

PROGRAMAÇÃO WEB

Prof. Ms. Wilson Lourenço

wilson.slourenco@sp.senac.br



CSS AVANÇADO

- Desde o surgimento do CSS, os desenvolvedores front-end utilizam diversas técnicas para alterar a exibição dos elementos no navegador. Mesmo assim algumas coisas eram impossíveis de se conseguir utilizando somente CSS.
- Mas conhecendo o comportamento dos navegadores ao exibir um elemento (ou um conjunto de elementos) e como as propriedades do CSS agem ao modificar um elemento é possível obter resultados impressionantes

- O CSS3 permite coisas antes impossíveis como elementos com cor ou fundo gradiente, sombras e cantos arredondados. Antes só era possível atingir esses resultados com o uso de imagens e às vezes até com um pouco de JavaScript.
- A redução do uso de imagens traz grandes vantagens quanto à performance e quantidade de tráfego de dados necessária para a exibição de uma página.

Seletores Avançados

- Os seletores CSS mais comuns e tradicionais são os que já vimos: por ID, por classes e por descendência.
- No entanto, há muitos outros seletores novos que vão entrando na especificação e que são bastante úteis.
- Já vimos alguns, como os seletores de atributo que usamos anteriormente.
- Vejamos outros.

Seletor de Irmãos

- Veja o seguinte HTML, que simula um texto com vários parágrafos, títulos e subtítulos no meio do documento:

```
<article>  
  <h1>Título</h1>  
  <p>Início</p>  
  <h2>Subtítulo</h2>  
  <p>Texto</p>  
  <p>Mais texto</p>  
</article>
```

- Como faremos se quisermos estilizar de uma certa maneira todos os parágrafos após o subtítulo?
- O seletor de irmãos (siblings) (~) serve pra isso. Ele vem do CSS3 e funciona em todos os navegadores modernos (e no IE7 em diante)

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

- Isso indica que queremos selecionar todos os p que foram precedidos por algum h2 e são irmãos do subtítulo (ou seja, estão sob a mesma tag pai).
- No HTML anterior, serão selecionados os dois últimos parágrafos (Texto e Mais texto).

Seletor de Irmão Adjacente

- Ainda com o HTML anterior, o que fazer se quisermos selecionar apenas o parágrafo imediatamente seguinte ao subtítulo?
- Ou seja, é um p irmão do h2 mas que aparece logo na sequência.

- Fazemos isso com o seletor de irmão adjacente - adjacent sibling:

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

- Nesse caso, apenas o parágrafo Texto será selecionado. É um irmão de h2 e aparece logo depois do mesmo.
- Esse seletor faz parte da especificação CSS2.1 e tem bom suporte nos navegadores modernos, incluindo o IE7.

Seletor de Filho Direto

- Se tivermos o seguinte HTML com títulos e seções de um artigo:

`<article>`

`<h1>Título principal</h1>`

`<section>`

`<h1>Título da seção</h1>`

`</section>`

`</article>`

- Queremos deixar o título principal de outra cor. Como fazer? O seletor de nome de tag simples não vai resolver:

`/* vai pegar todos os h1 da página */`

```
h1 {  
    color: blue;  
}
```

- Tentar o seletor de hierarquia também não vai ajudar:

`/* vai pegar todos os h1 do article, incluindo o de dentro da section */`

```
article h1 {  
    color: blue;  
}
```

- Entra aí o seletor de filho direto (>) do CSS2.1 e suportado desde o IE7 também.

/* vai pegar só o h1 principal, filho direto de article e não os netos */

```
article > h1 {  
    color: blue;  
}
```

Negação

- Imagine o seguinte HTML com vários parágrafos simples:

`<p>Texto</p>`

`<p>Outro texto</p>`

`<p>Texto especial</p>`

`<p>Mais texto</p>`

- Queremos fazer todos os parágrafos de cor cinza, exceto o que tem o texto especial. Precisamos destacá-lo de alguma forma no HTML para depois selecioná-lo no CSS. Uma classe ou ID resolve:

`<p>Texto</p>`

`<p>Outro texto</p>`

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

`<p>Mais texto</p>`

- Mas como escrever o CSS? Queremos mudar a cor dos parágrafos que não têm a classe especial. Um jeito seria mudar a cor de todos e sobrescrever o especial depois:

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

- No CSS3, há uma outra forma, usando o seletor de negação. Ele nos permite escrever um seletor que pega elementos que não batem naquela regra.

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

- Isso diz que queremos todos os parágrafos que não têm a classe especial.
- A sintaxe do `:not()` recebe como argumento algum outro seletor simples (como classes, IDs ou tags).
- Essa propriedade do CSS3 possui suporte mais limitado no IE, somente a partir da versão 9 (nos outros navegadores não há problemas).

PSEUDO-CLASSES

- Observe o seguinte HTML de uma lista de elementos:

``

`Primeiro item`

`Segundo item`

`Terceiro item`

`Quarto item`

``

- E se quisermos estilizar elementos específicos dessa lista?
- Por exemplo, o primeiro elemento deve ter cor vermelha e o último, azul.
- Com esse HTML simples, usando apenas os seletores que vimos até agora, será bem difícil.

- A solução mais comum seria adicionar classes ou IDs no HTML para selecioná-los depois no CSS:

<li class="primeiro">Primeiro item

Segundo item

Terceiro item

<li class="ultimo">Quarto item

- Agora é fácil usar cores diferentes para o primeiro e último itens da lista. Mas essa técnica exigiu alteração no HTML e exige que lembremos de colocar a classe correta, no ponto correto, toda vez que fizemos mudanças nos itens da lista.
- O CSS tem um recurso chamado de pseudo-classes que são como classes CSS já pré-definidas para nós.
- É como se o navegador já colocasse certas classes por padrão em certos elementos, cobrindo situações comuns como essa de selecionar o primeiro ou o último elemento.

- Há duas pseudo-classes do CSS3 que representam exatamente o primeiro elemento filho de outro (first-child) e o último elemento filho (last-child). Essas classes já estão definidas, não precisamos aplicá-las em nosso HTML e podemos voltar para o HTML inicial:

Primeiro item

Segundo item

Terceiro item

Quarto item

- No CSS, podemos usar pseudo-classes quase da mesma forma que usaríamos nossas classes normais.
- Repare que para diferenciar um tipo do outro, mudou-se o operador de ponto para dois pontos:

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

- O suporte a esses seletores é completo nos navegadores modernos.
- O first-child vem desde o IE7, Firefox 3 e Chrome 4.
- E o last-child só a partir do IE9 mas desde o Firefox 1 e Chrome 1.

- Um seletor ainda mais genérico do CSS3 é o `:nth-child()` que nos permite passar o índice do elemento.
- Por exemplo, podemos pegar o terceiro item com:

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

- Porém, o mais interessante é que o `nth-child` pode receber uma expressão aritmética para indicar quais índices selecionar. É fácil fazer uma lista zebra, com itens ímpares de uma cor e pares de outra:

```
/* elementos pares */
```

```
li:nth-child(2n) { color: green; }
```

```
/* elementos ímpares */
```

```
li:nth-child(2n+1) { color: blue; }
```

- O suporte existe a partir do IE9, Firefox 3.5 e Chrome 1.

Pseudo Classes de Estado

- Queremos mudar a cor de um link quando o usuário passa o mouse por cima.
- Ou seja, queremos mudar seu visual a partir de um evento do usuário (no caso, passar o mouse em cima).

- Uma solução ingênua seria criar um código JavaScript que adiciona uma classe nos links quando o evento de mouseover acontece (e remove a classe no mouseout).
- Entretanto, o CSS possui excelentes pseudo-classes que representam estados dos elementos e, em especial, uma que representa o momento que o usuário está com o mouse em cima do elemento, a :hover.

- É como se o navegador aplicasse uma classe chamada `hover` automaticamente quando o usuário passa o mouse em cima do elemento e depois retirasse a classe quando ele sai. Tudo sem precisarmos controlar isso com JavaScript.

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

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

- Podemos usar hover em todo tipo de elemento, não apenas links. Mas os links ainda têm outras pseudo-classes que podem ser úteis:

```
/* 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

- Pseudo-classes nos ajudam a selecionar elementos com certas características sem termos que colocar uma classe manualmente neles.
- Porém, o que fazer quando precisamos selecionar certo tipo de conteúdo que nem elemento tem?

- Exemplo: temos um texto num parágrafo:

`<p>A universidade tem os melhores cursos!</p>`

- Queremos dar um estilo de revista ao nosso texto e estilizar apenas a primeira letra da frase com uma fonte maior. Como fazer para selecionar essa letra? A solução ingênua seria colocar um elemento ao redor da letra para podermos selecioná-la no CSS:

`<p>A universidade tem os melhores cursos!</p>`

- Usando o CSS, pode-se usar o conceito de pseudo-elementos. São elementos que não existem no documento mas podem ser selecionados pelo CSS. É como se houvesse um elemento lá!
- Voltamos ao HTML inicial:

<p>A universidade tem os melhores cursos!</p>

- E no CSS:

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

- Temos ainda outro pseudo-elemento para selecionar a primeira linha apenas em um texto grande:

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

Novos Conteúdos

- Há ainda um outro tipo de pseudo-elemento mais poderoso que nos permite gerar conteúdo novo via CSS. Imagine uma lista de links que queremos, visualmente, colocar entre colchetes:

[Link 1]

[Link 2]

[Link 3]

- Podemos, claro, apenas escrever os tais colchetes no HTML. Mas será que o conteúdo é semântico?
- Queremos que esses colchetes sejam indexados pelo Google? Queremos que sejam lidos como parte do texto pelos leitores de tela?
- Talvez não. Pode ser um conteúdo apenas visual. Podemos gerá-lo com CSS usando os pseudo-elementos `after` e `before`.

- O HTML seria simples:

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

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

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

- E no CSS:

```
a:before {  
    content: '[';
```

```
}
```

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

- Ou ainda, imagine que queremos colocar a mensagem (externo) ao lado de todos os links externos da nossa página. Usando pseudo-elementos e seletores de atributo, conseguimos:

```
a[href^=http://]:after {  
    content: ' (externo)';  
}
```

- Isso pega todos os elementos `<a>` que começam com `http://` e coloca a palavra (externo) depois.

Exercícios – Aplicando esses conceitos na nossa página

Dicas para Estudo



Seja “CURIOSO”:

Procure revisar o que foi estudado.

Pesquise as referências bibliográficas.



Seja “ANTENADO”:

Leia a próxima aula.



Seja
“COLABORATIVO”:

Traga assuntos relevantes para a sala de aula.

Participe da aula.

Proponha discussões relevantes sobre o conteúdo.



Prof. Ms. Wilson Lourenço



**Dúvidas?
Não mais..**