



**UNIVERSIDADE EDUARDO MONDLANE**  
**FACULDADE DE ENGENHARIA**  
**DEPARTAMENTO DE ENGENHARIA ELECTROTÉCNICA**

# **Programação Web e SGC**

Introdução ao CSS

**Docentes:** Ruben Manhiça

**Maputo, 1 de abril de 2024**



# **Conteúdo da Aula**

1. Web Standards;
2. Introdução ao CSS





# Web Standards - Definição

Web Standards é um conjunto de normas, diretrizes, recomendações, notas, artigos, tutoriais e afins de caráter técnico, produzidos pelo **W3C** e destinados a orientar fabricantes, desenvolvedores e projetistas para o uso de práticas que possibilitem a criação de uma Web acessível a todos, independentemente dos dispositivos usados ou de suas necessidades especiais.





# Por que usar padrões?

- manutenção mais fácil
- tamanho do arquivo menor (economia de banda e velocidade no carregamento)
- maior acessibilidade para o usuário
- compatibilidade com diferentes navegadores





# CSS – Cascading Style Sheets

**Cascading Style Sheets (CSS)** é uma linguagem de definição de estilos usada para descrever a **apresentação/formatação** de um documento escrito em uma linguagem de marcação. Sua aplicação mais comum é definir estilos de paginas web escritas em **HTML** e **XHTML**.





# CSS – Cascading Style Sheets

O padrão do CSS é mantido pelo World Wide Web Consortium (W3C).

**<http://www.w3.org/Style/CSS/>**





# CSS – Cascading Style Sheets

## Uso principal:

CSS é utilizada por desenvolvedores web para definir:

- cores
- fontes
- Hyperlinks
- tabelas
- parágrafos
- espaçamento
- layout tableless





# CSS – Cascading Style Sheets

- **Idéia principal:** Separar o conteúdo do documento (escrito em HTML) da apresentação do documento (escrita em CSS).
- **Como isto é feito:** Através de um conjunto de regras (estilos) que são aplicadas aos elementos das páginas web (tags)
- **Regra de ouro do design CSS:**  
“Tudo que deve sobrar quando a folha de estilo é removida, é o texto que prove a informação e os elementos de HTML que descrevem a estrutura do documento”.







# CSS – Cascading Style Sheets

- É um padrão de formatação;
- Com CSS consegue-se separar o estilo do conteúdo;
- Propriedades de CSS podem ser aplicadas a todas tags, modificando seus atributos.





# CSS - Vantagens

- Diferentes e sofisticados estilos podem ser aplicados ao mesmo documento;
- Criação e manutenção do design ficam mais fáceis;
- Prevalece a consistência do site, o CSS pode garantir que todos os documentos tenham o mesmo desenho e arranjo;





# CSS - Vantagens

- Permite ao Designer um controle maior sobre:
  - Tamanho e cor da fonte;
  - Espaçamento entre linhas e caracteres;
  - Margem do texto, listas, cores e fundos;
  - Maior versatilidade no desenvolvimento de designer sem aumentar o tamanho do site;
  - Introduziu a utilização de Layers, permitindo a sobreposição de texto sobre texto ou imagem;





# CSS – Por que usar?

A separação de apresentação e estrutura do documento provê:

- maior facilidade para manipulação do conteúdo, pois este está separado da formatação
- opções mais sofisticadas de formatação
- Facilidade de manutenção através do compartilhamento de estilos entre paginas (evita repetição)
- conteúdo pode ser aproveitado com folhas de estilos diferentes para meios diferentes (impressão, web, celulares, voz)
- Permite a criação de temas ou skins, cada utilizador pode escolher um look and feel diferente para o site

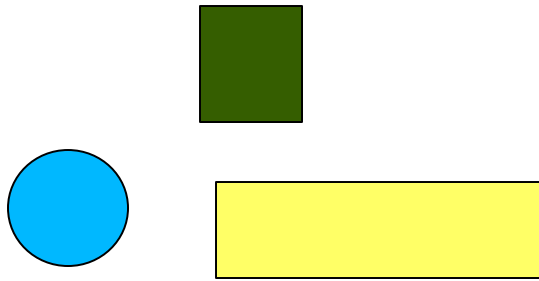




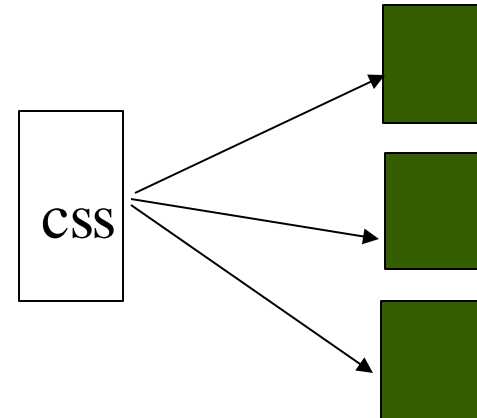
# CSS – Por que usar?

- **Consistência**

Consistência é muito importante para a navegabilidade no website. A forma mais simples de manter consistência em um conjunto de paginas web é usando Cascading Style Sheets (CSS).



paginas sem CSS tem dificuldade de padronização e manutenção

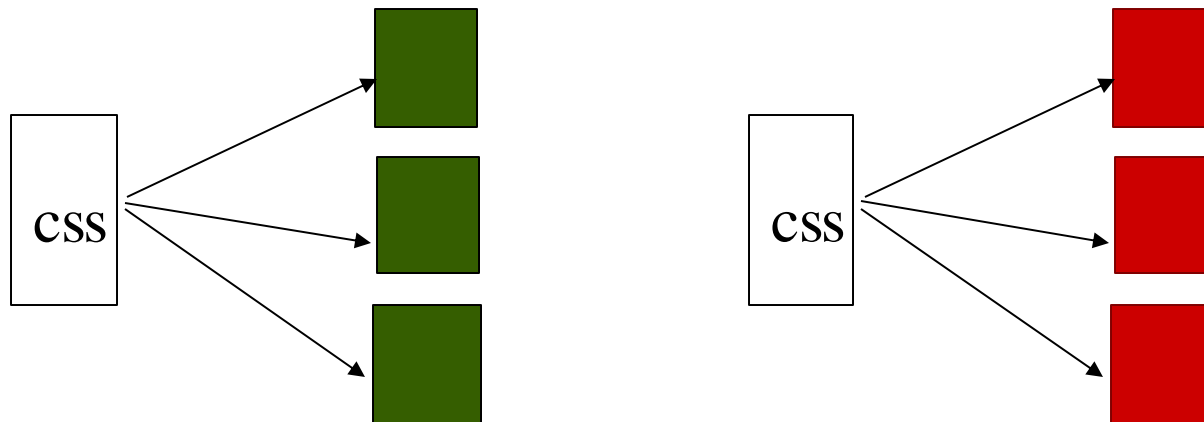




# CSS – Por que usar?

- **Manutenção**

A forma mais simples de dar manutenção e realizar alterações no estilo de um website é usando CSS





# Lembrem-se Sempre!!!





# CSS – Interna X Externa

Existem duas maneiras de se utilizar CSS

**CSS interno** – Definida no mesmo arquivo que o documento html. O CSS interno pode ser declarado:

**Dentro da tag <body>, definido como atributos das tags html normais.** Escopo: estilo vale somente para um elemento da pagina

**Dentro da tag <head> atraves de uma tag <style>.**

- **Escopo:** Estilo vale para todas as tags de um dado tipo, dentro de um único documento html

**CSS externo** – Definido em um arquivo .css separado do documento html.

- **Escopo:** Pode ser compartilhada por múltiplos documentos html.







# CSS Interna usando atributos das tags html

```
<html>
```

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

```
<body>
```

```
<P style="color: green; font-weight: bold; text-align: center"> O sapo nao lava o pe </P>
```

```
<A style="font-size: 18px; color:red"
HREF="http://www.sapo2.pt/"> Veja por que</A>
```

```
</body>
```

```
<html>
```





# Estrutura de uma regra CSS

```
seletor { propriedade: valor; }
```

Ex:

```
h1{  
font-size:16pt;  
font-family: verdana;  
}
```

**Seletor:** é o nome do estilo. Duas normas:

- nome de um elemento html **h1** (aplicável a todos h1).
- uma variável como “**meuestilo**” associado manualmente as tags através do atributo **class**

**propriedade : valor;** Especifica atributos de formatação e seus valores separados por : Listas de elementos são separadas por ;





# CSS – Interna para head Exemplo

```
<html>
<head>
<style type="text/css">
...
</style>
</head>
```

elementos afetados pela folha de estilo

**<body>**

**<H1>** Trabalho de html **</H1>**

**<P>**A sigla HTML deriva da expressão inglesa HyperText Markup Language. Trata-se de uma linguagem de marcação utilizada para produzir paginas na Internet. Esses códigos podem ser interpretados pelos browsers para exibir as paginas da World Wide Web.

**</P>**

**</body>**

**</html>**





# CSS Externa - teste.css

```
@charset "utf-8";
```

```
.body {  
font-family: "Arial", sans-serif;  
margin-left: 15%;  
margin-right: 15%; }
```

```
.titulo1 {  
color: red;  
font-family: "Arial";  
font-size: 25px; }
```

```
.texto{  
font-family: "Times";  
font-size: 16px; }
```

```
.codigo {  
font-family: monospace;  
font-size: 12px; }
```





# CSS – Externa Exemplo

```
<html>
<head>
<link type="text/css" rel="stylesheet" href="teste.css">
</head>

<body class="body">

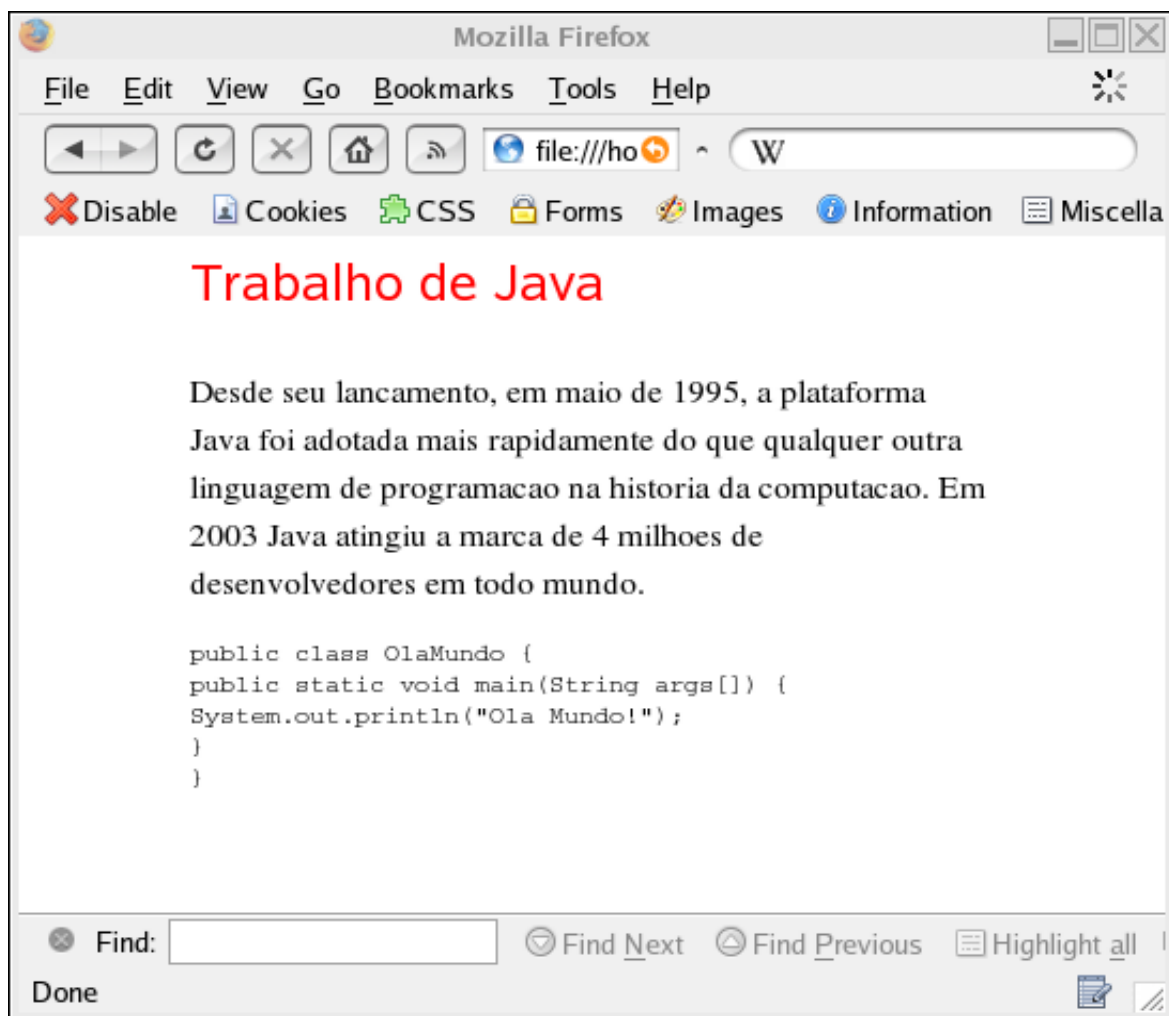
<P class="titulo1">Trabalho de  Java </P>

<P class="texto">Desde seu lancamento, em maio de 1995, a plataforma
Java foi adotada mais rapidamente do que qualquer outra linguagem de
programacao na historia da computacao. Em 2003 Java atingiu a marca
de 4 milhoes de desenvolvedores em todo mundo. </P>
<P class="codigo">
public class OlaMundo { <BR>
    public static void main (String args[]); <BR>
        System.out.println("Ola Mundo"); }<BR>
}; <BR>
</P>
</body>
</html>
```





# CSS – Externa Exemplo





# CSS – Externa Exemplo

## Reutilizando a folha de estilos

```
<html>
<head>
<link type="text/css" rel="stylesheet" href="teste.css"></head>

<body class="body">

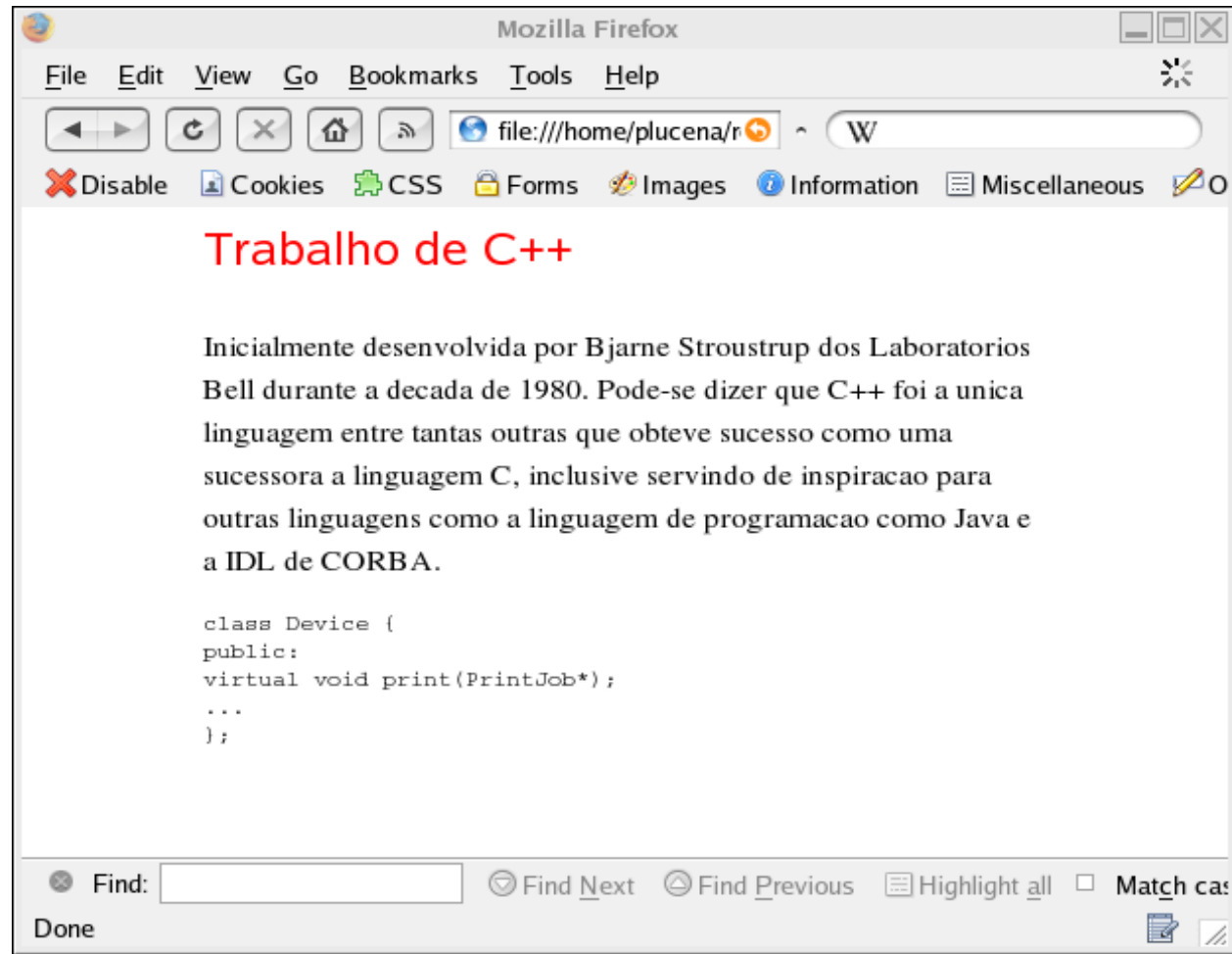
<P class="titulo1">Trabalho de C++</P>

<P class="texto"> Inicialmente desenvolvida por Bjarne Stroustrup dos
Laboratórios Bell durante a década de 1980. Pode-se dizer que C++ foi a
única linguagem entre tantas outras que obteve sucesso como uma sucessora
à linguagem C, inclusive servindo de inspiração para outras linguagens
como a linguagem de programação Java e a IDL de CORBA.</P>

<P class="codigo">
class Device { <BR>
    public: <BR>
        virtual void print(PrintJob*); <BR>
        ... <BR>
    };<BR>
</P>
</body>
</html>
```



# CSS – Externa Exemplo







# HTML – Folhas de Estilo

- Sintaxe das folhas de estilo

- Elemento – tag de html sem os sinais < e >. Geralmente é chamado de seletor;
- Atributo – Atributo CSS usado como estilo;
- Valor – Valor atribuído ao estilo;

- Formato 1

```
elemento { atributo: valor }
```

Ex:

```
p { font-family: Arial } <!-- todas as ocorrências de <p> terão a fonte  
    Arial-->
```

- Formato 2

```
elemento1 [, elemento2, ..., elementon] { atributo: valor }
```

Ex:

```
h1, h2, h4 { color: red } <!-- todas as ocorrências das tags <h1>, <h2> e  
    <h4> serão vermelhas -->
```





# HTML – Folhas de Estilo

- Formato 3

```
elemento { atributo1: valor1; atributo2: valor2;...}
```

Ex.:

```
body { background-color: blue; font-family: Arial }
```

```
<!-- definição de vários atributos para um elemento -->
```

- Formato 4

```
elemento1 elemento2 ... Elementon { atributo: valor }
```

Ex.: `p i { color: yellow }`

Todas as ocorrências marcadas com as tags `<i>` e `<p>` serão amarelas.





# HTML – Folhas de Estilo

- Plano de fundo

- background-color – especifica a cor de fundo do elemento

- Valores possíveis - <cor> | transparent

Ex.: `table { background-color: #cccc99 }`

- background-image – define a imagem de fundo

- Valores possíveis – <url> | none

Ex.: `body { background-image: url(foto.jpg) }`

- background-attachment – define se a imagem de fundo rola com o conteúdo ou não

- Valores possíveis – scroll | fixed

Ex.: `body { background-image: url(foto.jpg); background-attachment: fixed }`





# HTML – Folhas de Estilo

- Plano de fundo

- background-repeat – especifica se a imagem de fundo será repetida e como será a repetição
- Valores possíveis – repeat-x | repeat-y | repeat | no-repeat

Ex.: `body { background-image: url(foto.jpg); background-repeat: no-repeat }`

- background-position – define a posição da imagem de fundo
- Valores possíveis – distância para o topo (percentual ou pixels) e/ou distância para a margem esquerda (percentual ou pixels) | [top|center|bottom] e/ou [left|center|right]

Ex.: `body { background-image: url(foto.jpg) background-position: center center }`





# HTML – Folhas de Estilo

- Plano de fundo
  - background – especifica vários atributos de uma só vez.
  - Valores possíveis – os mesmos para os atributos background-color, background-image, background-repeat, background-attachment, background-position

```
Ex.: body { background: fixed url(foto.jpg) no-repeat center center }
```

- Cores
  - color – especifica a cor do texto de um elemento
  - Valores possíveis - <cor>

```
Ex.: p { color: #ffaabb }
```





# HTML – Folhas de Estilo

- Propriedades de fontes

- font-family – especifica lista de fontes
- Valores – <nomedafamília>|<famíliagenérica>

Ex.: a { font-family: "Times Roman", serif }

Obs.: as famílias genéricas podem ser: serif, sans-serif, cursive, fantasy, monospace

- font-size – especifica o tamanho da fonte
- Valores – <tam-abs>|<tam-relat>|<tamanho>|<percent>
  - tam-abs: xx-small, x-small, small, medium, large, x-large, xx-large
  - tam-relat: larger, smaller
  - tamanho: valor em pontos
  - percent: valor percentual em relação ao padrão

Ex.: body { font-size: 12pt }





# HTML – Folhas de Estilo

- Propriedades de fontes
  - font-stretch – especifica largura (expansão ou condensação) da fonte
  - Valores – normal | wider | narrower | ultracondensed | extra-condensed | condensed | semi-condensed | semi-expanded | expanded | extra-expanded | ultra-expanded

Ex.: a { font-stretch: condensed }

- font-style – especifica o estilo da fonte
- Valores – normal | italic | oblique

Ex.: p { font-style: italic }





# HTML – Folhas de Estilo

- Propriedades de fontes
  - font-variant – especifica se a fonte será exibida em caixa alta ou normal
  - Valores – normal | small-caps

Ex.: h1 { font-variant: small-caps }

- font-weight – especifica a espessura da fonte
- Valores – normal | bold | bolder | lighter | 100-900

Ex.: p { font-weight: bold }  
span { font-weight: 800 }

- font – especifica vários atributos de uma só vez

Ex.: td { font: italic bold 12pt Arial, Courier }







# HTML – Folhas de Estilo

- Propriedades de textos

- letter-spacing – define a distância entre as letras
- Valores – normal | <valor em px ou pt>

Ex.: th { letter-spacing: 12pt }

- line-height – espaçamento entre linhas de texto
- Valores – normal|<valor em px ou pt>|<percentual>

span { line-height: 14pt }

- text-align – define o alinhamento do texto
- Valores – left | right | center | justify

Ex.: td { text-align: justify }





# HTML – Folhas de Estilo

- Propriedades de textos

- text-decoration – define efeitos adicionais ao texto
- Valores – none|underline|overline|line-through|blink

Ex.: `h3 { text-decoration: underline }`

- text-indent – recuo da primeira linha de texto em um bloco
- Valores – <valor> | <percentual>

`span { text-indent: 30px }`

- text-transform – controla a capitalização do texto
- Valores – none | capitalize | uppercase | lowercase

Ex.: `td { text-transform: uppercase }`





# HTML – Folhas de Estilo

- Propriedades de textos

- vertical-align – alinhamento em uma célula

- Valores – top | middle | bottom

- Ex.: `th { vertical-align: middle }`

- white-space – manipulação dos espaços em branco

- Valores – normal | pre | nowrap

- pre – considera espaços adicionais

- nowrap – não quebra a linha nos espaços em branco

- `p { white-space: nowrap }`

- word-spacing – espaçamento entre palavras

- Valores – normal | <valor em px ou pt>

- Ex.: `h1 { word-spacing: 12pt }`





# HTML – Folhas de Estilo

- Caixas

- border – define largura, estilo e cor para as quatro bordas
- Valores – <valor em px|thin|medium|thick> | <estilo> | <cor>

Ex.: `table { border: 1px solid red }`

Opções para estilo:

- none – sem borda
- hidden – oculta
- dotted – pontilhada
- dashed – tracejada
- solid – linha sólida
- double – linha dupla
- groove - entalhada





# HTML – Folhas de Estilo

- Caixas
  - border-bottom, border-left, border-right, border-top – define largura, estilo e cor para as bordas individualmente
  - mesmas opções de border.
  - border-color – define as cores das quatro bordas.
  - border-bottom-color, border-left-color, border-right-color, border-top-color – define a cor das bordas individualmente
  - border-style – define o estilo para as quatro bordas
  - border-bottom-style, border-left-style, border-right-style, border-top-style – define o estilo das bordas individualmente





# HTML – Folhas de Estilo

- Caixas
  - border-width – define as larguras das quatro bordas.
  - border-bottom-width, border-left-width, border-right-width, border-top-width – define as larguras das quatro individualmente
  - padding – define as margens ao redor (externo) do elemento
  - padding: acima direita abaixo esquerda

Ex.: `img { padding: 10px 20px 15px 12px }`

- padding-bottom, padding-left, padding-right, padding-top – define as margens individualmente





# HTML – Folhas de Estilo

- Caixas

- padding – define as margens internas do elemento
- padding: acima direita abaixo esquerda

Ex.: `th {padding: 10px 20px 15px 12px}`

- padding-bottom, padding-left, padding-right, padding-top –  
define as margens internas individualmente





# HTML – Folhas de Estilo

- Dimensões
  - height – define a altura do elemento
  - height: auto|<valor em px>|<valor percentual>

Ex.: `table { height: 100px }`

- width – define a largura do elemento
- width: auto|<valor em px>|<valor percentual>

Ex.: `img { width: 100px }`

Obs.: A imagem no IE não distorce ao se definir somente uma das dimensões.







# HTML – Folhas de Estilo

- Posicionamento

- position – define o modo de posicionamento do elemento
- position: absolute|relative|static

Ex.: `table { position: relative }`

- left – define a posição do elemento a partir da margem esquerda
- left: auto|<valor em px>|<valor percentual>

Ex.: `div { position: absolute; left: 10px }`

- right – define a posição do elemento a partir da margem direita
- right: auto|<valor em px>|<valor percentual>

Ex.: `h1 { position: absolute; right: 10px }`





# HTML – Folhas de Estilo

- Posicionamento

- top – define a posição do elemento a partir do topo

- top: auto|<valor em px>|<valor percentual>

Ex.: h2 { position: absolute; top: 20px }

- bottom – define a posição do elemento a partir da base

- bottom: auto|<valor em px>|<valor percentual>

Ex.: p { position: absolute; bottom: 5px }

- visibility – define se o elemento está visível ou não

- visibility: inherit | visible | hidden

Ex.: div { visibility: hidden }





# HTML – Folhas de Estilo

- Listas
  - list-style-type – define o tipo de marcador usado na lista
  - list-style-type: <tipo>
  - Tipos possíveis:
    - disc - disco
    - circle - círculo
    - square - quadrado
    - decimal – número inteiro
    - lower-roman – romano minúsculo
    - upper-roman – romano maiúsculo
    - lower-alpha – letra minúscula
    - upper-alpha – letra maiúscula
    - none - nenhum

Ex.: `ul { list-style-type: decimal }`





# HTML – Folhas de Estilo

- Listas

- list-style-position – define a posição do marcador

- list-style-type: inside | outside

Ex.:ul { list-style-type: decimal; list-style-position: inside }

- list-style-image – define a imagem que será usada como marcador na lista

- list-style-image: url(<caminho do arquivo>)

Ex.:ul {list-style-image: url("../imagens\lista.gif") }

- list-style – define tipo, imagem e posicionamento de uma única vez

Ex.:ul {list-style: url("../imagens\lista.gif") inside }

Obs.: Não faz sentido definir o tipo da lista e uma imagem como marcador. A imagem vai se sobrepor ao tipo definido.





# HTML – Folhas de Estilo

- **Classes**

- Definidas para dar maior flexibilidade às folhas de estilo;
- Podem ser criadas para um elemento específico ou para qualquer um;
- Aplicadas com o atributo class;
- Sintaxe:

`<nomedatag class="nomedaclasses">`

Ex.:

`<p class="txtPequeno">Site desenvolvido por ...</p>`





# HTML – Folhas de Estilo

- **Classes**

- Definição para tag específica – somente as tags definidas e com o atributo class indicando o nome da classe serão formatadas.

- `elemento.nomedaclasse { atributo: valor }`

- Ex.:

```
p.verde { color: #00BB00 }
```

- Definição geral – o estilo não é definido para uma tag específica

- `.nomedaclasse { atributo: valor }`

```
Ex.: .azul { color: #0000BB }
```





# TPC

- Estudar e aprofundar os conceitos discutidos durante a aula;



**FIM!!!**

Duvidas e Questões?

