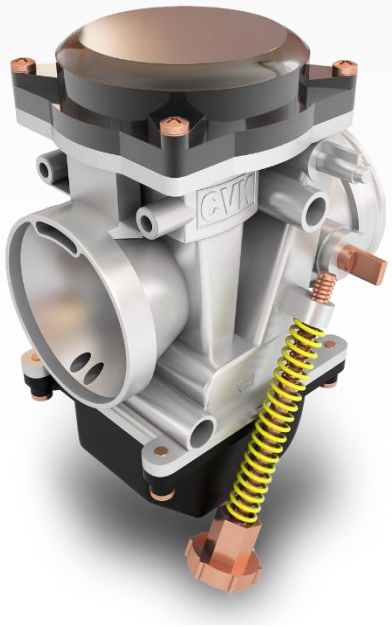
Desenvolvimento para Web

Módulo Básico



MOTORES DE RENDERIZAÇÃO

Motores

- ***Motores*** *de* ***Renderização***
  - ***Há*** uma ***diversidade*** de ***dispositivos*** que ***acessam*** a ***internet*** (***tablets***, ***smartphones***, ***pc's*** e *etc*)
  - ***Cada*** um ***desses meios*** de ***acesso utilizam*** um ***determinado browser*** para ***navegar*** na ***Web***
  - ***Não*** há como os ***desenvolvedores manterem*** um ***bom nível*** de ***compatibilidade*** com ***todos estes browsers levando*** em ***consideração*** a ***particularidade*** de ***cada um***

Motores

- ***Motores** de **Renderização***
  - Uma ***forma segura*** de ***manter*** o ***código compatível***, é ***nivelar*** o ***desenvolvimento*** pelos ***motores*** de ***renderização***
  - ***Cada browser utiliza*** um ***motor*** de ***renderização*** que é **responsável** pelo ***processamento*** do ***código*** da ***página***

Motores

- ***Motores** de **Renderização***
  - ***Principais browsers** e seus **motores***
    - **Webkit:** *Safari, Google Chrome*
    - **Gecko:** *Firefox, Mozilla, Camino*
    - **Trident:** *Internet Explorer 4 ao 9*
    - **Presto:** *Opera 7 ao 10*

Compatibilidade

- **HTML5**
  - ***Tabela*** de ***compatibilidade*** entre os ***browsers*** e ***alguns módulos*** do **HTML**

	Safari	Chrome	Opera	Firefox	IE 8	IE 9
Local Storage	s	s	s	s	s	s
Histórico de Sessão	s	s	s	s	s	s
Aplicações Offline	s	s	n	s	n	n
Novos tipos de campos	s	s	s	n	n	n
Form: Autofocus	s	s	s	n	n	n
Form: Autocomplete	n	n	s	n	n	n
Form: Required	s	s	s	n	n	n
Video, Audio e Canvas Text	s	s	s	s	n	s

Compatibilidade

- **HTML5**
  - A ***maioria***, mas ***não*** ***todas*** as ***funcionalidades*** da **HTML5**, já é ***suportada*** por ***um*** ou ***mais navegadores atuais***
  - ***Existem mecanismos desenvolvidos*** com a ***linguagem JavaScript*** que ***permitem detectar suporte*** para as ***funcionalidades criando condições*** de o ***desenvolvedor oferecer*** um ***conteúdo alternativo*** para uma ***funcionalidade*** ***não*** ***suportada*** por ***esses*** ou ***aquele navegador***
  - ***Modernizr*** é uma ***pequena biblioteca JavaScript destinada*** a ***detectar suporte nativo*** pelos ***navegadores*** para as ***funcionalidades*** das ***novas tecnologias*** para a ***Web*** (***download:*** *www.modernizr.com*)

Compatibilidade

- ***Exemplo:***
  - ***Uma*** vez ***linkada*** a ***biblioteca*** (***Modernizr***) a um ***documento Web***, é ***possível testar*** o ***suporte*** a ***uma nova funcionalidade***

```
if (Modernizr.canvas){
  // script para uso de canvas
} else {
    alert("Lamento, seu navegador não suporta canvas");
}
```

Template

- ***Exemplo (01):***
  - ***Template HTML5***

```
<!DOCTYPE html>
<html lang="pt-br">
  <head>
        <meta charset="utf-8" />
        <title>Exemplo de Template</title>
        <meta name="description" content="Template HTML5" />
        <meta name="keywords" content="lista de palavras-chave" />
        <meta name="author" content="Nome do Aluno" />
        <meta name="generator" content="HTML-kit 292" />
        <meta name="robots" content="all" />
<!-- continua... -->
```

Template

- ***Exemplo (01):***
  - ***Template HTML5***

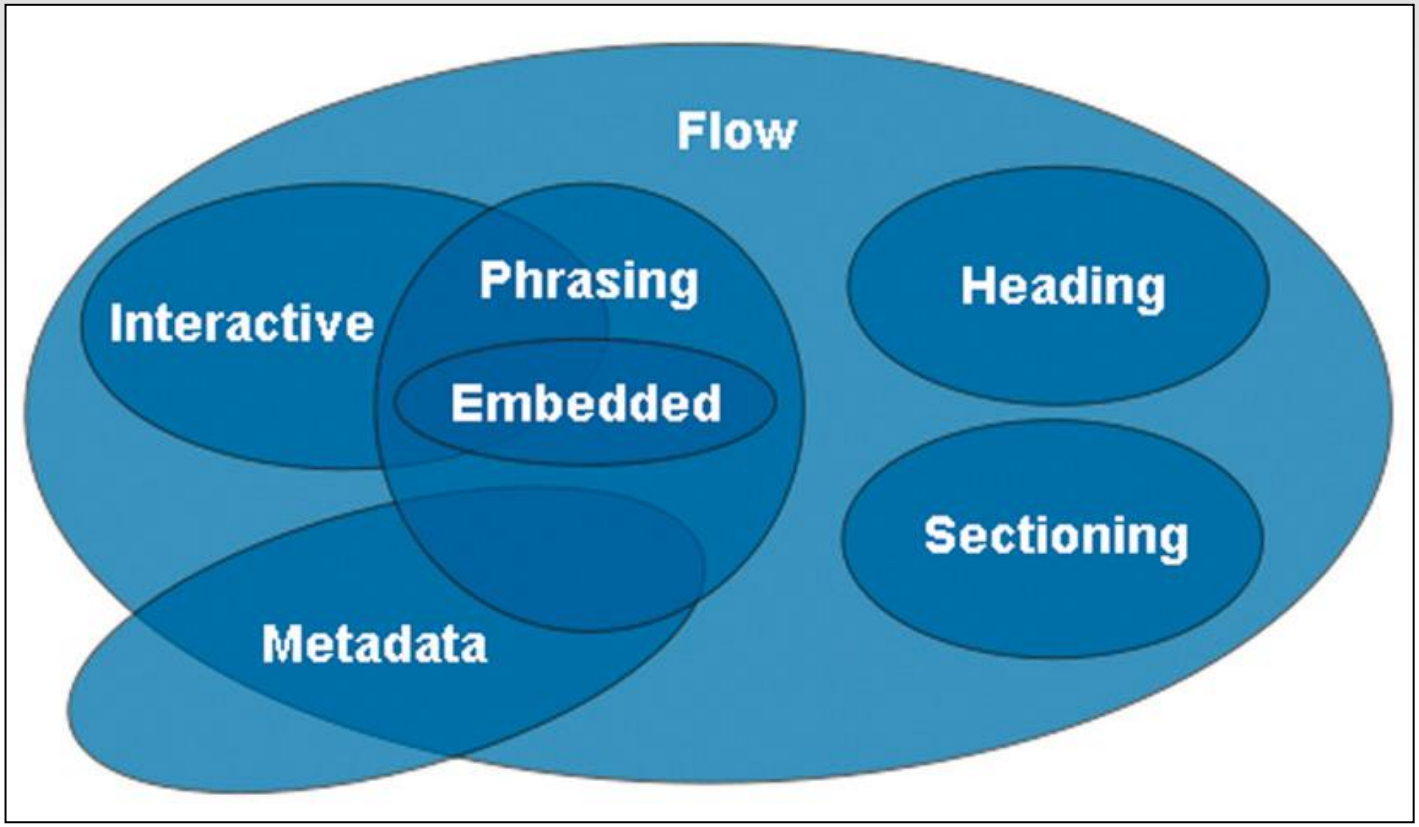
```
        <link rel="stylesheet" href="mais.css" />
        <style>
            /* estilos incorporados */
        </style>
        <script src="path/modernizr.min.js"></script>
        <script src="script.js"></script>
    </head>
    <body>
        <!-- Aqui os conteúdos da página -->
    </body>
</html>
```

Novidades na HTML5

- ***Novidades:***
  - A ***maioria*** dos ***elementos*** e ***atributos*** da **HTML4** ***continua válida*** na **HTML5**
  - ***Elementos antigos*** como (***h1***, ***p***, ***div***, ***span***, ***blockquote***) bem como os ***atributos*** (***class***, ***title***, ***accesskey***, ***name***, ***value***) etc, ***estão previstos*** na **HTML5** com as ***mesmas finalidades*** e ***valor semântico*** que ***tinham*** na **HTML4**
  - ***Alguns elementos*** de ***apresentação*** da **HTML4** ***sem valor semântico***, tais como os ***elementos*** (***b***, ***i***, ***u*** e ***small***), ***tiveram*** seu ***valor semântico definido*** na **HTML5**

Novidades na HTML5

- ***Categorias de Conteúdos:***



Atributos

- ***Atributos Globais***
  - ***São aqueles*** que ***podem*** ser ***usados*** com ***todos*** os ***elementos*** da **HTML5**
  - ***Podem*** ser ***utilizados não apenas*** com os ***elementos novos***, mas ***também*** com os ***elementos*** da **HTML4** que ***também fazem parte*** da **HTML5**

Atributos

- ***Atributos Globais***
  - ***accesskey*** *(****atributo tecla*** *de* ***acesso****)*
    - ***Destina-se*** a ***permitir*** que o ***usuário dê*** o ***foco*** ao ***elemento*** com ***uso*** do ***teclado*** (***atribui uma tecla*** de ***acesso*** ao ***elemento***)
    - O ***valor desse atributo*** é um ***caractere*** ou ***uma lista*** de ***caracteres*** do ***teclado***, ***separados*** por ***espaço***

Atributos

- ***Exemplo:***

```
<ul>
   <li><a href="/site-a.html" accesskey="a">Site A</a></li>
   <li><a href="/site-b.html" accesskey="b">Site B</a></li>
   <li><a href="/site-c.html" accesskey="c">Site C</a></li>
</ul>
```

  - O ***usuário*** deve ***prover meios*** de ***dar*** o ***foco*** aos ***links*** para os ***sites*** **A**, **B** e **C** quando o ***mesmo pressionar*** uma ***combinação*** de ***teclas*** (*Ctrl+Shift+a*, *Ctrl+Shift+b*, *Ctrl+Shift+c*) ***respectivamente***

Atributos

- ***Exemplo:***

```
<form action="/busca.jsp">
  <label>Buscar: <input type="busca" name="q" accesskey="b 0"></label>
  <input type="submit">
</form>
```

  - O ***foco*** deve ***ser dado*** ao ***campo*** de ***busca quando*** o ***usuário pressiona***, ***por exemplo***, a ***combinação*** das ***teclas*** *Ctrl+Shift+b*
  - ***Caso*** o ***usuário*** estiver ***utilizando*** um ***dispositivo móvel*** ou outro ***similar*** que possua ***somente teclado numérico***, o ***foco*** deve ser ***dado*** ao se ***pressionar*** a ***tecla*** *0* ***(zero)***

Atributos

- ***Atributos Globais***
  - ***class** (**atributo** de **classe**)*
    - ***Destina-se*** a ***atribuir*** uma ***classe identificadora*** para o ***elemento***
    - ***Trata-se*** de um ***atributo identificador geral***
    - Sua ***diferença*** para o ***identificador único id*** é que em um ***mesmo documento*** o ***nome*** da ***classe*** pode ser ***usado*** para ***identificar várias instâncias*** de ***um mesmo elemento***, bem como para ***identificar elementos distintos***

Atributos

- ***Exemplo:***

```
<p class="destaque">Texto marcado com a classe destaque</p>

<h1 class="destaque corum">Cabeçalho de nível 1
                    marcado com as classes destaque e corum</h1>
```

  - ***Para*** os ***elementos acima***, uma ***folha de estilo*** é ***capaz*** de ***identificar*** os ***diferentes elementos identificados*** com a ***classe*** *destaque* e ***estilizar todos eles*** com a ***mesma cor***

Atributos

- ***Atributos Globais***
  - ***contenteditable (atributo conteúdo editável)***
    - ***Atributo novo***, ***criado*** pela **HTML5**
    - ***Admite*** os ***valores*** *true* e *false*
    - ***Destina-se*** a ***permitir*** que o ***usuário edite*** os ***conteúdos*** do ***elemento*** no ***navegador***
    - É um ***atributo herdado***, pois, ***quando definido*** com o ***valor*** *true* para um ***elemento container***, faz com que ***seus elementos-descendentes*** sejam ***editáveis***

Atributos

- ***Exemplo:***

```
<div contenteditable="true">
   <h1>Cabeçalho de nível 1</h1>
   <p>Parágrafo <span>um</span></p>
   <p contenteditable="false">Parágrafo dois</p>
   <img src="img/foto.png" alt="Foto ilustrativa" />
</div>
```

  - No ***exemplo acima***, ***todos*** os ***conteúdos*** do *div* são ***editáveis***
  - ***Para*** o ***2º parágrafo***, ***não*** é ***possível editar*** o ***seu conteúdo textual*** *(Parágrafo dois)*, entretanto, é ***possível editá-lo*** como ***um todo*** (***apagando-o***, *por exemplo*)

Atributos

- ***Atributos Globais***
  - ***contextmenu** (**atributo menu** de **contexto**)*
    - ***Não consta*** da ***especificação*** do **W3C** para a **HTML5**, porém, está ***previsto*** na **HTML5.1** ***Nightly***
    - ***Trata-se*** de um ***atributo novo***, ***não suportado*** por ***nenhum navegador atual***
    - ***Esse atributo cria*** um ***menu*** de ***contexto personalizado*** para o ***elemento*** em que foi ***inserido***
    - ***Menu*** de ***contexto*** é ***aquele*** que se ***abre quando*** o ***usuário clica*** com o ***botão direito*** do ***mouse sobre*** o ***elemento*** para o ***qual*** foi ***projetado***

Atributos

- ***Exemplo (01):***

```
<form name="exemplo">
    <label>Nome base: <input name="nbase" type="text" contextmenu="menu_nome"></label>
    <menu type="context" id="menu_nome">
        <command label="Escolher randomicamente um nome"
            onclick="document.forms.exemplo.elements.nbase.value = escolherNome()">
        <command label="Preencher demais campos"
            onclick="preencherCampos(document.forms.exemplo.elements.nbase.value)">
    </menu>
</form>
```

  - No ***exemplo acima***, ***criamos*** um ***menu*** de ***contexto*** que se ***abre quando*** o ***usuário entra*** em ***um campo*** de ***texto*** de ***um formulário rotulado*** como "*Nome base*"
  - O ***evento não*** é ***necessariamente disparado com*** o ***clique*** do ***botão direito*** do ***mouse*** (***no exemplo***, será ***disparado quando*** o ***usuário promover*** o ***foco*** no ***elemento*** *input*)

Atributos

- ***Atributos Globais***
  - ***dir (atributo direção*** do **texto***)*
    - ***Destina-se*** a ***especificar*** a ***direção*** de ***escrita*** do ***texto***
    - ***Admite dois valores***
      - ***ltr: sigla*** do ***inglês*** *left to right*
      - ***rtl: sigla*** do ***inglês*** *right to left*

Atributos

- ***Exemplo:***

```
<p dir="ltr">Texto com atributo de direção ocidental</p>
<p dir="rtl">Texto com atributo de direção oriental</p>

<p style="direction:ltr">Texto com estilização de direção ocidental</p>
<p style="direction:rtl">Texto com atributo de direção oriental</p>
```

  - A ***especificação*** **HTML5** ***recomenda expressamente*** o ***uso*** do ***atributo*** *dir* em ***lugar*** da ***estilização*** com a ***propriedade*** *direction* ***(CSS)***

CSS

CSS

- ***Definição***
  - ***CSS** é a **abreviação** para o **termo** em **inglês Cascading Style Sheet***
  - *A **definição** mais **precisa** e **simples** para **folha** de **estilo** encontra-se no **site** do **W3C***

Folha de estilo em cascata é um mecanismo simples para adicionar estilos (por exemplo: fontes, cores, espaçamentos) aos documentos web

CSS

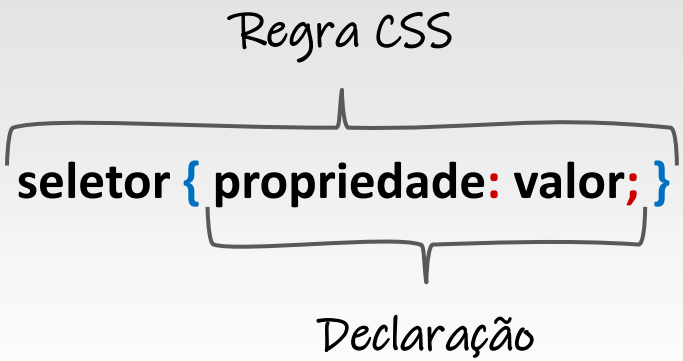
- ***Finalidade***
  - *A **HTML** foi **criada** para ser uma **linguagem exclusivamente** de **marcação** e **estruturação** de **conteúdos***
  - ***Não** cabe à **HTML prover informações** ao agente do usuário (**browser**) acerca da **apresentação** dos **elementos***
  - ***Por exemplo: cores** de **fontes**, **tamanhos** de **textos**, **posicionamentos** e **todo** o **aspecto visual** de **um documento** **não** devem ser **funções** da **HTML***
  - ***É responsabilidade** da **CSS** **todas** as **funções** de **apresentação** de um **documento** (**HTML** + **CSS**)*

CSS

- ***Regra***
  - ***Regra** de **estilo** ou de **estilização** é a **unidade básica** de **uma folha** de **estilo***
  - *O **termo** **unidade básica** **significa** a **menor porção** de **código** capaz de **produzir efeito** de **estilização***
  - ***Uma regra CSS** é **composta** de **dois fragmentos:** o **seletor** e a **declaração***
  - *A **declaração** **compreende** uma **propriedade** e **um valor***

CSS

- ***Regra***
  - ***Sintaxe:***



seletor: é o elemento ou elementos da marcação HTML nos quais a regra CSS será aplicada

Declaração: determina os parâmetros de estilização (propriedade e valor)

propriedade: define qual será a característica do seletor a ser estilizada

valor: é a quantificação ou a qualificação da estilização da propriedade

CSS

- ***Regra***
  - *A **sintaxe** abaixo, ,representa uma **regra CSS**, a qual poderá **conter** **várias declarações** **separadas** por **ponto** e **vírgula***
  - ***Sintaxe:***

```
p {
    color: white;
    background-color: black;
    font-size: 14px;
}
```

Seletor: é o **elemento** p (parágrafo)

Propriedades: são três – **color**, define a cor os textos, **background-color**, define a cor de fundo e **font-size**, define o tamanho da fonte

Valores: são três – **white**, define a cor branca, **black**, define a cor preta, e **14px**, define o tamanho da fonte

Declarações: são três – color: white; backgroung-color: black; font-size: 14px;

CSS

- ***Regra***
  - *Para **criar** um **grupamento** de **seletores**, **separamos um** do **outro** por **vírgula***
  - ***Exemplo:***

```
h2, p, ul, em { font-size: 18px; }
```

Definindo o tamanho da fonte igual a 18px para os conteúdos textuais dos elementos h2, p, ul e em (foram agrupados com uso de vírgula)

CSS

- ***Regra***
  - *Quando o **valor** de **uma propriedade** for **uma palavra composta**, **separada** por **espaços**, deve-se **usar sinais** de **aspas duplas** ou **aspas simples***
  - ***Exemplo:***

```
p { font-family: "Times New Roman"; }
p { font-family: 'Times New Roman'; }
```

```
p { font-family: Sans-serif; }
```

Não se usam aspas em palavras compostas separadas por hífen

CSS

- ***Regra***
  - *A **sintaxe** da **regra CSS** **não** é **sensível** ao **tamanho** de **caixa** da **fonte** e **múltiplos espaços** são **tratados** como **espaço simples***
  - ***Usar** ou **não** **espaços** entre os **componentes** da **regra CSS** **fica** a **critério** do **desenvolvedor***
  - ***Exemplo:***
    - ***Criar** uma **borda** de **1px** em **linha contínua** e na **cor vermelha** para o **conteúdo** do **elemento** **h1***

```
h1 { border: 1px solid red; }
h1 {border:1px solid red; }
h1 {border: 1px solid red; }
```

```
H1 { border: 1px solid RED; }
h1 {BORDER: 1px solid red; }
```

CSS

- ***Comentários***
  - ***Facultativo**, mas de **grande utilidade** na escrita de folhas de estilo, é o **sinal** para **inserir comentários***
  - ***Similar** a qualquer **linguagem** de **programação**, em **CSS existe** um **sinal próprio** para **marcar comentários***
  - ***Exemplo:***

```
/* Este é um comentário, em folha de estilo, em uma linha */

/* Este é um bloco de comentário, em folha de estilo, em
linhas diferentes contendo muitas informações sobre um
trecho da folha de estilo */
```

Modelo

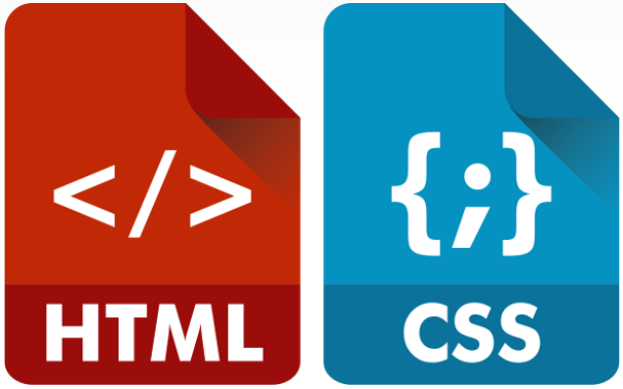
- ***Modelo de Formatação***
  - *No **modelo CSS** **cada elemento** **cria** **uma** **caixa** dentro da qual está **inserido** o **conteúdo** do **elemento***
  - ***Elemento** **nível** de **bloco**, a **caixa** tem **largura** **igual** à **largura** da **linha** (ocupa toda a largura da tela) e **altura** **suficiente** para **acomodar** o **conteúdo***
  - ***Elemento** **inline**, a **caixa** tem a **largura** do **conteúdo***

Modelo

- ***Container***
  - ***Denominado*** *de* ***bloco*** *de* ***conteúdo****, trata-se de um* ***box retangular*** *que* ***contém outros boxes chamados*** *de* ***descendentes***
  - *A* ***conceituação*** *de* ***container*** *é importante para o* ***entendimento*** *dos* ***mecanismos*** *de* ***cálculo*** *de* ***dimensões*** *e* ***posicionamento*** *de* ***boxes descendentes***
  - *Um* ***box descendente não precisa*** *ter necessariamente suas* ***dimensões confinadas*** *aos* ***limites*** *do* ***container****, podendo* ***ultrapassá-lo*** *(****overflow****)*

Modelo

- ***Elementos nível de bloco e boxes bloco***
  - *São aqueles* ***elementos*** *da* ***marcação HTML formatados visualmente*** *como* ***blocos*** *de* ***conteúdos***
  - ***Exemplos: parágrafos p*** *e os* ***cabeçalhos h1 – h6***



Modelo

- ***Box bloco Anônimo***

```
<div>
    Texto qualquer contido diretamente no DIV
    <p> Texto de um parágrafo contido do DIV </p>
</div>
```

Qual será o comportamento do texto contido diretamente no div? Inline ou de bloco?

Modelo

- ***Box bloco Anônimo***

```
<div>
    Texto qualquer contido diretamente no DIV
    <p> Texto de um parágrafo contido do DIV </p>
</div>
```

Quando inserimos em um container um elemento nível de bloco, estaremos forçando os conteúdos inseridos diretamente no container a se comportarem como box bloco (boxes blocos anônimos)

Modelo

- ***Elementos inline e boxes inline***
  - *São aqueles* ***elementos*** *da* ***marcação HTML*** *que* ***não formam novos blocos*** *de* ***conteúdo***
  - ***Conteúdo*** *é* ***distribuído*** *em* ***linha***
  - ***Exemplos: elemento*** *para* ***forte ênfase*** ***strong*** *e as* ***imagens***

Modelo

- ***Box inline anônimo***

```
<p> Texto <strong> enfatizado </strong> mais texto </p>
```

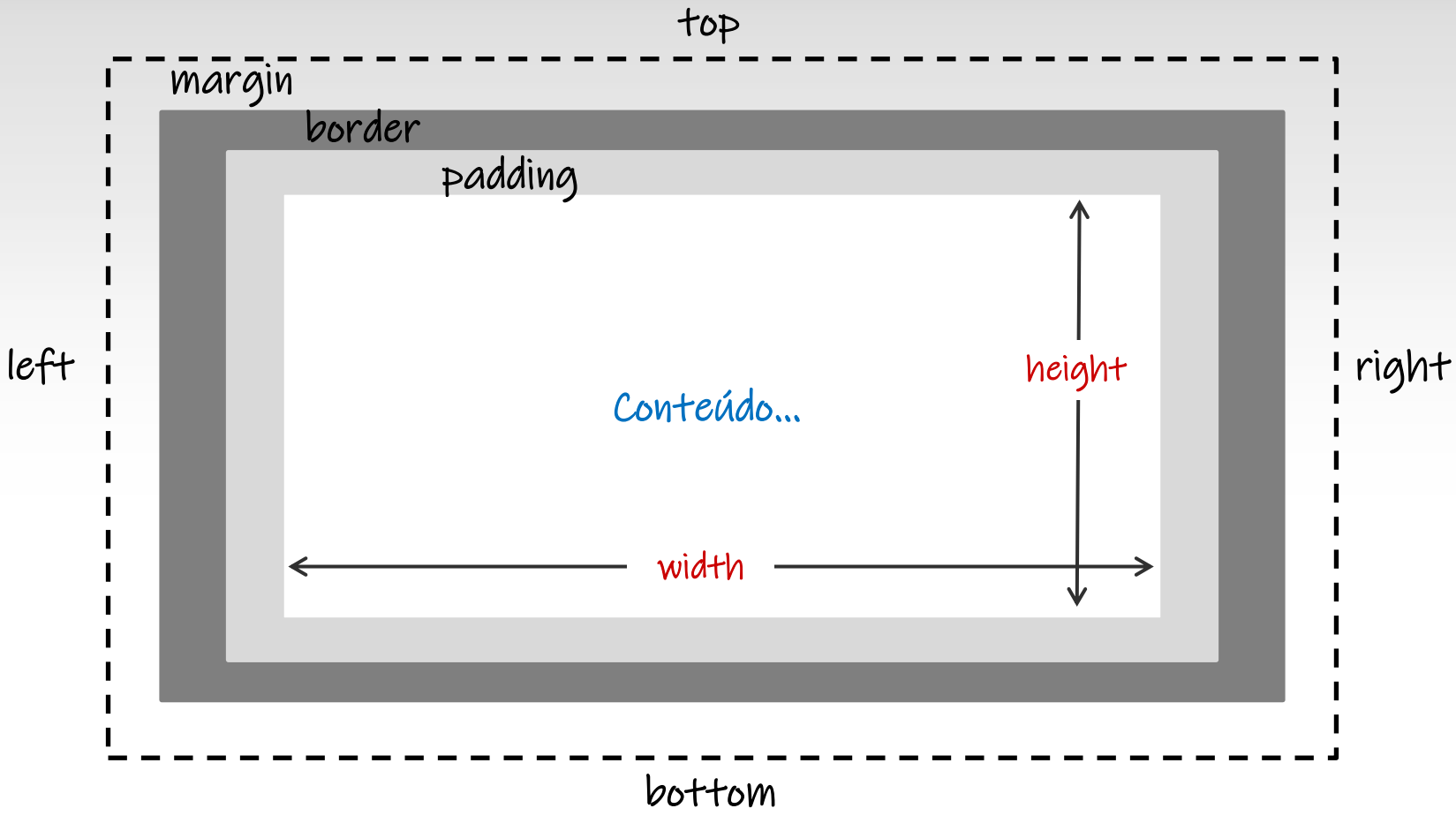
O elemento p criou um box bloco contendo três boxes inline.

O box para enfatizado, que foi gerado pelo elemento strong, e os boxes para Texto e mais texto, gerados por um elemento nível de bloco, o parágrafo.

Esses dois últimos são chamados de boxes inline anônimos.

Model CSS

- ***Box Model CSS***



Model CSS

- ***Box Model CSS***
  - ***Descreve*** *os* ***boxes criados*** *pelos* ***elementos*** *da* ***marcação HTML***
  - *No* ***diagrama destacamos*** *uma* ***área mais interior****,* ***denominada área*** *dos* ***conteúdos****, cujas* ***dimensões*** *(****largura*** *e* ***altura****) são* ***definidas*** *pelas* ***propriedades CSS width*** *e* ***height***
  - ***Uma área chamada*** *de* ***enchimento****, cuja* ***espessura*** *é* ***definida*** *pela* ***propriedade CSS padding***

Model CSS

- ***Box Model CSS***
  - ***Ao redor** do **enchimento**, temos **uma borda** cuja **espessura**, **cor** e **tipo** são **definidos** por **border***
  - *Um **espaço denominado margem** com **espessura** definida pela **propriedade margin***

Model CSS

- ***Box Model CSS***
  - ***Considere a marcação HTML e a estilização a seguir:***

```css
body {margin:0; padding:0; font-size:20px;}
p {
    width: 400px;
    text-align: justify;
    line-height:1;
    background: #f6f6f6;
    margin: 20px 80px 100px 40px;
    border: solid #ccc;
    border-width: 10px 30px 50px 15px;
    padding: 35px 50px 20px 0;
}
```

CSS

```html
<body>
<div>
  <p>Lorem ipsum dolor sit amet, consectetuer adipiscing elit. </p>
</div>
</body>
```

HTML

Model CSS

- ***Box Model CSS***
  - ***Considere a marcação HTML e a estilização a seguir:***

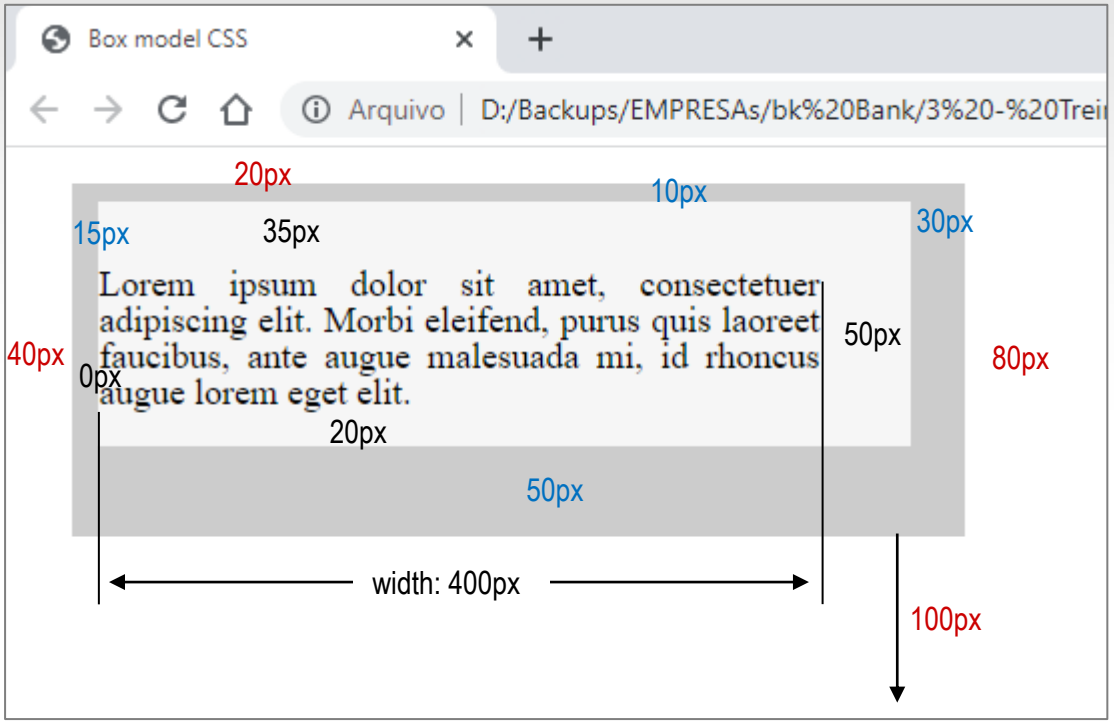


Figura 1 - Renderização do código (Box Model CSS)

Model CSS

- ***Box Model CSS Modificado***
  - *Se **analisarmos** minuciosamente o **Box Model CSS**, notamos que o mesmo é **contraintuitivo**, ou seja, uma vez que a **propriedade width** refere-se à **largura** do **conteúdo** do **box**, o que ocorre é que a **largura total** do **box** acaba sendo **variável** em relação à **propriedade width**, já que a ela são **acrescidas** as **larguras** de **padding** e **border** (variáveis) para se obter a **largura total** (o mesmo ocorre com a altura do box)*

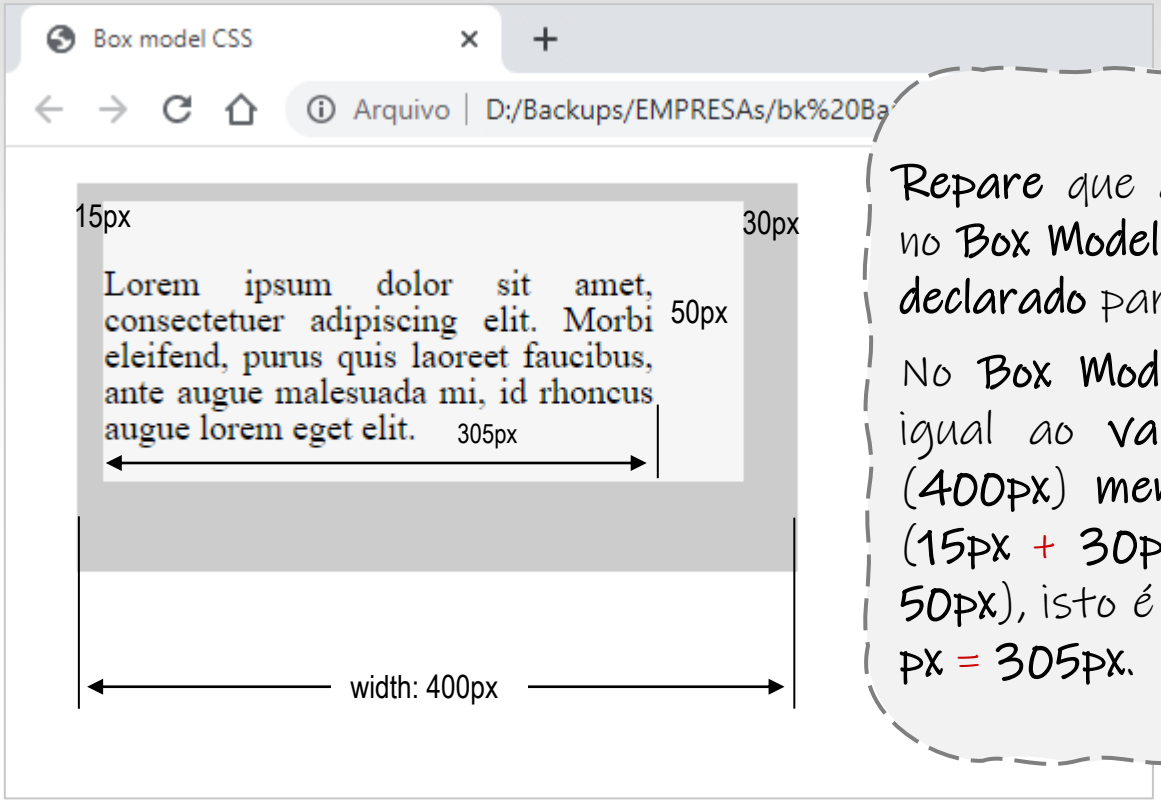
Model CSS

- ***Box Model CSS Modificado***
  - *As **CSS3 criaram** uma **nova propriedade CSS** denominada **box-sizing**, capaz de **alterar** esse **comportamento padrão** do **Box Model**, fazendo com que as **larguras** de **padding** e **border** sejam **incluídas** na **largura width** declarada por **regras CSS***
  - ***Utilize** a **regra** abaixo para **modificar** o **comportamento** do **Box Model** para **todos** os **boxes***

```
html { box-sizing: border-box; }
*, *:before, *:after { box-sizing: inherit; }
```

Model CSS

- ***Box Model CSS Modificado***



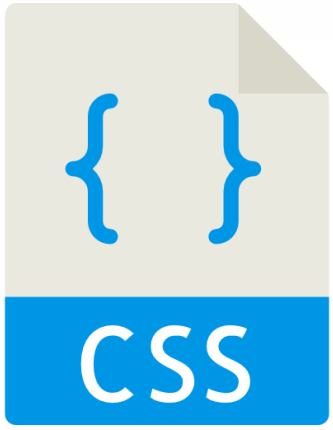
Repare que a largura do conteúdo que no Box Model tradicional é igual ao valor declarado para width (400px)

No Box Model modificado passa a ser igual ao valor declarado para width (400px) menos as larguras de border (15px + 30px = 45px) e padding (0 + 50px), isto é 400px – 15px – 30px – 50 px = 305px.

Figura 2 – Box Model CSS Modificado

Propriedades CSS

- ***Propriedades CSS (Box Model)***
  - *O **box criado** no **modelo** é um **quadrilátero** em que **cada um** dos **lados** é **identificado**, por um termo em **inglês designativo** da **posição** do **lado***
  - ***Os lados superior, direito, inferior** e **esquerdo** são identificados por **top**, **right**, **bottom** e **left**, respectivamente*



Propriedades CSS

- ***Propriedade Margin***
  - ***Define** as **dimensões** das **margens** de **um box***
  - ***Podemos definir cada** uma das **quatro margens** do **box** com **valores diferentes***
  - ***Exemplos:***

```
margin-top: 20px;
margin-right: 30px;
margin-bottom: 5px;
margin-left: 10px;
```

Sintaxe Convencional

```
margin: 20px 30px 5px 10px;
```

Sintaxe Abreviada

Propriedades CSS

- ***Propriedade Margin***
  - *As **propriedades** **CSS** **background**, **font**, **margin**, **border**, **border-top**, **border-right**, **border-bottom**, **border-left**, **border-width**, **border-color**, **border-style**, **transition**, **transform**, **padding**, **list-style**, **border-radius** e **outline** admitem a **sintaxe abreviada** (declarar uma lista de valores separados por um espaço)*
  - ***Exemplo:***
    - *Se as **quatro margens são iguais, declare um valor***

```
margin: 20px;  /* margem de 20px nos quatro lados */
```

Propriedades CSS

- ***Propriedade Margin***
  - ***Exemplos:***
    - *Se as* ***quatro margens*** *são* ***iguais duas*** *a* ***duas, declare dois valores***

```
margin: 15px 10px;  /* margens superior e inferior de 15px e direita e
                       esquerda de 10px */
```

    - ***Declare três valores***

```
margin: 20px 10px 15px;  /* margem superior de 20px margens direita e
                            esquerda de 10px e margem inferior de 15px */
```

Propriedades CSS

- ***Propriedade Padding***
  - ***Define** as **dimensões** do **enchimento** (ou espessura) entre o **conteúdo** e a **borda***
  - ***Podemos** definir **cada** um dos **quatro enchimentos** do **box** com **valores diferente***
  - ***Exemplos:***

```
padding-top: 20px;
padding-right: 30px;
padding-bottom: 5px;
padding-left: 10px;
```

Sintaxe Convencional

```
padding: 20px 30px 5px 10px;
```

Sintaxe Abreviada

Propriedades CSS

- ***Propriedade Border***
  - ***Define espessura**, o **estilo** e a **cor** das **bordas** do **box***
  - ***Cada** uma dessas **três características** da **borda** pode ser **declarada separadamente** para **cada lado** do **box***
  - ***border-width***
    - ***Define** a **espessura** da **borda***

```
border-top-width: 2px;
border-right-width: 3px;
border-bottom-width: 4px;
border-left-width: 1px;
```

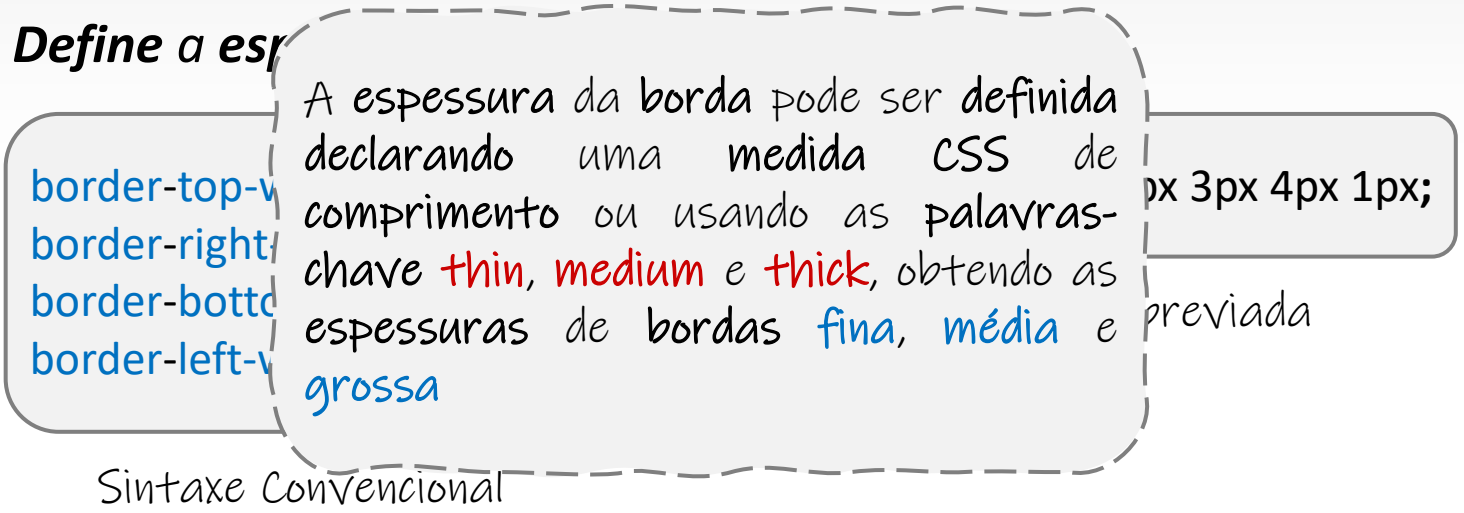
Sintaxe Convencional

```
border-width: 2px 3px 4px 1px;
```

Sintaxe Abreviada

Propriedades CSS

- ***Propriedade Border***
  - ***Define espessura**, o **estilo** e a **cor** das **bordas** do **box***
  - ***Cada** uma dessas **três características** da **borda** pode ser **declarada separadamente** para **cada lado** do **box***
  - ***border-width***
    - ***Define** a esp*



Propriedades CSS

- ***Propriedade Border***
  - ***border-color***
    - ***Define a cor da borda***

```
border-top-color: red;
border-right-color: yellow;
border-bottom-color: black;
border-left-color: cyan;
```

Sintaxe Convencional

```
border-color: red yelow black cyan;
```

Sintaxe Abreviada

Para declarar a cor da borda, podemos utilizar os valores CSS para cores ou o valor transparent

Propriedades CSS

- ***Propriedade Border***
  - ***border-style***
    - ***Define** o **estilo** da **borda***

```
border-top-style: solid;
border-right-style: ridge;
border-bottom-style: double;
border-left-style: inset;
```

Sintaxe Convencional

```
border-style: solid  ridge  double  inset;
```

Sintaxe Abreviada

Propriedades CSS

- ***Propriedade Border***
  - ***border-style***
    - *É **permitido** aplicar **nove estilos** para **bordas** ou **declarar** o **valor none** para **definir ausência** de **bordas***
    - *Pode parecer **estranha** e **inútil** a **declaração none**, mas, na prática, é **muito usada** para **retirar bordas** colocadas por **padrão** ou **declaradas** anteriormente em **elementos específicos** da **marcação** ou, ainda, para retirar a **borda-padrão** colocada em **imagens** que **são links***

Propriedades CSS

- ***Propriedade Border***
  - ***border-style***

Estilo	Descrição
none	*Define **espessura 0** para a borda*
hidden	*O **mesmo efeito** de **none**, mas com **precedência** na **resolução** de **bordas conflitantes***
dotted	*Borda **pontilhada***
dashed	*Borda **tracejada***
solid	*Borda **contínua** ou **sólida***
double	*Borda constituída de **duas linhas contínuas**. A **soma** das **espessuras** das **linhas** com a do **espaço** que as **separa** é **igual** ao **valor** de **border-width***

Tabela 1 – Estilos "border-style"

Propriedades CSS

- ***Propriedade Border***
  - ***border-style***

Estilo	Descrição
groove	*Borda com **aparência entalhada***
ridge	*Borda com **aparência** de **ressalto***
inset	*Borda em **baixo-relevo***
ouset	*Borda em **alto-relevo***

Tabela 1 – Estilos "border-style"

Propriedades CSS

- ***Propriedade Border***
  - ***border-style***

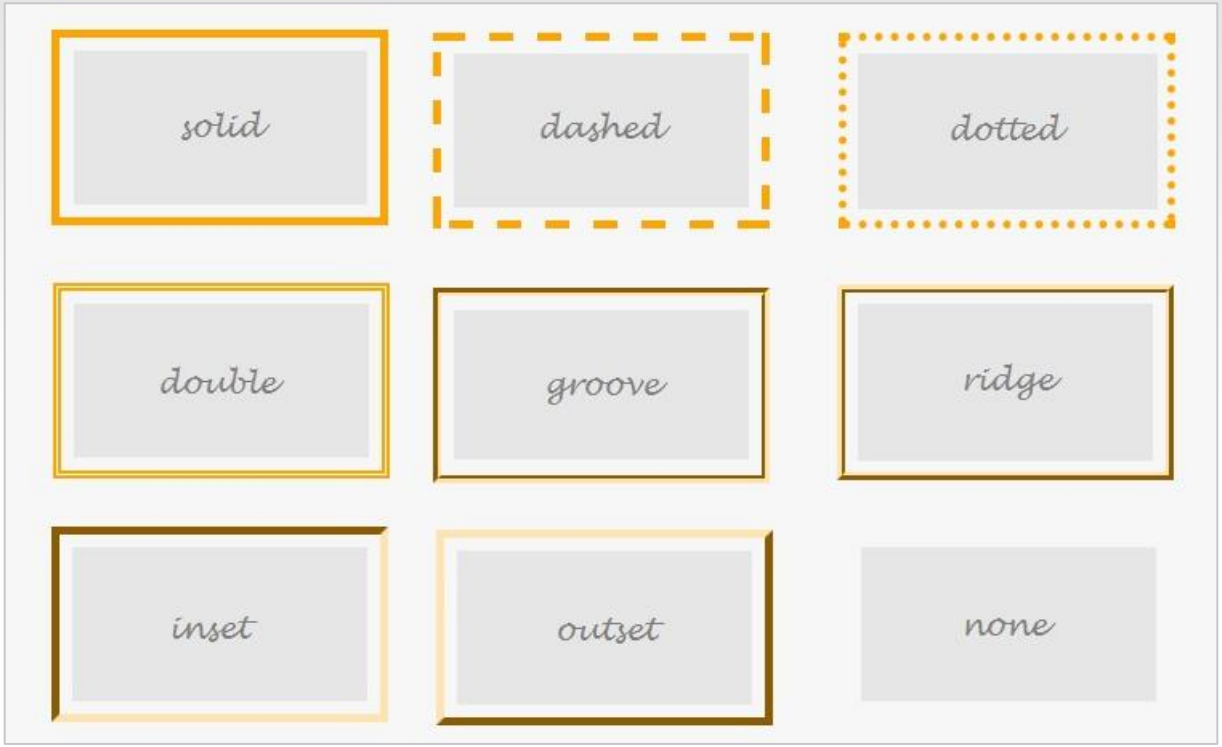


Figura 3 – Exemplos de "border-style"

Propriedades CSS

- ***Propriedade Border***
  - ***border***
    - ***Define** abreviadamente a **borda***

```
border-top: 1px solid red;
border-right: 2px ridge yellow;
border-bottom: 4px double black;
border-left: 6px inset cyan;
```

Sintaxe Convencional

```
border: 2px dotted white;
```

Sintaxe Abreviada

Propriedades CSS

- ***Propriedade Border***
  - ***border***
    - *Na **declaração abreviada** **border**, **não** é **obrigatório declarar** os **três valores***
    - *As **regras CSS** apresentada abaixo **são válidas***

```
border: 5px;
border: dotted;
border: red;
border: 2px  double;
border: solid  red;
border: 4px  blue;
```

Sintaxe Convencional

Propriedades CSS

- ***Propriedade Border***
  - ***border***
    - *Os **valores não declarados** são **interpretados** pelo **navegador** como sendo o **valor inicial** da **propriedade***
    - *Os **valores iniciais** das **três propriedades** são:*

```
border-width: medium;
border-style: none;
border-color:     /* o mesmo valor da propriedade color
                     do elemento em que se aplica a borda */
```

Categorias CSS

- ***Categorias de valores CSS***
  - *Para **aplicar** uma **regra CSS**, o **agente** de **usuário** (**browser**) **identifica** o **valor** da **propriedade** e **renderiza** o **elemento**, **casado** pelo **seletor**, de acordo com o **valor***
  - ***Exemplos:***

```
p { font-family: Arial, Sans-serif; }  /* estiliza p com fonte na
                                          família especificada (valor) */

p { width: 400px; }  /* estiliza p com largura 400px */

p { font-size: 120%; }  /* estiliza p com tamanho de fonte 1.2 vezes
                           o valor de referência */
```

Categorias CSS

- ***Categorias de valores CSS***
  - ***Exemplos:***

```
p { background-color: red; }  /* estiliza p com fundo na cor vermelha */

p { height: 2em; }  /* estiliza p com altura 2 vezes o valor de referência */
```

Observe que alguns valores são absolutos e outros relativos (porcentagem e em)

Categorias CSS

- ***Categorias de valores CSS***
  - *Os **valores CSS** podem ser **agrupados** em **oito categorias:***
    1. ***Palavra-chave***
    2. ***Número***
    3. ***Número não negativo***
    4. ***Número** com **unidade** de **medida***
    5. ***Número não negativo** com **unidade** de **medida***
    6. ***String***
    7. ***Notação funcional***
    8. ***Casos especiais***

Categorias CSS

- ***Categorias de valores CSS***
  - ***Palavra-chave***
    - *Um **valor CSS** é do tipo **palavra-chave** quando **expresso** por **uma string predefinida** nas **especificações***
    - *O **exemplo típico** para esse caso é quando **usamos palavra-chave** para **definir cores***

```
p {
    color: red;
    background-color: acqua;
    border-color: teal;
 }
```

Categorias CSS

- ***Categorias de valores CSS***
  - ***Palavra-chave***
    - ***Outros exemplos** de **palavra-chave** para **expressar valores CSS***

Palavra-chave	Utilizada
inherit	*Para definir uma propriedade que deverá ser herdada*
collapse	*Para definir bordas de células de tabelas que devam ser unidas*
italic	*Para definir fonte em itálico*
uppercase	*Para definir texto em caixa-alta*

Tabela 2 – Palavra-chave

Categorias CSS

- ***Categorias de valores CSS***
  - ***Número***
    - *Um **valor CSS** é do **tipo** **número** quando **expresso** por **um número** **inteiro** ou por um **número** **real***
    - *A **especificação adota** a **sintaxe** **<integer>** para **designar** **números** **inteiros** e **<number>** para **designar números** **reais***

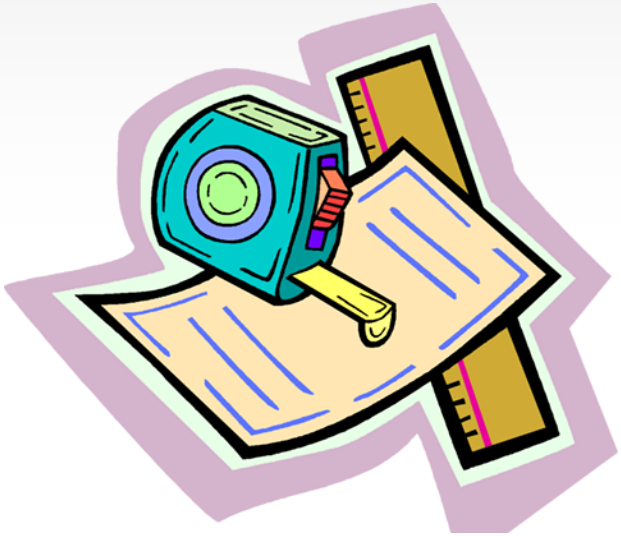
Categorias CSS

- ***Categorias de valores CSS***
  - ***Número não Negativo***
    - *Muitas **propriedades CSS** que **admitem** um **valor** do **tipo número** fazem **restrição** quanto à **faixa** de **números admitidos***
    - *Por exemplo, a **propriedade width**, destinada a **definir** a **largura** de um **elemento**, que **não admitem números negativos***
    - *Nesses casos, a **sintaxe prevista** nas **especificações** é **<non-negative-integer>** para **números inteiros não negativos** e **<non-negative-number>** para **números reais não negativos***

Categorias CSS

- ***Categorias de valores CSS***
  - ***Número com Unidade de Medida***
    - *Os **valores CSS**, quando **expressos** com **números** **seguidos** por **uma unidade** de **medida**, são **classificados** em **cinco categorias***

1. ***Comprimento***
2. ***Medida Relativa***
3. ***Medida Absoluta***
4. ***Porcentagem***
5. ***Ângulo***



Categorias CSS

- ***Categorias de valores CSS***
  - ***Número com Unidade de Medida***
    1. ***Comprimento***
       - ***Refere-se** às **medidas** **horizontal** e **vertical***
       - *A **sintaxe prevista** nas **especificações** para **designar** essa **categoria** é **<length>***
       - *Um **valor CSS** que usa uma **medida** de **comprimento** é formado por um **número** seguido da **abreviatura** para uma **unidade** de **medida***
       - ***Exemplo:** 14px, 12em, 18pt*

Categorias CSS

- ***Categorias de valores CSS***
  - ***Número com Unidade de Medida***
    2. ***Medida Relativa***
       - ***É aquela*** *cujo* ***valor*** *é determinado em* ***função*** *de outro* ***valor*** *de uma* ***propriedade*** *que lhe* ***serve*** *de* ***referência***
       - ***Definir*** ***medidas relativas*** *em uma* ***folha*** *de* ***estilo facilita*** *o* ***escalonamento*** *e possibilita servi-la para* ***diferentes tipos*** *de* ***mídia***

Categorias CSS

- ***Categorias de valores CSS***
  - ***Número com Unidade de Medida***
    2. ***Medida Relativa***
       - ***Unidades** de **medidas relativas** nas **CSS3***

Unidade	Relativa
em	*à **fonte-size** do **elemento** (ou do **elemento-pai**). **Elemento pai** é o **elemento** no qual um **elemento está contido***
ex	*ao **valor x-height** (**altura** da **letra x minúscula**) da **fonte** do **elemento***
px	*ao **dispositivo gráfico** (tela, por exemplo) de **renderização***
rem	*à **font-size** do **elemento raiz** do **documento** (**html**)*

Tabela 3 – Unidades de medidas relativas

Categorias CSS

- ***Categorias de valores CSS***
  - ***Número com Unidade de Medida***
    2. ***Medida Relativa***
       - ***Unidades** de **medidas relativas** nas **CSS3***

Unidade	Relativa
vw	*à **largura** da **viewport** (**área** de **renderização**)*
vh	*à **altura** da **viewport***
vm	*à **largura** ou **altura** da **viewport** (a **menor** das **duas**)*
ch	*à **largura** do **número** "0", **renderizado** de **acordo** com **font-size**. Se **não existir** "0" na **fonte especificada**, a **largura média** dos **caracteres deverá** ser **usada***

Tabela 3 – Unidades de medidas relativas

Categorias CSS

- ***Categorias de valores CSS***
  - ***Número com Unidade de Medida***
    3. ***Medida Absoluta***
       - *É **aquela** cujo **valor** é **determinado** e **fixo***
       - ***Essas unidades** são **úteis** para uso quando se **conhece** as **dimensões físicas** da **mídia** (tela, impressora etc.) para a qual a **folha** de **estilo** será **servida***

Categorias CSS

- ***Categorias de valores CSS***
  - ***Número com Unidade de Medida***
    3. ***Medida Absoluta***
       - ***Unidades** de **medidas absolutas** nas **CSS3***

Unidade	Descrição
in	*polegada; 1 polegada = 2,54 cm*
cm	*centímetro*
mm	*milímetro*
pt	*ponto; 1 ponto = 1/72 polegada*
pc	*pica; 1 pica = 12 pontos*

Tabela 4 – Unidades de medidas absolutas

Categorias CSS

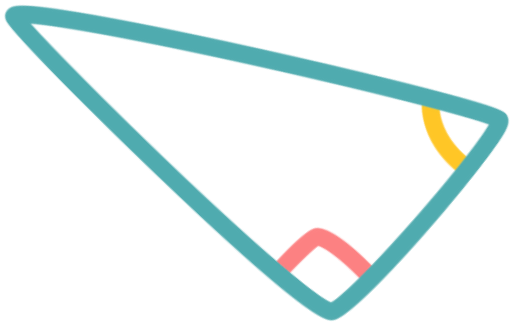
- ***Categorias de valores CSS***
  - ***Número com Unidade de Medida***
    4. ***Porcentagem***
       - *O **formato** para **definir** um **valor CSS** em **porcentagem** é um **número** **imediatamente seguido** pelo **sinal %***
       - *A **sintaxe prevista** nas **especificações** para **designar** essa **categoria** é **<percentagem>***
       - ***Porcentagens** são **valores dependentes** de **outro valor**, por **exemplo:** de **um valor** do **tipo <length>***

Categorias CSS

- ***Categorias de valores CSS***
  - ***Número com Unidade de Medida***
    4. ***Porcentagem***
       - *As **propriedades CSS** que admitem **valores** em **porcentagem** também definem qual o **valor** de **referência** a **considerar** para **cálculo** da **porcentagem***
       - *O **valor** de **referência** pode ser o **valor** de **outra propriedade** do **mesmo elemento** ao qual a **porcentagem** foi **aplicada**, o de um **elemento ancestral** ou o **valor** de um **contexto** de **formatação**, como a **largura** de um **bloco** de **conteúdo***

Categorias CSS

- ***Categorias de valores CSS***
  - ***Número com Unidade de Medida***
    5. ***Ângulo***
       - *O **formato** para **definir** um **valor CSS** em **medida angular** é um **número** imediatamente seguido por **uma unidade** de **medida angular***
       - *A **sintaxe prevista** nas **especificações** para **designar** essa **categoria** é **<angle>***



Categorias CSS

- ***Categorias de valores CSS***
  - ***Número com Unidade de Medida***
    5. ***Ângulo***
       - ***Unidades** de **medida angular** em **CSS***

Unidade	Descrição
deg	*Graus*
grad	*Grados*
rad	*Radianos*
turn	*Volta*

Tabela 5 – Unidades de medida angular

Categorias CSS

- ***Categorias de valores CSS***
  - ***Número com Unidade de Medida***
    5. ***Ângulo***
       - ***Unidades** de **medida angular** em **CSS***

Unidade	Descrição
turn	*Volta*

São usados, por exemplo, para definir propriedades destinadas à mídia speech (mídia falada, tal como leitores de tela) ou às transformações previstas nas CSS3

Tabela 5 – Unidades de medida angular

Categorias CSS

- ***Categorias de valores CSS***
  - ***Número não Negativo com Unidade de Medida***
    - ***Valores CSS expressos*** *com* ***números não negativos*** *com* ***unidade*** *de* ***medida*** *são* ***classificados*** *em* ***duas categorias*** *(**hora** e **frequência**)*

Categorias CSS

- ***Categorias de valores CSS***
  - ***Número não Negativo com Unidade de Medida***
    - ***Hora***
      - *O **formato** para **definir** um **valor CSS** em **medida** de **hora** é um **número imediatamente** seguido por uma **unidade identificadora** de **tempo** em **segundos***
      - *A **sintaxe prevista** nas **especificações** para **designar** essa **categoria** é **<time>***
      - ***Unidades** de **medida** de **tempo** em **CSS***

Unidade	Descrição
ms	*Milissegundo*
s	*Segundo*

Tabela 6 – Unidades de medida de tempo

Categorias CSS

- ***Categorias de valores CSS***
  - ***Número não Negativo com Unidade de Medida***
    - ***Hora***
      - *O **formato** para **definir** um **valor CSS** em **medida** de **hora** é um **número imediatamente** seguido por uma **unidade identificadora** de **tempo** em **segundos***
      - *A **sintaxe prevista** nas **especificações** para **designar** essa **categoria** é **<time>***
      - ***Unidades** de **medida** de **tempo** em **CSS***

São usados, por exemplo, para definir propriedades destinadas à mídia speech (mídia falada, tal como leitores de tela) ou duração de animações e transições previstas nas CSS3

Categorias CSS

- ***Categorias de valores CSS***
  - ***Número não Negativo com Unidade de Medida***
    - ***Frequência***
      - *O **formato** para **definir** um **valor CSS** em **medida** de **frequência** é um **número imediatamente** seguido por uma **unidade identificadora** de **frequência** em **hertz***
      - *A **sintaxe prevista** nas **especificações** para **designar** essa **categoria** é **<frequency>***
      - ***Unidades** de **medida** de **tempo** em **CSS***

Unidade	Descrição
Hz	*Hertz*
KHz	*Quilohertz*

Tabela 7 – Unidades de medida de frequência

Categorias CSS

- ***Categorias de valores CSS***
  - ***String***
    - ***Valores CSS expressos** com **strings** devem ser **grafados** com **aspas simples** (') ou **duplas** (")*
    - *Sendo da mesma grafia, uma **não** pode **ocorrer dentro** de **outra**, a **menos** que seja **escapada** com uma **barra invertida** (\)*
    - *Um **string não** pode **conter** uma **quebra** de **linha**, a **menos** que se use o **caractere** \A, que **representa** uma **nova linha** em **CSS***
    - *Para fins de **legibilidade**, é **possível quebrar** uma **string** em **substrings** com uso de **caractere barra invertida** (\)*

Categorias CSS

- ***Categorias de valores CSS***
  - ***String***
    - ***Exemplos diversos** da **aplicabilidade** dessas **sintaxes***

```
"Esta é uma 'string'."      /* aspas simples dentro de aspas duplas */
'Esta é uma "string".'      /* aspas duplas dentro de aspas simples */
"Esta é uma \"string\"."    /* aspas duplas escapadas dentro de aspas duplas */
'Esta é uma \'string\'.'    /* aspas simples escapadas dentro de aspas simples */

"Esta string está na primeira linha. \A E esta na segunda"

"Esta é uma string longa\
 que foi quebrada para\
 fins de legibilidade."
```

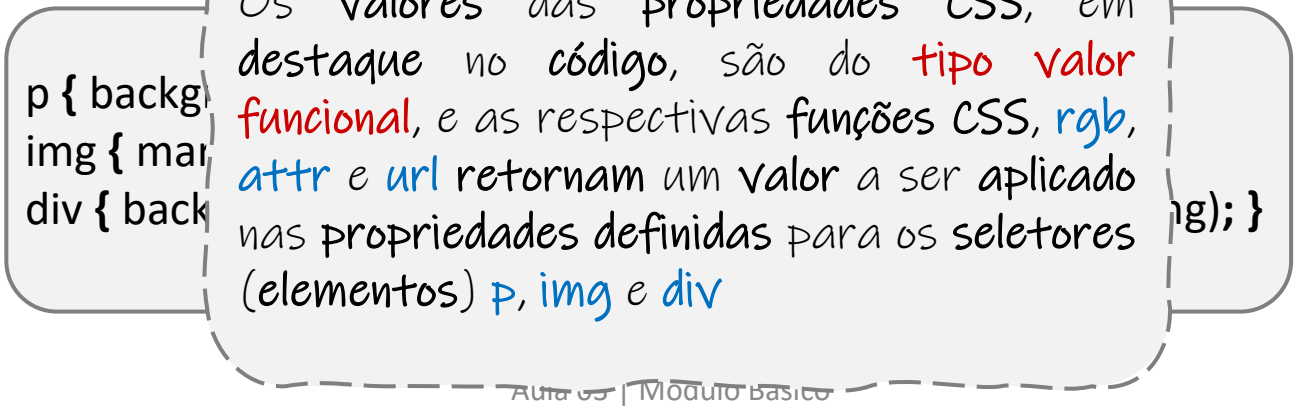
Categorias CSS

- ***Categorias de valores CSS***
  - ***Notação Funcional***
    - ***Valores CSS*** *podem ser* ***expressos*** *por uma* ***função****, e nesses casos são* ***classificados*** *como* ***valores*** *em* ***notação funcional***
    - *Em* ***CSS3****,* ***valores funcionais*** *são usados para* ***definir*** ***cores****,* ***atributos*** *e* ***URI****s (Uniform Resource Identifier)*
    - *A* ***sintaxe*** *para* ***escrita*** *de um* ***valor funcional*** *é: nome da função seguido de uma lista de argumentos entre* ***parênteses***

```
p { background-color: rgb(255, 0, 0); }
img { margin-top: attr(height, px); }
div { background-image: url(https://bkbank.com.br/logo.png); }
```

Categorias CSS

- ***Categorias de valores CSS***
  - ***Notação Funcional***
    - ***Valores CSS*** *podem ser* ***expressos*** *por uma* ***função****, e nesses casos são* ***classificados*** *como* ***valores*** *em* ***notação funcional***
    - *Em* ***CSS3****,* ***valores funcionais*** *são usados para* ***definir*** ***cores****,* ***atributos*** *e* ***URIs*** *(Uniform Resource Identifier)*
    - *A* ***sintaxe*** *para* ***escrita*** *de um* ***valor funcional*** *é: nome da função seguido d* ***teses***



Categorias CSS

- ***Categorias de valores CSS***
  - ***Casos Especiais***
    - ***Valores CSS*** *que* ***não*** *se* ***enquadram*** *em* ***nenhuma*** *das* ***categorias anteriores*** *pertencem a uma* ***categoria denominada*** *"casos especiais"*
    - *Os* ***valores CSS enquadrados*** *nessa* ***categoria*** *são os* ***valores*** *para* ***definição*** *de* ***famílias*** *de* ***fontes*** *e* ***valores*** *para* ***definição*** *de* ***cores*** *em* ***sintaxe hexadecimal***

```
p { background-color: #f00; } ou p { background-color: #ff0000; }

h1 { font-family: Arial, Verdana, Sans-serif; }
```

Cores CSS

- ***Cores***
  - *Os* ***valores possíveis*** *são um* ***valor hexadecimal****, as* ***palavras-chave*** ***transparent****,* ***currentColor*** *e* ***outras*** *palavras-chave que são o* ***nome*** *de* ***algumas cores****,* ***valores RGB****,* ***RGBA****,* ***HSL*** *e* ***HSLA***
  - ***Observe*** *as* ***declarações CSS típicas*** *com* ***uso*** *de* ***cada um desses valores*** *para* ***definição*** *da* ***propriedade color:***

```
color: #ff0000;      /* hexadecimal minúsculas */
color: #FF0000;      /* hexadecimal maiúsculas */
color: #f00;         /* hexadecimal abreviada */
```

Cores CSS

- ***Cores***
  - ***Observe** as **declarações CSS típicas** com **uso** de **cada um desses valores** para **definição** da **propriedade color:***

```
color: pink;                        /* palavra-chave */
color: rgb(255, 200, 32);           /* RGB inteiros */
color: rgb(100%, 26%, 47%);  /* RGB porcentagem */
color: rgba(200, 100, 57, 0.4);    /* RGB inteiros com opacidade */
color: rgba(90%, 86%, 37%, 0.6);   /* RGB procentagem com opacidade */
color: hsl(120, 75%, 50%);          /* HSL */
color: hsla(120, 75%, 50%, 0.8);   /* HSL com opacidade */
color: transparent;    /* palavra-chave */
color: currentColor;   /* criada nas CSS3 */
```

Cores CSS

- ***Cores***
  - ***Hexadecimal** e **RGB***
    - *A **representação** de **uma cor** em **notação hexadecimal** começa com um **sinal** de **tralha** (**#**) seguido de **seis números hexadecimais** (os **números hexadecimais** são **compostos** de **combinações** de: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F)*
    - *A **grafia** dos **números hexadecimais** é **insensível** ao **tamanho** da **caixa** (insensitive case)*

Cores CSS

- ***Cores***
  - ***Hexadecimal** e **RGB***
    - *A **representação** de **uma cor** em **notação hexadecimal** começa com um **sinal** de **tralha** (**#**) seguido de **seis números hexadecimais** (os **números hexadecimais** são **compostos** de **combinações** de: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F)*
    - *A **grafia** dos **números hexadecimais** é **insensível** ao **tamanho** da **caixa** (insensitive case)*

Note que na faixa de 0 a 255 existem 256 números. Dessa formal, o total possível de combinações de números para formação de cores é igual a 256 x 256 x 256 = 16.777.216, o que significa que o uso dessa representação de cores permite que se defina mais de 16 milhões de cores

Cores CSS

- ***Cores***
  - ***Hexadecimal** e **RGB***
    - *As **CSS admitem** que a **declaração** de **cores** com **uso** de **números hexadecimais** seja **abreviada** quando se **tratar** de **cores** nas quais cada um dos **três componentes** seja **representado** por **dígitos iguais**, ou seja, **números hexadecimais** no **formato #XXYYZZ***
    - *Nesses casos, a **abreviatura** se faz **#XYZ**, ficando **subtendido** que cada um dos **três dígitos** da **forma abreviada** é **dobrado***

```
color: #ff9900;     → color: #f90;
color: #55ddaa;     → color: #5DA;
color: #FFEECC;     → color: #fec;
color: #88BB43;     /* não é possível abreviar */
color: #333344;     → color: #334;
```

Cores CSS

- ***Cores***
  - ***Transparent***
    - *O **valor** **transparent** para **cores** foi **criado** pelas **especificações** para as **CSS1** e era **válido apenas** para a **propriedade** **background-color***
    - *As **CSS2** **estenderam** a **validade** desse **valor** para a **propriedade** **boder-color**, e as **CSS3** preveem **validade** do **valor** para **qualquer** **propriedade CSS** que **admita declaração** de **cor***
    - *Como o próprio nome sugere, esse **valor** **destina-se** a **tornar** **transparente** a **propriedade** cuja cor for **declarada** com ele*

Cores CSS

- ***Cores***
  - ***transparent***
    - *As **CSS1** e **CSS2** definiram o **valor inicial** da **cor** da **borda** de um **elemento** como sendo **igual** à **cor** do **próprio elemento***
    - ***Exemplo:***
      - *Se **definirmos** uma **espessura** e um **estilo** para a **borda** de um **elemento h1** e **omitirmos** a **cor** da **borda**, ele **será** a **cor definida** para o **elemento h1***

```
h1 {
    color: red;
    border: 2px solid;
}
```

Cores CSS

- ***Cores***
  - ***Palavra-chave***
    - ***As especificações** para as **CSS3 preveem** **três grupos** de **palavras-chave***
    - ***(1) Palavras-chave** para definir **cores básicas***
      - *Constituído de **16 palavras-chave**, a saber: aqua, black, blue, fuchsia, gray, green, line, maroon, navy, olive, purple, red, silver, teal, white e yellow*
    - ***(2) Palavras-chave** para definir **cores estendidas***
      - *Constituído de **147 palavras-chave** (**incluindo** as **16** para **cores básicas**), como: brown, coral, cian, dimgray, indigo, khaki, mintcream, mistyrose, sienna, turquoise, tomato e whitesmoke*

Cores CSS

- ***Cores***
  - ***Palavra-chave***
    - *(3)* ***Palavras-chave*** *para definir* ***cores*** *de* ***acordo*** *com o* ***SO***
      - *Constituído de* ***28 palavras-chave*** *que se* ***referem*** *a* ***componentes*** *da* ***interface gráfica*** *do* ***SO*** *do* ***usuário*** *(esses valores são de uso prático bastante limitado)*

Cores CSS

- ***Cores***
  - *As **recomendações** de **acessibilidade** ao **conteúdo web** (**WCAG2.0**), em **relação** a **cores:***
    - *1.4.1 **Utilização** da **cor:** a cor não é utilizada como o único meio visual de transmitir informações, indicar uma ação, pedir uma resposta ou distinguir um elemento visual (nível A)*
  - *Dessa forma, é **errado criar** uma **marcação como:***

```
<p> Se você é a favor clique no link verde, mas se é contra clique no link vermelho </p>
```

Cores CSS

- ***Cores***
  - ***currentColor***
    - ***Declara** que a **cor** a ser **aplicada** a uma **determinada propriedade CSS** em **um elemento** é **igual** à **cor declarada** para a **propriedade color** de seu **elemento ancestral mais próximo***
  - ***Exemplo:***
    - ***Considerando elementos p como elementos-filho de div***

```
div { color: red; }
p { border: 1px solid currentColor; }
```

Cores CSS

- ***Cores***
  - ***currentColor***
    - ***Declara** que a **cor** a ser **aplicada** a uma **determinada propriedade CSS** em **um elemento** é **igual** à **cor declarada** para a **propriedade color** de seu **elemento ancestral mais próximo***
  - ***Exemplo:***
    - ***Considerando elementos p como elementos-filho de div***

Todos os parágrafos contidos no div terão uma borda na cor vermelha (red). O uso desse valor para declarar cor não é muito comum, mas dependendo do contexto poderá torna-se bastante útil. Note que, no exemplo, basta alterar a cor declarada para o div para que seja alterada a cor da borda de todos os parágrafos nele contidos.

Cores CSS

- ***Cores***
  - ***HSL***
    - *As **CSS3 criaram** o valor **HSL** para **definir cores**. A **declaração** de **cores** com uso de **HSL** (hue, saturation, lightness), **não existia** nas **CSS2.1**. Ela **permite** que **você declare** as **cores** com **uso** de **três parâmetros:***

      hue = tom; **saturation** = saturação; e **lightness** = luminosidade
  - ***Sintaxe:***

    ```
    color: hsl(120, 75%, 50%)
    ```

    - *O **1º valor** é para o **tom** (**hue**) da **cor**. O seu **valor** é **um número** que **representa** a **medida** de **um ângulo** (0 a 360 graus) apontando para um **tom** de **cor** na **roda** de **cores***

Cores CSS

- ***Cores***
  - ***HSL***
    - ***Valores** do **ângulo** e os **respectivos tons** de **cor***

Ângulo	Tons de Cor
0	*vermelho*
60	*amarelo*
120	*verde*
180	*ciano*
240	*azul*
300	*púrpura*
360	*vermelho*

Cores CSS

- ***Cores***
  - ***HSL***
    - ***Exemplos** de **declaração** de **cor***

```
color: hsl(0, 100%, 50%)      /* cor vermelha */
color: hsl(120, 100%, 50%)    /* cor verde */
color: hsl(120, 100%, 25%)    /* cor verde-escura */
color: hsl(120, 100%, 75%)    /* cor verde-clara */
color: hsl(120, 75%, 75%)     /* cor verde pastel */
```

Cores CSS

- ***Cores***
  - ***HSL***
    - *Note que, o **uso** de **HSL** para **declarar cores** é **semelhante** ao **uso** de **RGB***
    - *A **vantagem** do **uso** do **HSL sobre RGB** é que **HSL proporciona** uma maneira **mais simples** de se **obter variações** de uma **mesma cor***
    - *O **segundo** (**saturation**) e o **terceiro** (**lightness**) **parâmetros** dessa **declaração** são **expressos** em **porcentagens**, e a **sintaxe** para a **declaração determina** que o **sinal** de **porcentagem** é **obrigatório** mesmo se o **valor** for **0** (zero)*

Cores CSS

- ***Cores***
  - ***HSLA***
    - *As **CSS3 criaram** uma **extensão** do **valor HSL** denominada **HSLA** para **definição** de **cores**. Esse **valor acrescenta** um **4º parâmetro** à **declaração** (hue, saturation, lightness, alpha). O **4º parâmetro**, **denominado alpha**, define a **opacidade** (ou transparência) da **cor**.*
  - ***Sintaxe:***

```
color: hsla(120, 75%, 50%, 0.6);
```

    - *Os **três primeiros parâmetros** têm o **mesmo significado** de **quando** se **usa** a **declaração HSL**, e o **4º** é um **valor compreendido entre 0** e **1**. O **valor 0** representa **transparência total**, e o **valor 1**, **opacidade total**. Assim, um **valor igual** a 0.6 **significa** 60% **opaco** ou 40% **transparente**.*

Cores CSS

- ***Cores***
  - ***RGBA***
    - *As **CSS3 criaram** uma **extensão** do **valor RGB** denominada **RGBA** para **definição** de **cores**. Esse **valor acrescenta** um **4º parâmetro** à **declaração** (red, green, blue, alpha). O **4º parâmetro**, **denominado alpha**, define a **opacidade** (ou transparência) da **cor**.*
  - ***Sintaxe:***

```
color: rgba(255, 204, 102, 0.4);
```

    - *Os **três primeiros parâmetros** têm o **mesmo significado** de **quando** se **usa** a **declaração RGB**, e o **4º** é um **valor compreendido entre 0** e **1**. O **valor 0** representa **transparência total**, e o **valor 1**, **opacidade total**. Assim, um **valor igual** a 0.7 **significa** 70% **opaco** ou 30% **transparente**.*

Cores CSS

- ***Cores***
  - ***Cores** do **SO***
    - *As **CSS2 preveem** um **mecanismo capaz** de **definir cores** com **base** no **sistema operacional** do **usuário***
    - ***Esse mecanismo** é **usado** para **estilizar elementos** e **controles utilizados** em **Interfaces** de **Usuário** (**UI** – User Interface) de forma que assumam uma **cor semelhante** às **cores usadas** em **elementos** e **controles** de **interface** do **sistema operacional** do **usuário***
    - *Por exemplo, os **valores ButtonFace** e **ButtonText** **destinam-se** a **simular** as **cores usadas** na **face** e no **texto** dos **botões** da **interface** do **Windows** ou do **Mac**, caso o **visitante** da **página** esteja **usando Windows** ou **Mac***

Cores CSS

- ***Cores***
  - ***Cores** do **SO***
    - *As **CSS2 previram 28 valores** para **simular** as **cores** do **SO***
  - ***Exemplo:***
    - ***Estilizando** a **cor** de **face** de **um botão** com a mesma **cor** de **face** dos **botões** do **SO** do **usuário***

```
button { background-color: ButtonFace; }
```

Esse mecanismo foi colocado em desuso pelas CSS3.

Por questões de retrocompatibilidade, os navegadores deverão continuar oferecendo suporte para ele; contudo, em novos projetos, use a alternativa prevista nas CSS3.

Cores CSS

- ***Cores***
  - ***Cores** do **SO***
    - ***Para substituir** a **definição** de **cores** com **base** no **SO** do **usuário**, que foi **colocado** em **desuso**, as **CSS3 criaram** uma **propriedade** nova denominada **appearance**, que se **destina** a **definir** a **aparência** de **um elemento** com **base** no **SO utilizado** pelo **usuário***
    - *Os **valores possíveis** para essa **propriedade são:** icon, window, desktop, workspace, document, tooltip, status-bar, dialog, message-box, button, caption, small-caption, push-button, hyperlink, radio-button, checkbox, menu-item, tab, menu, menubar, pull-down-menu, pop-up-menu, list-menu, radio-group, checkbox-group, outline-tree, range, field, combo-box, signature, password*

Valor CSS

- ***Valor CSS***
  - ***É importante conhecer** o **conceito** de **valor CSS** para **efeito** de **aplicação** da **regra CSS** no **elemento***
  - ***Exemplo:***

```
p { font-size: 120%; }
```

    - *O **valor 120%** para a **propriedade font-size** usada **nessa regra CSS enquadra-se** no **grupamento** de **valores denominado** "número com unidade de medida"*
    - ***Como** o **navegador aplica** um **tamanho** de **fonte igual** a **120%? Qual** o **valor em pixels? 120% do quê?***

Valor CSS

- ***Valor CSS***
  - ***Para aplicar valores CSS**, o **navegador precisa**, em **certos casos**, **efetuar cálculos** e, em outros, "**tirar algumas conclusões**" para **chegar** ao **valor** a **aplicar***
  - *Ao **longo** do **processo** de **investigação**, o **navegador** passa por **etapas**, e em **cada etapa chega** a um **tipo** de **valor***
  - ***Observe** as **regras CSS***

```
p { font-family: Arial, Sans-serif; }      /* estiliza p com fonte na família
                                              especificada (valor) */
p { width: 400px; }                        /* estiliza p com largura 400px */
p {font-size : 120%; }                     /* estiliza p com tamanho de fonte
                                             1.2 vezes o valor de referência */
```

Valor CSS

- ***Valor CSS***
  - ***Observe** as **regras CSS***

```
p { background-color: red; }   /* estiliza p com fundo na cor vermelha */
p { height: 2em; }             /* estiliza p com altura 2 vezes o valor de
                                  referência */
```

  - ***Observe** que **alguns valores** são **absolutos** e **outros relativos**, tais como as **medidas CSS** em **porcentagem** e **em***
  - *Para **aplicar valores CSS**, os **mecanismos** das **CSS consideram cinco tipos** de **valores***
  - *A **todas** as **propriedades CSS** é **atribuído**, por padrão, **um valor** denominado **valor inicial***

Valor CSS

- ***Valor CSS***
  - ***Observe** as **regras CSS***
    - *O **valor inicial** de **cada** uma das **propriedades CSS** é **definido** por uma **folha** de **estilo nativa** do **agente** do **usuário** (**browser**)*
    - ***Infelizmente**, **não há padronização** para o **valor inicial** das **propriedades CSS**, e **cada navegador implementa** essa **funcionalidade** à sua maneira*
    - *Esse **comportamento** pode **trazer inconsistência** de **renderização** em **diferentes navegadores***

Valor CSS

- ***Valor CSS***
  - ***Felizmente**, **todos** os **navegadores adotam** o **mesmo valor** para **muitas** das **propriedades CSS***
  - *As **inconsistências**, em sua **maioria**, estão **relacionadas** à **definição** de **valores iniciais** para **margin** e **padding***
  - ***Exemplo** (valor inicial de algumas propriedades CSS)*

```
border: none;
color: black;
background: transparent;
font-family: serif;
font-size: 16px;
```

Valor CSS

- ***Valor CSS***
  - ***Todas** as **propriedades CSS admitem**, como **valor**, a **palavra-chave initial** para **forçar** a **adoção** do **valor inicial** da **propriedade***
  - ***Exemplo:***

```
<div>
  <p>Parágrafo com <em>palavra</em> marcada com ênfase. </p>
</div>
```

HTML

```
div {
      color: red;
      border: 1px  solid  blue;
      }
p { color: initial; }
```

CSS

Valor CSS

- ***Valor CSS***
  - ***É comum encontrarmos*** *em* ***fóruns*** *e* ***matérias publicadas*** *em* ***blogs*** *a* ***indicação*** *do uso de* ***uma regra CSS*** *para* ***zerar*** *os* ***valores*** *de* ***margin*** *e* ***padding*** *de* ***todos*** *os* ***elementos*** *da* ***marcação*** *com* ***uso*** *do* ***seletor universal***
  - ***Exemplo:***

```
* {
    margin: 0;
    padding: 0;
}
```

Vinculando CSS

- ***Vinculando Folhas** de **Estilo***
  - *Ao **escrever** sua **folha** de **estilo** é preciso **informar** ao **documento** onde ele **deve buscá-la**. Ou seja, **você precisa** de **um método capaz** de **vincular** a **folha** de **estilo** ao **documento** a qual ela **será aplicada***
  - *As **folhas** de **estilos** podem ser **escritas** no **próprio documento HTML** ao qual **serão aplicadas** ou ser **arquivos externos independentes**, **gravados** com a **extensão .css**, por exemplo, um **arquivo chamado** de main.css, e **lincados** ao **documento***

Vinculando CSS

- ***Vinculando Folhas** de **Estilo***
  - ***Estilos inline***
    - *O **método** **direto** e **simples** de **aplicar estilos** a um **elemento** da **marcação** é com o **emprego** do **atributo** **style** da **HTML***
    - ***Permite escrever** as **regras** de **estilo diretamente** dentro da **tag** de **abertura** do **elemento** a **estilizar***

```
<p style="width: 200px; color: white; background: red; font-size: 1.8em;">
        <!– Parágrafo com aplicação de estilos inline -->
</p>
```

Vinculando CSS

- ***Vinculando Folhas** de **Estilo***
  - ***Estilos inline***
    - *O **método** **direto** e **simples** de **aplicar estilos** a um **elemento** da **marcação** é com o **emprego** do **atributo** **style** da **HTML***
    - ***Permite escrever** as **regras** de **estilo diretamente** dentro da **tag** de **abertura** do **elemento** a **estilizar***

Esse método dificulta a manutenção e retira um dos maiores poderes da folha de estilo, que é o controle centralizado da apresentação.
Toda vez que for preciso alterar a apresentação, será necessário percorrer todo o código de marcação do documento ou centenas de documentos, se o site for grande, à procura das regras de estilo inline.

Vinculando CSS

- ***Vinculando Folhas** de **Estilo***
  - ***Estilos incorporados***
    - ***Outro método** de **escrever** a **folha** de **estilos** no **próprio documento HTML** é com o **emprego** do **elemento style***
    - ***Permite escrever** as **regras** de **estilo dentro** das **tags** <style></style>, **declaradas** na **seção HEAD** do **documento***

```
<head>
...
<style rel="stylesheet" type="text/css" media="all">
body {
  margin: 0;
  padding: 0;
  font-size: 80%;
  color: black;
  background: white;
}
</style>
</head>
```

Vinculando CSS

- ***Vinculando Folhas** de **Estilo***
  - ***Estilos incorporados***
    - *O **elemento** **style** deve **estar contido** na **seção** **HEAD** do **documento**, todavia, em **marcação HTML5** o uso de **atributos** no **elemento** **style** é **facultativo***
    - *O **atributo** **type** **informa** qual **tipo** de **dado** está sendo **enviado**, e o **atributo** **media** **informa** a qual **tipo** de **mídia** devem ser **aplicados** os **estilos***

Vinculando CSS

- ***Vinculando Folhas** de **Estilo***
  - ***Estilos incorporados***
    - *Os **valores** do **atributo** **media** e a **mídia** a que se **destinam** são **elencados** na **tabela**:*

Valor	Mídia
screen	*Telas de monitores*
tty	*Teletipo e similares*
tv	*Dispositivos tipo televisão*
projection	*Projetores*
handheld	*Dispositivos portáveis*
print	*Impressoras e visualização no modo impressão*

Vinculando CSS

- ***Vinculando Folhas** de **Estilo***
  - ***Estilos incorporados***
    - *Os **valores** do **atributo** **media** e a **mídia** a que se **destinam** são **elencados** na **tabela**:*

Valor	Mídia	Nota
braille	*Dispositivos táteis*	
aural	*Sintetizadores de voz*	*em desuso pela CSS3*
all	*Todos os tipos de mídia*	
speech	*Sintetizadores de voz*	*criada pela CSS3*
embossed	*Impressoras braile*	*criada pela CSS3*

Vinculando CSS

- ***Vinculando Folhas** de **Estilo***
  - ***Estilos externas***
    - ***É aquela** que **não** foi **escrita** no **documento HTML***
    - ***Trata-se** de um **arquivo** de **texto contendo** as **regras** de **estilo** e os **comentários CSS***
    - *Um **arquivo** de **folha** de **estilos** deve ser **gravado** com a **extensão .css** e **pode** ser **vinculado** a **um documento HTML** de **duas maneiras distintas** (**lincadas** e **importadas**)*

Vinculando CSS

- ***Vinculando Folhas** de **Estilo***
  - ***Folhas** de **Estilo Lincadas***
    - ***Vinculamos** uma **folha** de **estilo externa** a um **documento empregando** o **elemento link***
    - ***Esse elemento** deve **estar contido** na **seção HEAD** do **documento** e tem por **finalidade associar outros documentos** ao **documento** no qual ele está contido*

```
<head>
  ...
  <link rel="stylesheet" type="text/css" href="estilos.css" media="all">
  ...
</head>
```

Vinculando CSS

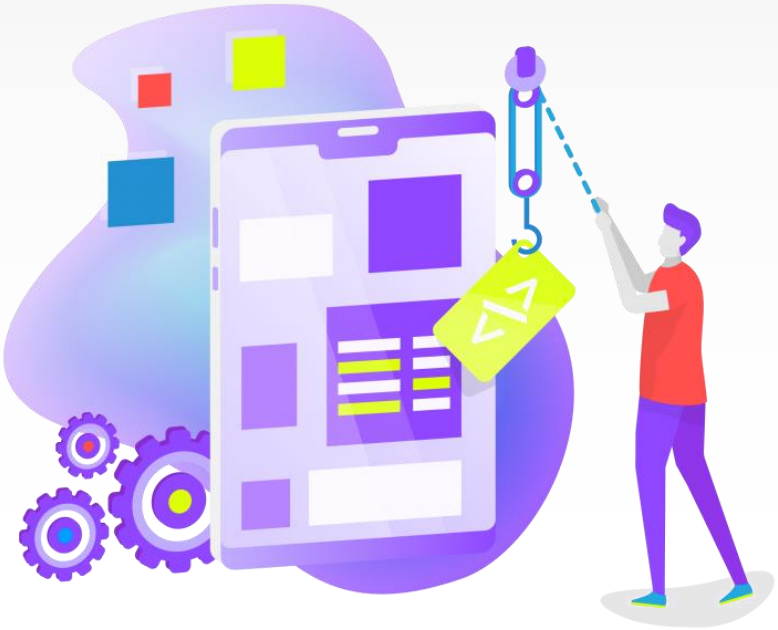
- ***Vinculando Folhas** de **Estilo***
  - ***Folhas** de **Estilo Importadas***
    - ***Esse método** permite **vincular folha** de **estilo externa** a outra **folha** de **estilo externa** usando a **diretiva @import** dentro da **folha** de **estilo***
    - ***É possível importar mais** de **uma folha** de **estilo** para **dentro** de **uma folha** de **estilo***

```
@import "main.css"
body {
  margin: 0;
  font: 62.5% Arial, Sans-serif;
}
... mais regras de estilo ...
```

Vinculando CSS

- ***Vinculando Folhas** de **Estilo***
  - ***Folhas** de **Estilo Importadas***
    - *A **diretiva** **@charset** destina-se a **declarar** a **codificação** de **caracteres** de uma **folha** de **estilo** e deve **ocupar** a **primeira linha** na **folha** de **estilo***

```
@charset "utf-8"
@import "main.css"
body {
  margin: 0;
  font: 62.5% Arial, Sans-serif;
}
... mais regras de estilo ...
```



EXERCÍCIOS

Referências

Silva, M. S. Fundamentos de HTML5 e CSS3. 1. ed. São Paulo: Novatec, 2015.

Duckett, J. HTML e CSS Projete e Construa Websites. 1. ed. Rio de Janeiro: Alta Books, 2011.

Hyslop, B., Castro, E. HTML and CSS: Visual Quickstart Guide. 8. ed. Barcelona: Peachpit Press, 2013.