



Desenvolvimento de páginas

Parte 5 – Recursos avançados de HTML e CSS

Informática – CEFET

Prof. Dacy Câmara Lobosco

Prof. Diogo Silveira Mendonça

Prof. Thiago Delgado Pinto

Prof. Rafael Escalfoni



CSS



CSS

- A separação da formatação permite:
 - Melhorar o conteúdo de acessibilidade (para portadores de deficiência);
 - Prover mais flexibilidade e controle sobre a apresentação do documento;
 - Permitir o compartilhamento de formatação entre diversas páginas;
 - Reduzir a complexidade e a repetição;
 - Configurar a apresentação da página para diferentes formas (tela, impressão, narração por software, etc.);



CSS avançado

- Estilizando elementos
- Estilizando por ID
- Estilizando por Classe



Seletores

- Seletores são usados para declarar quais elementos de marcação o estilo se aplica.
- O estilo poderá ser aplicado:
 - A todos os elementos de um certo tipo;
 - A elementos com certo atributo;
 - A elementos com certa encadeamento/aninhamento dentro das marcações;



Pseudo-classes e pseudo-elementos

- Pseudo-classes e pseudo-elementos são usados para identificar elementos de marcação e ações específicas do usuário ao qual um bloco de declaração se aplica.
- **Pseudo-classes** selecionam elementos inteiros
- **Pseudo-elementos** selecionam elementos parciais



O que usar, ID ou Classe?

- ID – é mais rápido
 - Atribui características visuais a um único elemento
- Classe facilita o reúso de código e flexibilidade
 - **É possível atribuir mais de uma classe a um elemento**



Cascata

A prioridade para o efeito cascata em ordem crescente:

1. folha de estilo padrão do navegador do usuário;
2. folha de estilo do usuário;
3. folha de estilo do desenvolvedor;
 - a. estilo externo (importado ou ligado).
 - b. estilo incorporado (definido na seção head do documento);
 - c. estilo *inline* (dentro de um elemento HTML);
4. declarações do desenvolvedor com *!important*;
5. declarações do usuário com *!important*;



Cascata e Herança no CSS

```
<div id="pai">
  <h1>Sou um título</h1>
  <h2>Sou um subtítulo</h2>
</div>
```

```
#pai {
  color: blue;
}
```

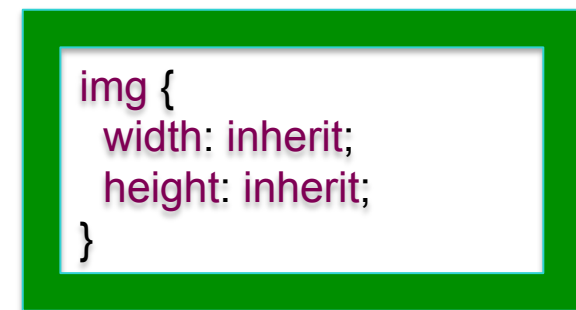
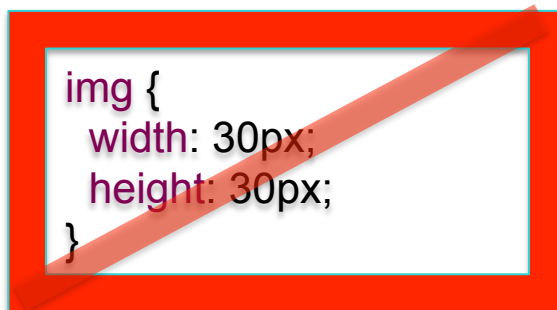
- As tags filhas herdam a cor da tag pai
- As propriedades que afetam a caixa do elemento (tamanho, margin e padding) **não são aplicadas em cascata em elementos filhos**
 - **E se quisermos aplicar???**

Exemplo

- Fazer uma imagem ocupar todo o espaço destinado a ela:

```
<div>  
    
</div>
```

```
div {  
  border: 2px solid;  
  border-color: red;  
  width: 30px; height: 30px;  
}
```





CSS Reset

- Perceba que às vezes, o CSS do navegador pode interferir no visual planejado.
 - **CSS Reset**, uma folha de estilo que atribui um **valor básico** para **todas as características** do CSS, sobrescrevendo **totalmente os estilos padrão do navegador**
- *Aguardem...*



Display – Block vs Inline

- Elementos HTML, podem ser renderizados como blocos (block) ou em linha (inline)
 - Por padrão, são elementos em bloco:
 - `<h1>` a `<h6>`, `<p>`, `<div>`, ``...
 - Por padrão, são elementos em linha:
 - `<a>`, `<small>`, ``, ``, ``...



Position – Static, Relative, Absolute

- Há propriedades para posicionamento de elementos – `top`, `left`, `bottom` e `right` que não são obedecidas por padrão, pois dependem de outra – `position`
- `Position` determina qual é o modo de posicionamento de um elemento
 - `static` – padrão; permanece no local original
 - `relative` – posiciona nas coordenadas em relação à posição original do elemento (relativa à origem)
 - `absolute` – posiciona nas coords em relação a um ancestral diferente de `static` ou porção visível da página
 - `fixed` – elemento renderizado na posição indicada



Posicionamento - float e clear

- float: faz o elemento “fluir”
- clear: determina qual será o comportamento dos elementos subsequentes
 - Limpa o float ao lado – `left`, `right` ou `both`

Exemplo – sem clear

Cabecalho

Menu lateral

- Item 1
- Item 2
- Item 3
- Item 4

Conteúdo principal

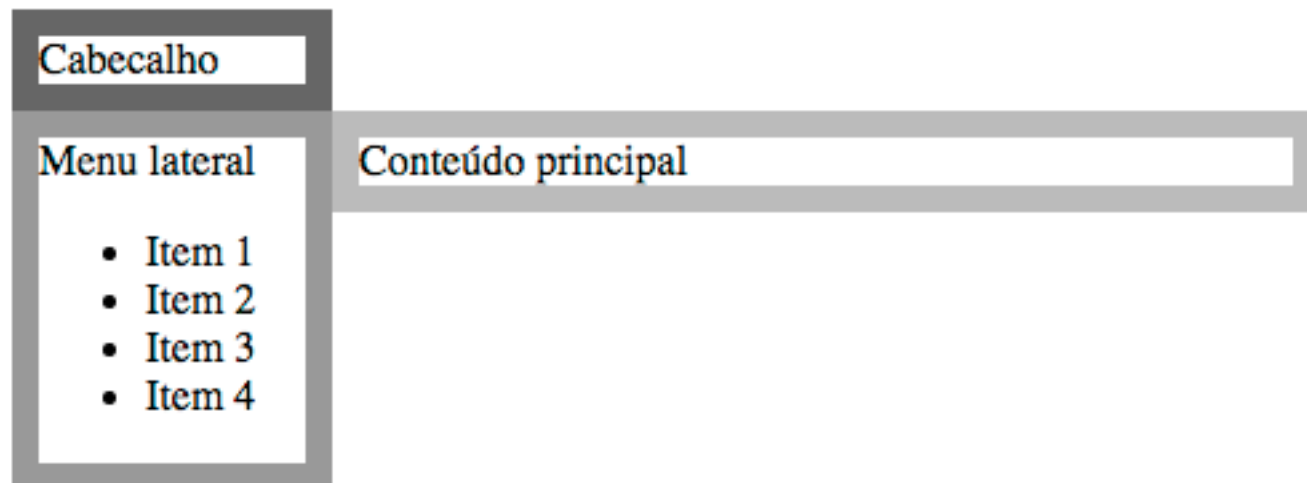
CSS:

```
header, aside, section {  
    float:left;  
}
```

```
<header>  
    Cabecalho  
</header>  
<aside>  
    Menu lateral  
    <ul>  
        <li>Item 1</li>  
        <li>Item 2</li>  
        <li>Item 3</li>  
        <li>Item 4</li>  
    </ul>  
</aside>  
<section>  
    Conteúdo principal  
</section>
```

Exemplo – com clear:left

```
header, aside, section {  
    float:left;  
}  
aside {  
    clear: left;  
}
```





Seletores de atributo do CSS₃


Seletores avançados do CSS.

- Um dos seletores CSS mais versáteis é o seletor de atributo, com ele podemos verificar a presença ou valor de um atributo para selecioná-lo:
 - `input[value] { color: #cc0000; }`
O seletor acima age em todos os elementos da tag `<input>` que têm o atributo "value".
- Também é possível verificar se o atributo tem um valor específico:
 - `input[type="text"] { border-radius: 4px; }`



Seletores avançados

- Seletores de atributos com prefixo
 - `div[class|="menu"] { border-radius: 4px; }`
- Também é possível buscar por uma palavra específica no valor, não importando o valor completo do atributo. Por exemplo:
 - `input[value~="problema"] { color: #cc0000; }`



```
/* busca por inputs com valor de "name" iniciando em "usuario" */
input[name^="usuario"] {
  color: 99ffcc;
}
/* busca por inputs com valor de "name" terminando em "teste" */
input[name$="teste"] {
  background-color: #ccff00;
}
/* busca por inputs com valor de "nome" contendo "tela" em
qualquer posição */
input[name*="tela"] {
  color: #666666;
}
```



Seletores Avançados

- Seletores irmãos

`h2 ~ p { font-style: italic; }`

- Seletores irmãos adjacentes

`h2 + p { font-variant: small-caps; }`

- Seletores de filho direto

/ vai pegar todos os h1 da página */*

`h1 { color: blue; }`

Tentar o seletor de hierarquia também não vai ajudar:

/ vai pegar todos os h1 dos article, incluindo de dentro da section */*

`article h1 { color: blue; }`



Seletor de negação

Imagine o seguinte o HTML com vários parágrafos simples:

```
<p>Texto</p>
```

```
<p>Outro texto</p>
```

```
<p class="especial">Texto especial</p>
```

```
<p>Mais texto</p>
```

- CSS:

```
p { color: gray; }
```

```
p.especial { color: black; /* restaura cor do especial */ }
```

- ou

```
p:not(.especial) { color: gray; }
```



Pseudo-classes

- Considere:

```
<ul>  
  <li>Primeiro item</li>  
  <li>Segundo item</li>  
  <li>Terceiro item</li>  
  <li>Quarto item</li>  
</ul>
```

- E se quisermos formatação especial no primeiro e último item???

```
<ul>  
  <li class="primeiro">Primeiro item</li>  
  <li>Segundo item</li>  
  <li>Terceiro item</li>  
  <li class="ultimo">Quarto item</li>  
</ul>
```



Problemas???



Solução

- No CSS:

- `li:first-child { color: red; }`

- `li:last-child { color: blue; }`

- Escolhendo pelo índice do item: `:nth-child()`

- `li:nth-child(3) { color: yellow; }`



Pseudo-classe de estado

/ seleciona o link no exato momento em que passamos o mouse por cima dele */*

```
a:hover { background-color:#FF00FF; }
```

/ seleciona todas as âncoras que têm o atributo "href", ou seja, links */*

```
a:link { background-color:#FF0000; }
```

/ seleciona todos os links cujo valor de "href" é um endereço já visitado */*

```
a:visited { background-color:#00FF00; }
```

/ seleciona o link no exato momento em que clicamos nele */*

```
a:active { background-color:#0000FF; }
```



Pseudo-elementos

`<p>Lorem ipsum et aepctis non aest setum!</p>`

○ Capitulando a primeira letra:

`p span { font-size: 200%; }`

ou

`<p>Lorem ipsum et aepctis non aest setum!</p>`

`p:first-letter{font-size: 200%; }`



Outros

`p:first-line { font-style: italic; }`

`[Link 1] [Link 2] [Link 3]`

HTML seria simples:

```
<a href="#">Link1</a>  
<a href="#">Link2</a>  
<a href="#">Link3</a>
```

E no CSS:

```
a:before { content: '['; }  
a:after {  
  content: ']; }  
}
```



Exercícios

- Capítulo 6 da apostila da caelum



Bordas arredondadas

○ border-radius

```
div {  
    /*todas as bordas arredondadas com um raio de 5px*/  
    border-radius: 5px;  
}  
.b{ /*borda superior esquerda e inferior direita com 5px borda superior direita e  
    inferior esquerda com 20px*/  
    border-radius: 5px 20px;  
}  
.c{ /*borda superior esquerda com 5px borda superior direita e inferior esquerda  
    com 20px borda inferior direita com 50px */  
    border-radius: 5px 20px 50px;  
}  
.d{ /*borda superior direita com 5px borda superior direita com 20px borda  
    inferior direita com 50px bordar inferior esquerda com 100px */  
    border-radius: 5px 20px 50px 100px;  
}
```



Sombreamento de texto

○ text-shadow

```
p{  
  /*deslocamento à esquerda de 5px e abaixo de 10px*/  
  text-shadow: 5px 10px red; //  
}
```

- Com blur:

```
p{  
  /*deslocamento à esquerda de 15px e abaixo de 10px e blur de 5px*/  
  text-shadow: 15px 10px 5px red;  
}
```

- Mais de uma sombra ao mesmo tempo:

```
p{  
  text-shadow: 10px 10px 5px red, -5px -5px 4px red;  
}
```



Sombreamento de borda

- box-shadow:

box-shadow: 20px 20px black;

- com blur:

box-shadow: 20px 20px 20px black;

- aumentando o tamanho da sombra:

box-shadow: 20px 20px 20px 30px black;

- inset (borda interna):

box-shadow: inset 0 0 40px black;

Opacidade



```
p{  
  opacity: 0.3;  
}
```





RGBA

- Padrão de definição de cores

/ todos equivalentes */*

color: #FFFFFF;

color: white;

color: rgb(255, 255, 255);

- A de Alpha – transparência:

/ branco com 80% de opacidade */ color:*

rgba(255,255,255, 0.8);



Prefixos

- Os prefixos dos fabricantes mais famosos são:
 - **-webkit-**: navegadores Webkit (Chrome, Safari, iOS, Android)
- - **-moz-**: Firefox (Mozilla)
 - **-ms-**: Internet Explorer (Microso)
 - **-o-**: Opera
- Exemplo:

```
p{
  /* Chrome até versão 3, Safari até versão 4 */
  -webkit-border-radius: 5px;
  /* Firefox até 3.6 */
  -moz-border-radius: 5px;
  /* Todas as versões modernas dos navegadores, incluindo IE e Opera
  que nunca precisaram de prefixo pra isso */
  border-radius: 5px;
}
```



Gradientes

○ Linear

```
.linear { background: linear-gradient(white, blue); }
```

○ Radial

```
.radial { background: radial-gradient(white, blue); }
```

○ Outros

```
.gradiente {  
  background: linear-gradient(45deg, #f0f9ff 0%, #cbefff 47%,  
    #a1dbff 100%);  
}
```



CSS3 Transitions

- Permite criar animações na mudanças de algum valor de CSS

- Exemplo

```
#teste {  
    position: relative;  
    top: 0;  
}  
#teste:hover {  
    top: 30px;  
    transition: top 2s;  
}
```

Se tiver várias animações ao mesmo tempo, podemos usar o atalho **all** (que já é o valor padrão, inclusive):



Tipos de animação com Transition

- linear - velocidade constante na animação; (padrão)
- ease - redução gradual na velocidade da animação;
- ease-in - aumento gradual na velocidade da animação;
- ease-in-out - aumento gradual, depois redução gradual na velocidade da animação;
- cubic-bezier(x1,y1,x2,y2) - curva de velocidade para animação customizada (avançado);



CSS Transforms – translate

○ Translate

- move o elemento

```
.header { /* Move o elemento no eixo horizontal */  
  transform: translateX(50px);  
}  
#main { /* Move o elemento no eixo vertical */  
  transform: translateY(-20px);  
}  
footer { /* Move o elemento nos dois eixos (X, Y) */  
  transform: translate(40px, -20px);  
}
```



CSS Transforms – rotate

- Rotaciona o elemento

- Exemplo:

```
#menu-departamentos {  
    transform: rotate(-10deg);  
}
```



CSS Transform - Scale

- Altera a escala do objeto

```
#novidades li {
```

```
  /* Alterar a escala total do elemento */
```

```
  transform: scale(1.2);
```

```
}
```

```
#mais-vendidos li {
```

```
  /* Alterar a escala vertical e horizontal do elemento */
```

```
transform: scale(1, 0.6);
```

```
}
```




CSS Transform - Skew

- Distorce o elemento

```
footer {
```

```
/* Distorce o elemento no eixo horizontal */
```

```
  transform: skewX(10deg);
```

```
}
```

```
#social {
```

```
/* Distorce o elemento no eixo vertical */
```

```
  transform: skewY(10deg);
```

```
}
```



CSS Transform

- Aplicando mais de uma propriedade ao mesmo tempo:

```
html {  
  transform: rotate(-30deg) scale(0.4);  
}
```