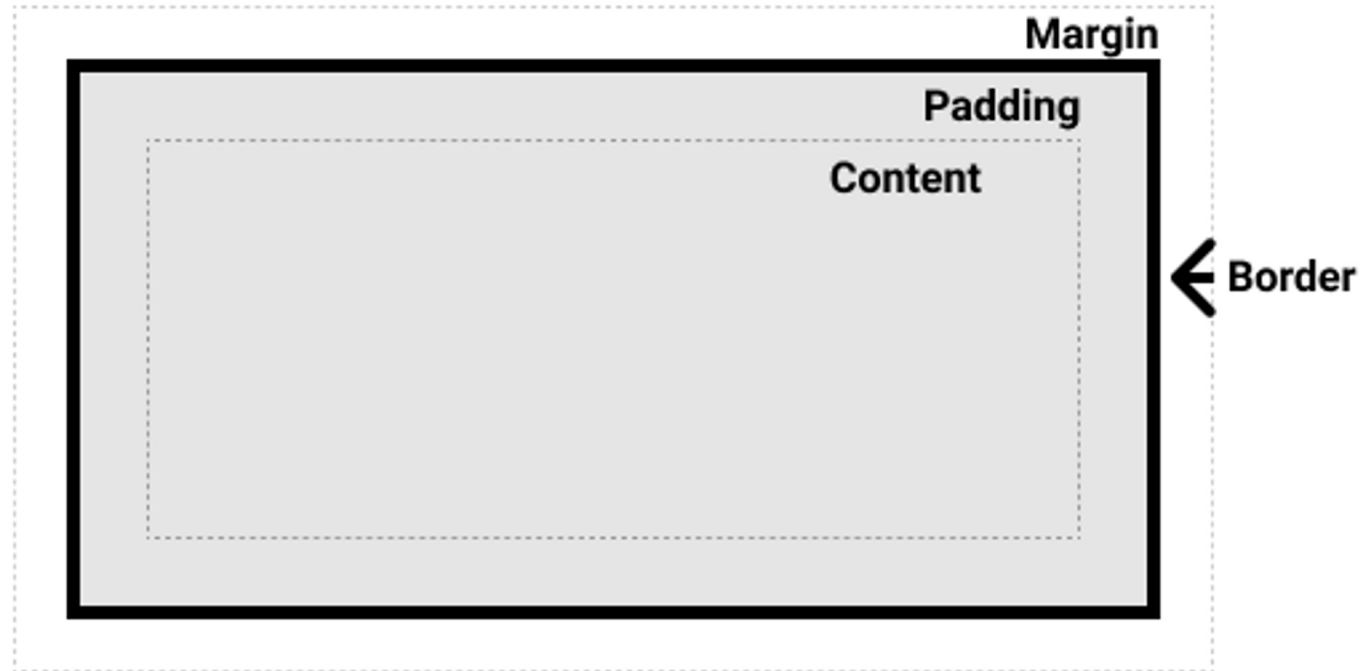
## Modelo de “caixa”

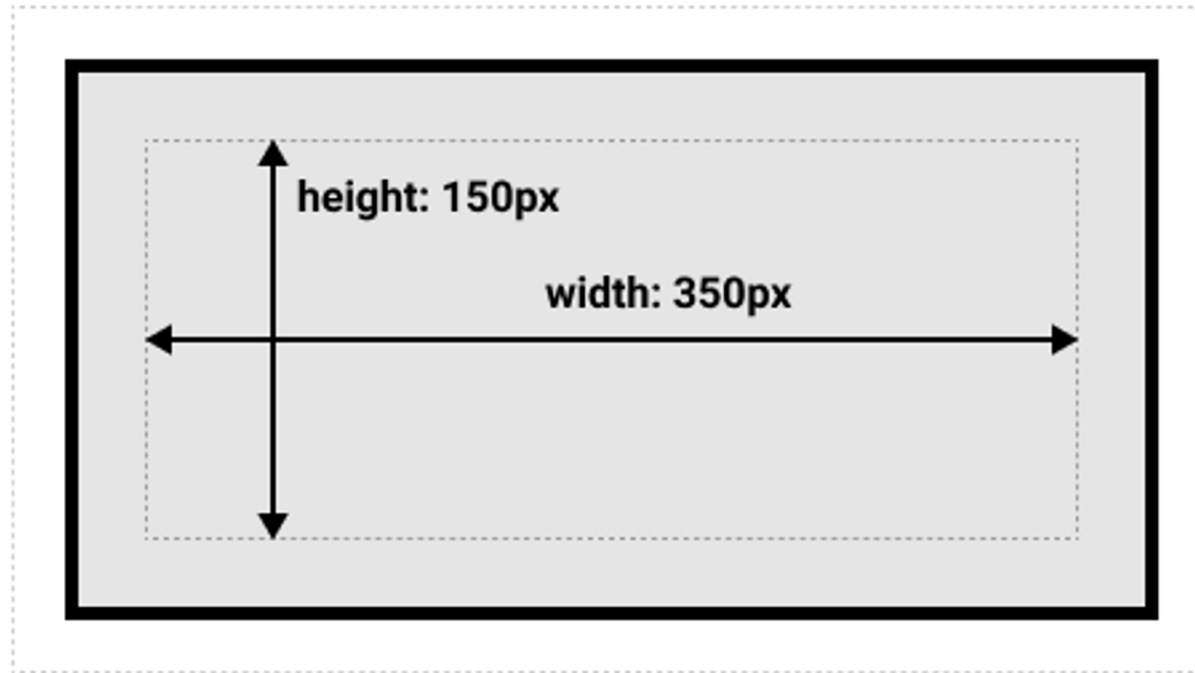


**padding**: o espaço ao redor do conteúdo (ex.: ao redor do texto de um parágrafo).

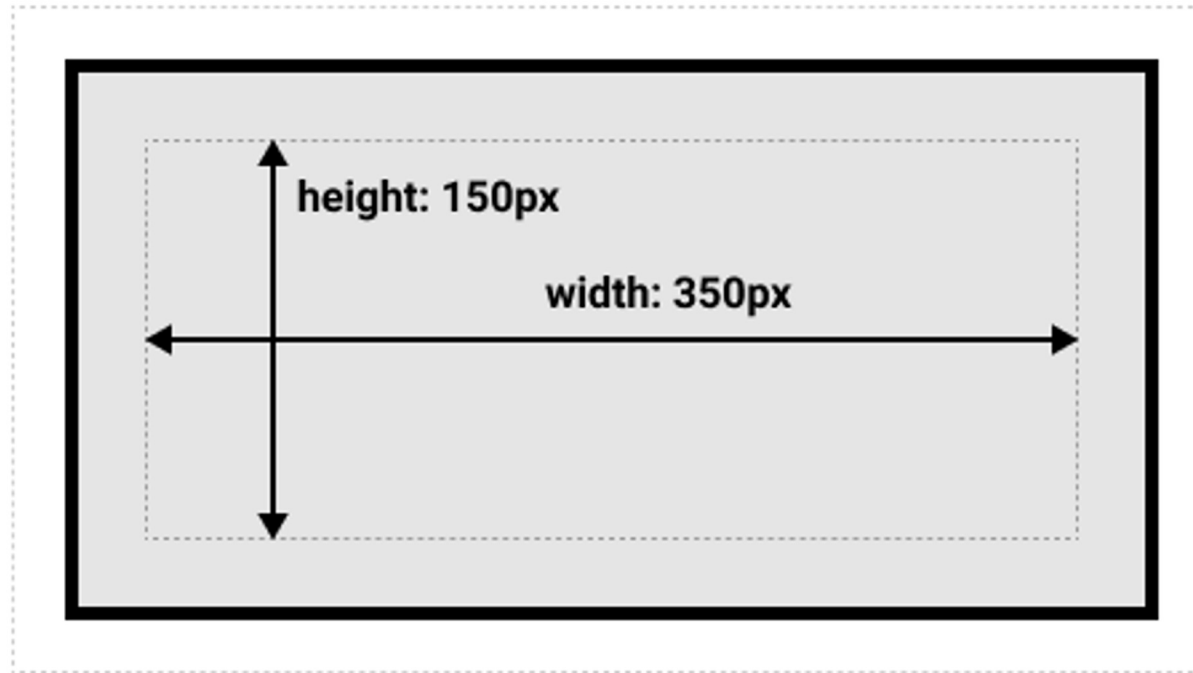
**border**: a linha sólida do lado de fora do padding.

**margin**: o espaço externo a um elemento.





```
.box {  
  width: 350px;  
  height: 150px;  
  margin: 10px;  
  padding: 25px;  
  border: 5px solid black;  
}
```



```
.box {  
  width: 350px;  
  height: 150px;  
  margin: 10px;  
  padding: 25px;  
  border: 5px solid black;  
}
```

Largura (width): 410px  
(350 + 25 + 25 + 5 + 5)

Altura (height): 210px  
(150 + 25 + 25 + 5 + 5)

# Estilizando a Página HTML

```
<p style="color: red">
```

```
<style>  
p {  
    color: red;  
}  
</style>
```

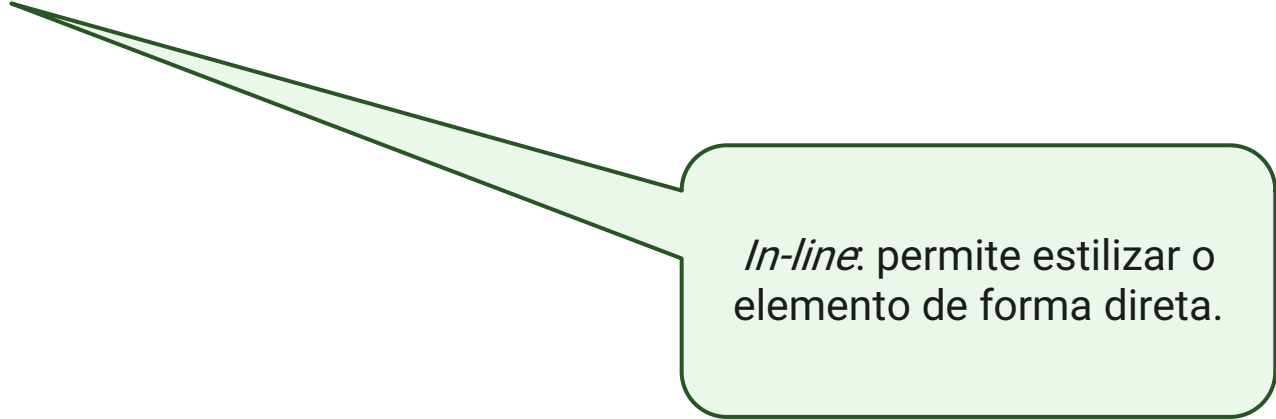
```
<link href="arquivo.css" rel="stylesheet">
```

## Incluindo CSS na Página HTML

```
<p style="color: red">
```

```
<style>  
p {  
    color: red;  
}  
</style>
```

```
<link href="arquivo.css" rel="stylesheet">
```



*In-line:* permite estilizar o elemento de forma direta.



```
<p style="color: red">
```

```
<style>  
p {  
    color: red;  
}  
</style>
```

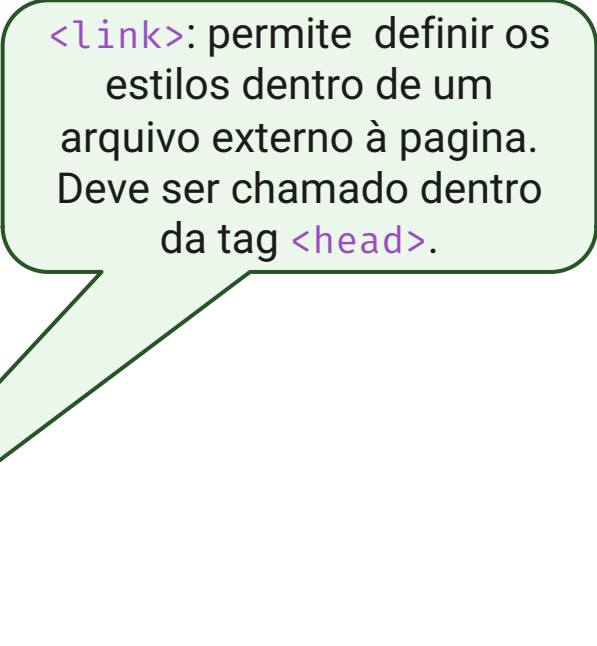
`<style>`: permite definir os estilos dentro da página. Pode ser incluída tanto dentro da tag `<head>` quanto da tag `<body>`.

```
<link href="arquivo.css" rel="stylesheet">
```

```
<p style="color: red">
```

```
<style>  
p {  
    color: red;  
}  
</style>
```

```
<link href="arquivo.css" rel="stylesheet">
```



`<link>`: permite definir os estilos dentro de um arquivo externo à página. Deve ser chamado dentro da tag `<head>`.

```
p {  
    color: red;  
    width: 500px;  
    border: 1px solid black;  
}
```

```
p {  
  color: red;  
  width: 500px;  
  border: 1px solid black;  
}
```

Cada linha de comando  
deve ser envolvida em  
chaves ({}).

```
p {  
    color: red;  
    width: 500px;  
    border: 1px solid black;  
}
```

Dentro de cada declaração, você deve usar dois pontos (:) para separar a propriedade de seus valores.

```
p {  
    color: red;  
    width: 500px;  
    border: 1px solid black;  
}
```

Dentro de cada conjunto de regras, você deve usar um ponto e vírgula (;) para separar cada declaração da próxima

```
p,  
li,  
h1 {  
    color: red;  
}
```

## Selecionando Múltiplos Elementos

```
p,  
li,  
h1 {  
    color: red;  
}
```

Seletores separados por  
vírgula.



```
.special {  
    color: orange;  
    font-weight: bold;  
}
```

```
.special {  
    color: orange;  
    font-weight: bold;  
}
```

Para quando você precisa encontrar uma maneira de selecionar um subconjunto de elementos sem alterar os outros

```
.special {  
    color: orange;  
    font-weight: bold;  
}
```

Para quando você precisa encontrar uma maneira de selecionar um subconjunto de elementos sem alterar os outros

```
<ul>  
    <li>Primeiro Item</li>  
    <li class="special" >Item Especial</li>  
    <li>Terceiro Item</li>  
</ul>
```

```
li.special {  
    color: orange;  
    font-weight: bold;  
}
```

Essa sintaxe significa  
"pegue qualquer elemento  
`li` que tenha uma  
classe `special`".

```
li.special {  
    color: orange;  
    font-weight: bold;  
}
```

Para aplicar em outros elementos utilizando a classe, temos que adicionar o outro elemento na lista de seletores.

```
li.special,  
span.special {  
    color: orange;  
    font-weight: bold;  
}
```

```
li em {  
    color: purple;  
}
```

```
h1 + p {  
    font-size: 200%;  
}
```

```
li em {  
  color: purple;  
}
```

Seleciona um elemento **em**  
que está aninhado dentro  
de um elemento **li**.

```
h1 + p {  
  font-size: 200%;  
}
```

```
li em {  
    color: purple;  
}
```

```
h1 + p {  
    font-size: 200%;  
}
```

Seleciona um parágrafo **p** que vem diretamente após um título **h1** no mesmo nível de hierarquia do documento HTML.



```
a:link {  
    color: pink;  
}
```

```
a:visited{  
    color: green;  
}
```

```
a:hover{  
    text-decoration: none;  
}
```

```
a:link {  
    color: pink;  
}
```

```
a:visited{  
    color: green;  
}
```

```
a:hover{  
    text-decoration: none;  
}
```

Estado é definido com a utilização dos dois pontos : e o nome do estado (depende de cada elemento HTML).

# Express Server

```
const express = require('express');  
const bodyParser = require('body-parser');  
  
const app = express();  
  
app.use(bodyParser.json());  
app.use(express.static(`${__dirname}/public`));  
  
app.listen(3001, () => console.log('Servidor na porta 3001.'));  
  
...
```

```
const express = require('express');  
const bodyParser = require('body-parser');
```

```
const app = express();
```

```
app.use(bodyParser.json());  
app.use(express.static(`${__dirname}/public`));
```

```
app.listen(3001, () => console.log('Servidor na porta 3001.'));
```

```
...
```

Coloque os arquivos CSS dentro de um diretório chamado **public** dentro do seu servidor.

```
const express = require('express');  
const bodyParser = require('body-parser');  
  
const app = express();  
  
app.use(bodyParser.json());  
app.use(express.static(`${__dirname}/public`));  
  
app.listen(3001, () => console.log('Servidor na porta 3001.'));  
  
...
```

Dentro da página HTML,  
utilize uma tag `<link>`  
dentro da tag `<head>` para  
utilizar as definições feitas  
no arquivo.