Universidade do Oeste de Santa Catarina

Disciplina: Desenvolvimento com HTML5 e CSS3

Professor: Jean Carlo Nascimento

Acadêmico: Ednilson Fernando Devilla

**CSS3**

# Negação

Quando todos os parágrafos já tem um estilo de cor definido e precisa destacar uma cor diferente em um outro paragrafo do texto, usando uma classe resolve. Abaixo segue o exemplo.

**HTML**

Quadro 1: Exemplo HTML

|  |
| --- |
| <p>Texto de exemplo</p>  <p>Texto de exemplo</p>  <p class="especial">Texto de exemplo especial</p>  <p>Mais textos de exemplos</p> |

Fonte: W3C, (2014).

**CSS**

Quadro 2: Exemplo CSS

|  |
| --- |
| **p**{  **color**: gray;  }  **p**.especial{  **color**: black; /\* cor do texto especial \*/  } |

Fonte: W3C, (2014).

Outra forma de se fazer é com elementos que não batem naquela regra, isso quer dizer que, queremos todos os parágrafos que não têm a classe especial. A sintaxe do *:not()* recebe como argumento algum outro seletor simples como classes, id ou tags. Abaixo o exemplo.

**CSS**

Quadro 3: Exemplo CSS :not()

|  |
| --- |
| **p**:not(.especial) {  **color**: gray;  } |

Fonte: W3C, (2014).

# Pseudo-classes

Quando se quer mudar a cor do primeiro e do ultimo elemento de uma lista, com elementos que já são pré definidos os elementos são chamamos de pseudo-classes. O primeiro elemento chamamos de *first-child* e o ultimo elemento de *last-child*. O HTML fica igual, vejamos o exemplo no quadro abaixo.

Quadro 4: Exemplo HTML

|  |
| --- |
| **<ul>**  **<li>**Primeiro item**</li>**  **<li>**Segundo item**</li>**  **<li>**Terceiro item**</li>**  **<li>**Quarto item**</li>**  **</ul>** |

Fonte: Caelum, (2010).

No CSS, podemos usar pseudo-classes, repare no exemplo apresentado no quadro 5 é que mudou o operador de ponto para dois pontos, diferenciando um do outro.

Quadro 5: Exemplo CSS

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

Fonte: Caelum, (2010).

# Pseudo elementos

Pseudo elementos são elementos que não existem no documento HTML mas podem ser selecionados através do CSS. Por exemplo o HTML fica de uma forma normal como apresentado no quadro 6.

Quadro 6: Exemplo HTML

|  |
| --- |
| <p>Exemplo de texto no HTML!</p> |

Fonte: Maujor, (2006).

O CSS apresentado no quadro 7, modifica apenas a primeira letra.

Quadro 7: Exemplo CSS

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

Fonte: Maujor, (2006).

Existe também outro Pseudo elemento que seleciona apenas a primeira linha.

Quadro 8: Exemplo CSS

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

# Fonte: Maujor, (2006).

# Transitions

Com as transitions conseguimos animar o processo de mudança de algum valor do CSS. Por exemplo, temos um elemento na posição *top:20px* e, quando passarmos o mouse em cima (*hover*), queremos que o elemento vá para *top:40px.* No quadro 9 podemos ver o exemplo.

Quadro 9: Exemplo CSS

|  |
| --- |
| #teste{  position: relative;  top: 0;  }  #teste:hover{  top: 30px;  } |

Fonte: Caelum, (2010)

No exemplo acima o elemento é deslocado de uma vez, quando passamos o mouse em cima, caso queira que o elemento mude suavemente basta usar o código apresentado no quadro 10.

Quadro 10: Exemplo CSS

|  |
| --- |
| #teste:hover{  transition: top 2s;  } |

Fonte: Caelum, (2010)

Isso indica que queremos animar a propriedade *top* durante 2 segundos.

Por padrão, a animação é linear, mas temos outros tipos para animações mais suaves:

Quadro 11: Exemplo de animações

|  |
| --- |
| **linear** - velocidade constante na animação;  **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); |

Fonte: Caelum, (2010).

Quadro 12: Exemplo CSS

|  |
| --- |
| #teste:hover{  transition: top 2s ease;  } |

Fonte: Caelum, (2010)

# Transforms

Através dessa nova especificação é possível alterar propriedades visuais dos elementos. Por exemplo, mostre uma escala maior ou menor que seu tamanho padrão ou alterar a posição do elemento sem sofrer interferência de sua estrutura.

No quadro 13 apresenta-se a propriedade *transalate*.

## 5.1 Translate

Move os elementos entre os eixos X e Y.

Quadro 13: Exemplo CSS

|  |
| --- |
| .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);  } |

Fonte: W3C, (2014).

## 5.2 Rotate

Rotaciona os elementos.

Quadro 14: Exemplo de CSS, que rotaciona os elementos.

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

Fonte: W3C, (2014).

## 5.3 Scale

Altera escalas de elementos.

Quadro 15: Exemplo de CSS, scale.

|  |
| --- |
| #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);  } |

Fonte: W3C, (2014).

## 5.4 Skew

Distorce os elementos.

Quadro 16: Exemplo CSS, distorce os elementos selecionados.

|  |
| --- |
| footer{  /\* Distorcer o elemento no eixo horizontal \*/  transform: skewX(10deg);  }  #social{  /\* Distorcer o elemento no eixo vertical \*/  transform: skewY(10deg);  } |

Fonte: W3C, (2014).

Ainda pode-se combinar varias transformações no mesmo elemeto. O que deve-se fazer é declarar uma depois da outra. No quadro 17 podemos ver o exemplo.

Quadro 17: Exemplo CSS, combinando vários elementos

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

# Fonte: W3C, (2014).

# Seletores avançados

## 6.1 Seletor de irmãos

O quadro 18 apresenta um HTML que simula vários parágrafos, títulos e subtítulos em um documento.

Quadro 18: Exemplo HTML

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

Fonte: Maujor, (2006).

Quadro 19: Exemplo CSS, seletor de irmãos

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

Fonte: Maujor, (2006).

O seletor de irmãos (~) 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).

## 6.2 Seletor de irmão adjacentes

Utilizando o HTML apresentado no quadro 18, quando se quer selecionar o paragrafo *<p>* e seguinte já existe um subtítulo *<h2>*, utilizamos o seletor de irmãos adjacentes.

Quadro 20: Exemplo CSS com seletor de irmão adjacentes

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

Fonte: Maujor, (2006).

Nesse caso apenas o texto será selecionado.

# Text-shadow

O text-shadow serve para colocar sombras em textos. Sua sintaxe é fácil de entender, recebe-se o deslocamento da sombra e a cor.

Quadro 21: Exemplo CSS, text-shadow

|  |
| --- |
| p{  text-shadow: 10px 10px red;  } |

Fonte: W3C, (2014).

É possível até passar mais de uma sombra ao mesmo tempo para o mesmo elemento:

Quadro 22: Exemplo CSS, passando mais de uma sombra

|  |
| --- |
| text-shadow: 10px 10px 5px red, -5px -5px 4px red; |

Fonte: W3C, (2014).

# Box-shadow

Podemos colocar sombras em qualquer elemento com o box-shadow, a sintaxe é parecida com a do text-shadow, deslocamento da sombra e a cor.

Quadro 23: Exemplo CSS com box-shadow

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

Fonte: W3C, (2014).

O box-shadow suporta mais um valor que faz a sombra aumentar ou diminuir, vejamos o exemplo no quadro 24

Quadro 24: Exemplo CSS

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

Fonte: W3C, (2014).

# Opacidade e RGBA

É possível mudar a opacidade de um elemento para que ele seja mais transparente com o uso da propriedade *opacity.* É simples, a propriedade *opacity* recebe um valor decimal entre 0 e 1, vejamos no quadro 25 o exemplo.

Quadro 25: Exemplo CSS, opacity

|  |
| --- |
| p{  opacity: 0.3;  } |

Fonte: W3C, (2014).

Além das cores hexadecimais normais (#FFFFFF pro branco), podemos criar cores a partir de seu valor RGB, passando o valor de cada canal (Red, Green, Blue) separadamente (valor entre 0 e 255).

Quadro 26: Exemplo CSS, com todas equivalentes

|  |
| --- |
| color: #FFFFFF;  color: white;  color: rgb(255, 255, 255); |

Fonte: W3C, (2014).

Porém, existe uma função chamada RGBA que recebe um quarto argumento, o chamado canal Alpha. Na prática, seria como o *opacity* daquela cor (um valor entre 0 e 1), vejamos no quadro 27.

Quadro 27: Exemplo CSS, com a função RGBA

|  |
| --- |
| /\* branco com 80% de opacidade \*/  color: rgba(255,255,255, 0.8); |

Fonte: W3C, (2014).

# Gradientes

O CSS3 traz suporte a declaração de gradientes sem que se precise trabalhar com imagens, além de ser simples a página fica mais leve e a renderização fica melhor por se adaptar a todos os tipos de resolução. Existe suporte a gradientes lineares e radiais. No quadro 28 podemos ver melhor a sintaxe básica.

Quadro 28:Exemplo CSS de gradiente

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

Fonte: Caelum, (2010)

No quadro 28 apresenta-se os exemplos dos tipos de gradientes lineares e radias, passando as cores branco *(white*) e azul (*blue*).

Podemos ainda usar gradientes com angulações diferentes e diversas paradas de cores.

Quadro 29: Exemplo CSS com angulações diferentes

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

Fonte: Caelum, (2010)

# Foco

Coloca o foco no elemento quando ele for selecionado, no exemplo no quadro 30 apresenta o HTML normal, a única mudança ocorre no CSS.

Quadro 30: Exemplo HTML

|  |
| --- |
| <input type="text"> |

Fonte: Caelum, (2010)

Veja que o HTML não muda, acima apresenta um simples *input* do tipo *text,* no quadro 31 apresenta o CSS referente ao quadro 30.

Quadro 31: Exemplo CSS, quando o foco está no input

|  |
| --- |
| input:focus{  background: yellow;  border: 1px solid black;  } |

Fonte: Caelum, (2010)

No quadro 31 apresenta que quando o foco está no *input*, à cor dele muda para amarelo (*yellow*), quando o foco sai do *input* a cor volta ao normal.

# Prefixos

Muitos dos novos recursos do CSS3 ainda são experimentais , isso quer dizer que foram incluídos nos rascunhos, mas ainda não são 100% oficias. As especificações ainda estão em aberto e vai demorar um pouco até serem finalizadas.

Existem recursos mais estáveis e menos estáveis. Por outro lado, alguns são bem recentes e talvez ainda possa haver mudança até a aprovação final da especificação.

Os navegadores desejam implementar os novos recursos antes mesmo da especificação terminar, para que os desenvolvedores possam começar a usar as novas propriedades e experimentá-las na prática. Entretanto, como a sintaxe final depois de aprovada pode ser diferente, os navegadores implementam as novas propriedades instáveis usando nomes provisórios.

Uma boa pratica na utilização é pegar o nome da propriedade e colocar um prefixo na especifico do fabricante na frente. Quando a especificação ficar estável tira-se o prefixo e deixa-se a sintaxe oficial.

Os prefixos dos fabricantes mais usados são:

Quadro 32: Prefixos dos fabricantes mais usados

|  |
| --- |
| * **-webkit-**: navegadores Webkit (Chrome, Safari, iOS, Android) * **-moz-**: Firefox (Mozilla) * **-ms-**: Internet Explorer (Microsoft) * **-o-**: Opera |

Fonte: Caelum, (2010)

Quadro 33: Utilizando os prefixos no CSS

|  |
| --- |
| 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;  } |

Fonte: Caelum, (2010)

No quadro 33 apresenta-se a utilização de prefixos utilizados no *border-radius,* fazendo com que o *border-radius* funcione em diversos navegadores, sem que sofra qualquer alteração.