

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<style>
    #hover:hover {
        color: red;
    }

    #active:active {
        color: blue;
    }

    *:default {
        box-shadow: 5px 5px greenyellow;
    }

    #first-child p:first-child {
        color: blue;
    }

    #first-child p:last-child {
        color: green;
    }

    #nth-child p:nth-child(2n) {
        color: green;
    }

    p {
        font-weight: bold;
    }
</style>
<title>PSEUDO-CLASSES</title>
```

```
</head>
<body>
```

```
<!--
```

## PSEUDO-CLASSES

Uma pseudo-classe CSS é uma palavra-chave adicionada a um seletor para especificar um estado especial do elemento selecionado.

Por exemplo, `:hover` pode ser usado para alterar a cor de um botão quando o usuário passar o cursor sobre ele.

Além disso, Pseudo-classes permitem que você aplique um estilo a um elemento não apenas em relação ao conteúdo da árvore do documento, mas também em relação a fatores externos como o histórico de navegação (`:visited`, por exemplo), o status do seu conteúdo (como `:checked` em certos elementos de um formulário), ou a posição do mouse (como `:hover`, que permite saber se o mouse está sobre um elemento ou não).

A sintaxe de uma Pseudo-classe seria:

```
seletor:pseudo-classe {
  propriedade: valor;
}
```

As Pseudo-classes que podemos utilizar são:

```
:active - Adicionar formatação para quando o mouse é clicado no elemento;
:checked - Adicionar formatação para quando um radio ou checkbox é checado;
:default - É usado quando temos um conjunto de elementos "radio", "checkbox", "button" ou "option" num campo de
formulário. Se algum desses elementos tiver algum comportamento padrão como "checked" ou "required", podemos usar
o "default" para aplicar um CSS somente ao elemento que tiver um comportamento padrão;
:dir() - Adiciona um comportamento para elementos que estejam direcionados à direita ou esquerda, para isso ele vai
receber um valor "rtl" ou "ltr", detalhe importante, a tag também deverá estar marcada com o atributo "dir:rtl"
ou "dir:ltr";
:disabled - Para desabilitar algum comportamento padrão de um elemento;
:empty - Para formatar elementos que não contenham filhos - conteúdo;
:enabled - Para habilitar algum comportamento que por padrão estaria desabilitado;
:first - Usada para representar a 1ª página de um elemento marcado com @page;
```

:first-child - Representa qualquer elemento que seja o 1º filho de um determinado elemento;

:first-of-type - Representa o primeiro elemento do seu tipo dentre os elementos existentes dentro de um elemento pai;

:fullscreen - Representa todos os elementos que estiverem do modo tela cheia;

:focus - Para adicionar alguma formatação ao elemento quando ele recebe o focus na navegação;

:hover - Adicionar formatação para quando o mouse está por cima do elemento;

:indeterminate - Representa elementos que estejam de alguma forma ligados a elementos de checagem, como "radio", "checkbox", como os labels ligados a esses elementos, por exemplo;

:in-range - Representa uma tag que possua valores de input para "min" e "max", se receber um valor que está entre o "min" e "max" ele irá mostrar a formatação que está em no in-range;

:invalid - Representa qualquer elemento <input> ou <form> cujo conteúdo não esteja válido, por exemplo elemento que não obedeceram o "required";

:lang() - Corresponde a elementos com base na linguagem em que eles estão determinados, por exemplo, elementos que recebam algum unicode específico de uma língua;

:last-child - Representa qualquer elemento que é o último filho de seu elemento pai;

:last-of-type - Representa o último elemento irmão entre uma lista de elementos filhos de um elemento pai;

:left - Representa todas as páginas à esquerda de um documento marcado com @page;

:link - permite que você selecione os links dentro de um elemento. Ela seleciona todos os links, até mesmo os que não foram visitados, incluindo os links já estilizados em outras classes ou ids com o :hover, :active ou :visited.

Para um funcionamento adequado é essencial que ela venha antes das regras: :visited – :hover – :active. O :focus é uma pseudo-class geralmente usada antes de a: hover ou depois, dependendo do resultado esperado.

:not() - Seleciona todos os elementos filhos dentro de um especificado menos o seletor que for colocado dentro como argumento em not(), é como dizer "quero todos menos esse";

:nth-child() - Seleciona elementos com base em suas posições em um grupo de elementos irmãos, usando os valores odd, even ou notação funcional;

:nth-last-child() - Faz a mesma coisa que o nth-child() só que de baixo para cima começando pelo último elemento;

:nth-last-of-type() - Seleciona um elemento pelo tipo, dentro de um elemento pai, começando pelo último elemento, utiliza números para identificar o elemento;

:nth-of-type() - Seleciona um elemento pelo tipo, dentro de um elemento pai, começando de cima para baixo, utiliza números para identificar o elemento;

:only-child - Representa um elemento sem nenhum elemento-irmão;

:only-of-type - Representa qualquer elemento dentro de um elemento pai que não possua irmãos de um determinado tipo;

:optional - Representa qualquer elemento <input>, <select>, ou <textarea> que não contenha o atributo required;

:out-of-range - Representa um elemento de <entrada> de input cujo valor atual está fora dos limites especificados pelos atributos "min" e "max" num input;

:read-only - Representa um elemento de entrada, como "input" e "textarea" onde não é possível editar texto;

:read-write - Representa um elemento (como o input ou textarea) que é editável pelo usuário;

:required - Adicionar formatação para quando um elemento requer preenchimento obrigatório;  
:right - Representa todas as páginas à direita de um documento marcado com @page;  
:root - Se equipara à raiz de uma árvore, que por sua vez representa o documento. Aplicado ao HTML, :root representa o elemento <html> e é idêntico ao seletor html, exceto que sua especificidade é mais alta. Por através dele podemos definir nossas variáveis CSS;

:scope - Pode ser usada como o ":root", mas pode ser usado para trazer uma formatação especial a tags que utilizem o atributo "scoped" para trazer algum tipo de formatação sobre essas tags;

:target - Representa um único elemento (o elemento alvo) com uma id correspondente ao fragmento da URL.

Por exemplo, a seguinte URL tem um fragmento (denotado pelo sinal de #) que aponta para o elemento chamado

section2:

<http://www.example.com/index.html#section2>

O seguinte elemento será selecionado pelo seletor :target quando a URL for igual acima:

```
<section id="section2">Exemplo</section>
```

:valid - Representa qualquer <input> ou outro elemento do <form> cujo conteúdo foi validado com sucesso. Isso permite, facilmente, adicionar uma aparência que ajude o usuário a identificar os campos validados.

:visited - Adicionar formatação para quando um link já foi visitado;

Veja um exemplo no CSS implementado na tag head logo acima...

-->

```
<p id="hover">
```

Passe o mouse por cima e veja mudar de cor (:hover)

```
</p><br><br>
```

```
<p id="active">
```

Click e segure e veja mudar de cor (:active)

```
</p><br><br>
```

```
<label for="check">Como possuí valor padrão vai ter o box-shadow (:default)</label>
```

```
<input type="checkbox" checked name="check"><br><br>
```

```
<div id="first-child">
```

```
<p>
```

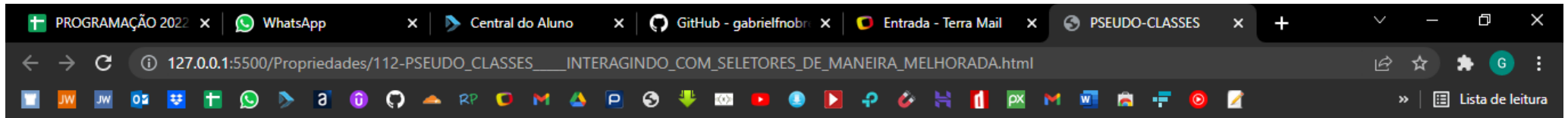
```
    1º Filho (first-child)
  </p>
  <p>
    2º Filho
  </p>
  <p>
    3º Filho (last-child)
  </p>
</div><br><br>

<div id="nth-child">
  <p>
    1º Filho
  </p>
  <p>
    2º Filho (nth-child(2n))
  </p>
  <p>
    3º Filho
  </p>
  <p>
    4º Filho (nth-child(2n))
  </p>
</div>
```

```
</body>
```


```
</html>
```

RESULTADO NO NAVEGADOR:



Passe o mouse por cima e veja mudar de cor (:hover)

Click e segure e veja mudar de cor (:active)

Como possui valor padrão vai ter o box-shadow (:default) 

1° Filho (first-child)

2° Filho

3° Filho (last-child)

1° Filho

2° Filho (nth-child(2n))

3° Filho

4° Filho (nth-child(2n))

