





# CSS – Folha de Estilo em Cascata

---

Prof.º Me Edson Martin Feitosa

# Introdução à CSS



Cascading Style Sheets (Folhas de Estilo em Cascata), assim como a HTML, não é uma linguagem de programação.



CSS é uma **linguagem de folha de estilos** usada amplamente na web e criada com o propósito principal de estilizar páginas HTML.

Seletor

p

{

color:

red;

}

Valor

Propriedade

# Introdução à CSS

- A linguagem CSS **funciona através de regras** com o principal **objetivo de estilizar visualmente o conteúdo HTML** da página.
- Com regras, podemos selecionar um ou mais elementos em uma página e aplicar estilos de acordo:

# Anatomia do CSS

## Formas de aplicar valor a uma propriedade:

- Código de cores;
- Unidades associadas a números;
- Valores pré-definidos;
- Múltiplos valores
  - shorthand property
  - alternativa de valores
- Funções
- Pseudo-classes
- Pseudo-elemento

```
<style>
.seletor {
  /* Valores comuns encontrados em um CSS */
  color: #fff; /*código de cores*/
  font-size: 30px; /*unidades associadas a números */
  text-align: center; /* valores pré-definidos */
  /* múltiplos valores */
  border: solid 5px navy; /* shorthand property */
  font-family: Helvetica, sans-serif; /*alternativa de valores */
  /* funções */
  width: calc(100% - 100px);
  background-color: rgb(100, 30, 50);
}
/* pseudo-classe */
.seletor:hover {
  background-color: orangered;
}
/* pseudo-elemento */
.seletor::first-letter {
  font-size: 60px;
}
</style>
```

Aplicado a <div class="seletor">Conteúdo</div>

# Unidade de Medida

pt	pontos, em que 1 ponto = $1/72$ de polegada
pc	Paicas
em	ems
in	polegadas
cm	centímetros
mm	milímetros
%	parte porcentual de algum outro valor, portanto é proporcional.
px	pixels: É proporcional. Depende da resolução de vídeo.

# Tipos de seletores

- \* : seletor universal, aplica a todos os elementos.
- elemento : usar o próprio elemento como seletor.
- .classe : seletor de classe.
- #id : seletor de id.
- [atributo] : seletor de atributo.
- : : seletor de pseudo-classe.
- :: : seletor de pseudo-atributo.
- Combinar seletores (ex.: DIV e p)
  - Div + p : seletor irmãos adjacentes (um seguido do outro)
  - Div ~ p : seletor de irmãos dentro do mesmo pai (não precisa ser um seguido do outro)
  - Div > p : seletor filho (filho direto)
  - Div p : seletor descendente
- Div, p : aplicar o css nos dois elementos (aplicar o mesmo conjunto de css a mais de um seletor)

# Inserindo CSS em uma página

- Existe **três maneiras de inserir as regras** em uma página:
  1. (INLINE) Dentro do elemento, atributo style

```
<p style="color: sienna; margin-left: 20px">
Este é um parágrafo
</p>
```
  2. (INTERNO) Dentro da tag <head> utilizando <style>

```
<head>
  <title>Código CSS - Entendendo a folha
de estilos</title>
  <style type="text/css">
    h1{
      font-style: italic;
    }
  </style>
</head>
```
  3. (EXTERNO) Arquivo externo: utilizando a tag <link>

```
<link rel="stylesheet" type="text/css"
href="url do arquivo">
```



# Prioridades na aplicação de diversos css



Quando existem várias regras CSS a serem aplicadas no documento a última definida possui a prioridade.



As regras INLINE possuem prioridade sobre as outras.



Por questões de boas práticas deve-se priorizar arquivo EXTERNO.

# Cascata

- Uma propriedade aplicada em um elemento HTML é herdada pelos seus elementos internos:

```
<div style="color: sienna; margin-left: 20px">
```

```
<p>
```

Este é um parágrafo

```
</p>
```

```
</div>
```

- O elemento `<p>` herdará as propriedades do elemento `<div>`

# Cascata

- Quando várias regras se aplicam a um mesmo elemento o CSS obedece algumas regras de precedência:
  - Regras internas tem precedência sobre regras externas
  - Regras inline tem precedência sobre regras internas
  - Seletor de elemento tem menor precedência
  - Seletor de classe tem mais precedência que seletor de elemento
  - Seletor de id tem a maior precedência que seletor de classe

# Pseudo-Classe

- São tipos de classes pré-definidas, é escrito com o nome do elemento + dois pontos + nome da pseudo-classe.
- Exemplo:
  - **a:link** utilizado para links não visitados
  - **a:visited** utilizado para links visitados
  - **a:hover** quando o mouse está em cima do link (pode ser aplicado a qualquer elemento)
  - **a: focus** selecionado o link pela tecla tab do teclado
  - **a:active** mouse clicado ou enter pressionado

# Pseudo-Classe

- **:first-child** e **:last-child**: Essas pseudo-classes permitem estilizar o primeiro e o último filho de um elemento pai. Por exemplo, você pode usá-las para estilizar o primeiro ou o último item de uma lista.
- **:nth-child()**: Esta pseudo-classe permite selecionar elementos com base em sua posição em relação aos irmãos. Por exemplo, `:nth-child(odd)` seleciona elementos ímpares, e `:nth-child(3n)` seleciona a cada terceiro elemento.
- **:not()**: Você pode usar `:not()` para excluir elementos específicos de uma seleção. Por exemplo, `p:not(.destaque)` selecionará todos os parágrafos que não possuem a classe "destaque".
- **:nth-of-type()**: Funciona de maneira semelhante ao `:nth-child()`, mas seleciona elementos com base em seu tipo. Por exemplo, `p:nth-of-type(odd)` selecionará parágrafos ímpares.
- **:enabled** e **:disabled**: Estas pseudo-classes se aplicam a elementos de formulário e permitem estilizar elementos habilitados e desabilitados, respectivamente.
- **:checked**: Esta pseudo-classe é usada para estilizar elementos de formulário que estão marcados, como caixas de seleção ou botões de rádio.
- **:target**: Estiliza o elemento correspondente a uma âncora que foi vinculada por meio de um fragmento de URL. É frequentemente usado para criar efeitos de rolagem suave.
- **:empty**: Seleciona elementos que não contêm nenhum conteúdo ou apenas contêm espaços em branco.
- **:nth-last-child()**: Funciona de forma semelhante ao `:nth-child()`, mas conta os elementos a partir do final do elemento pai.
- **:required** e **:optional**: Essas pseudo-classes se aplicam a campos de entrada de formulário e permitem estilizar campos obrigatórios e opcionais, respectivamente.
- **:valid** e **:invalid**: Permitem estilizar campos de entrada de formulário com base em sua validade de acordo com as restrições definidas.

# Pseudo-elemento

Os pseudo-elementos são seletores especiais em CSS que permitem estilizar partes específicas de um elemento HTML. Eles são representados por dois pontos (:) e são usados para criar estilos para partes específicas de um elemento

## Exemplos:

**::before:** Permite adicionar conteúdo antes do conteúdo do elemento alvo. Você pode usá-lo para criar ícones, marcas ou outros elementos decorativos antes do conteúdo real.

**::after:** Funciona de maneira semelhante ao ::before, mas adiciona conteúdo após o conteúdo do elemento alvo.

**::first-line:** Estiliza a primeira linha de texto dentro do elemento. Isso é útil para aplicar estilos a cabeçalhos ou outros elementos de texto.

**::first-letter:** Estiliza a primeira letra de um elemento de texto. Isso é frequentemente usado para criar estilos iniciais de parágrafos ou cabeçalhos.

**::selection:** Estiliza o texto selecionado pelo usuário. Você pode definir estilos para a seleção de texto, como cores de fundo e cores de texto.

**::placeholder:** Estiliza o texto de espaço reservado em campos de entrada de formulário. Isso permite que você estilize o texto de exemplo que aparece antes que o usuário insira um valor.

**::marker:** Estiliza o marcador de lista (por exemplo, "•" em listas não ordenadas) de um elemento <li> em listas.

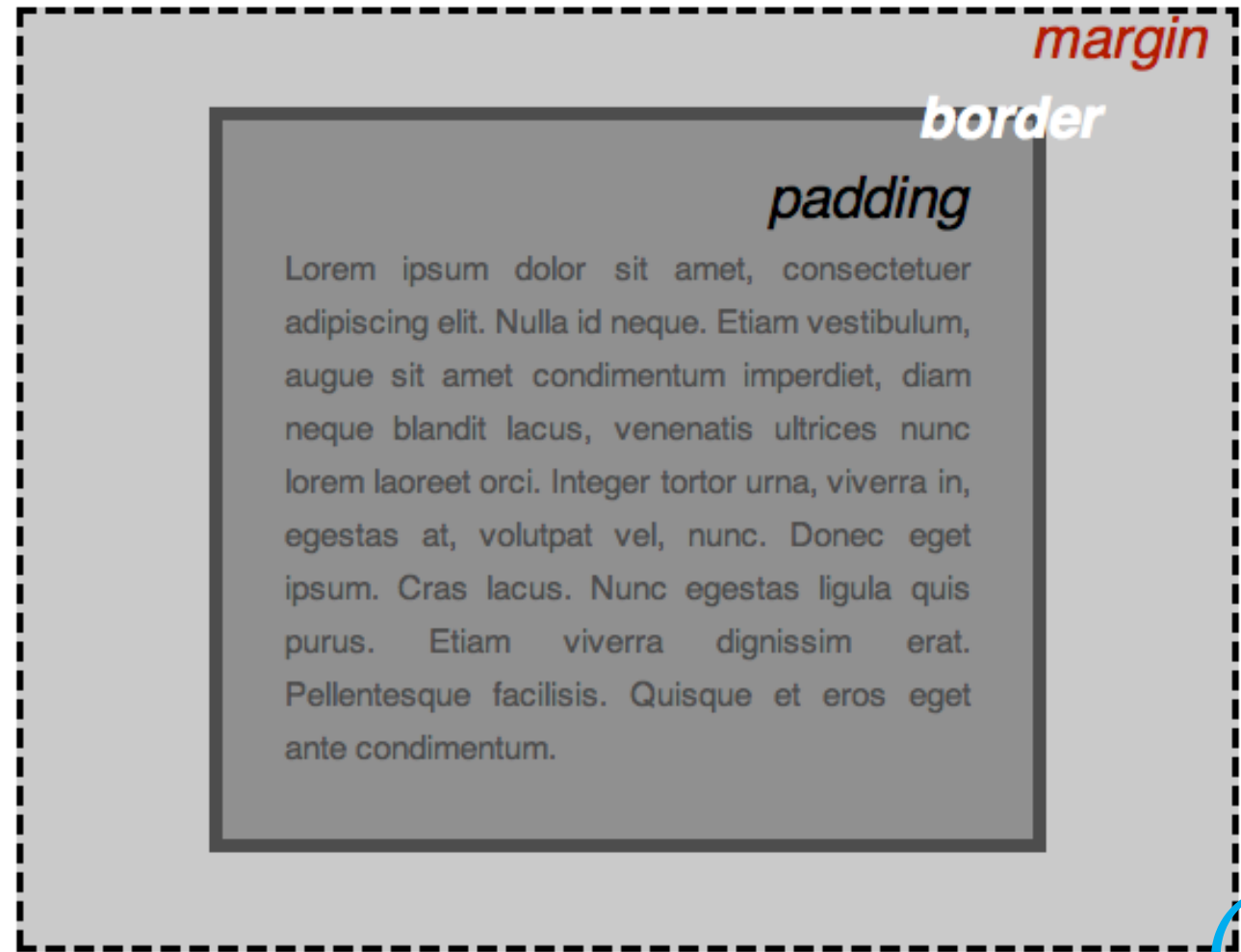
**::backdrop:** Usado principalmente para estilizar o plano de fundo de elementos que usam o elemento <dialog>. Pode ser usado para criar uma camada escura de fundo em caixas de diálogo modais.

**::cue:** Estiliza partes do conteúdo multimídia, como legendas de vídeo ou áudio. Por exemplo, ::cue pode ser usado para estilizar legendas de vídeo.

# Layout CSS

- O layout CSS é baseado principalmente no modelo de caixas. Cada um dos blocos que ocupam espaço na página tem propriedades:
  - 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

# Layout CSS

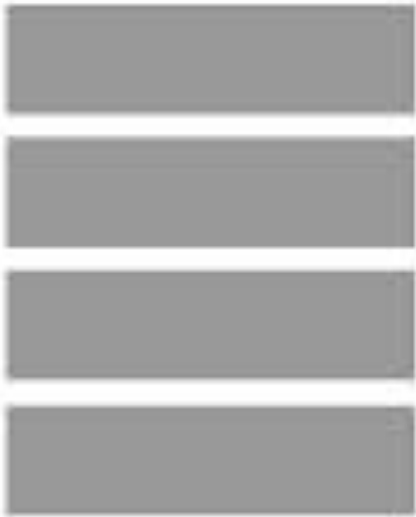




# Elementos in-line e elementos block-level

- A diferença básica é que o browser acomoda os elementos de **bloco** um abaixo do outro e os **inline** à esquerda um ao lado do outro.

**BLOCK:**



**INLINE:**



# Elementos in-line e elementos block-level

## Elementos **block-level**:

- div
- h1, h2, h3
- p
- ul, ol
- table
- form
- fieldset
- hr

## Elementos **in-line**:

- span
- a
- img
- td
- label
- input
- select
- textarea

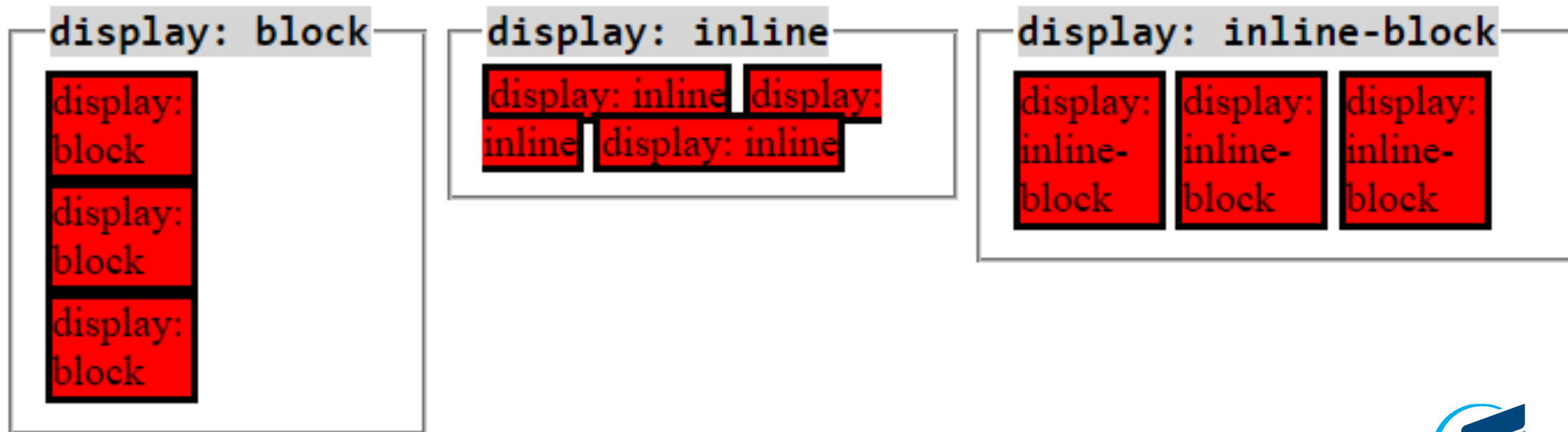


Facens

# Elementos in-line e elementos block-level

- Há ainda um terceiro tipo de elemento: o **inline-block** que nada mais é do que um elemento de bloco acomodado como um elemento inline.
- Com CSS você pode alterar o tipo do elemento. Com a propriedade `display` é possível utilizar os três valores:
  - Block
  - inline
  - inline-block

# Elementos in-line e elementos block-level



# Elementos in-line e elementos block-level

- A tag `div` é block-level, mas se aplicarmos o valor `inline` a propriedade `display` ela começará a se comportar como uma tag inline. Exemplo:

```
div {  
    display: inline;  
}
```

# Elementos in-line e elementos block-level

- Centralizando uma imagem

```
img
{
    display: block;
    margin: 0 auto;
}
```

- Imagens são elementos inline o que quer dizer que não ocupa o espaço na página e pode ter margin e outros valores de espaçamento atrelados a ele.
- margin: 0 auto; — Quando você indica dois valores em uma propriedade como margin ou padding, **o primeiro valor afeta o topo do elemento e o chão** (0, nesse caso), e no **segundo valor os lados esquerdo e o direito** (aqui, auto é um valor especial que divide o espaço horizontal disponível igualmente entre esquerda e direita).