

Propriedade border radius e novidades do CSS3

Transcrição

Até agora vimos no CSS diversas maneiras de fazer nosso *layout*, mas algumas coisas ainda nos limitam muito.

Toda vez que criamos um elemento na página, como uma *div*, o que realmente aparece é uma caixa. Até agora só conseguimos criar *layouts* baseados em caixas, o que não é de todo ruim. Muitos sites se utilizam desse formato. Um deles é o próprio Alura!



Mas será que conseguimos fugir um pouco deste formato (utilizando caixas com bordas arredondadas, cortadas, triângulos e etc.) para enriquecer nosso *layout*? Até certo tempo tínhamos um trabalho considerável para implementar isso, porém agora ficou muito mais fácil com o CSS3.

Antigamente, para fazermos, por exemplo, uma caixa de texto com bordas arredondadas tínhamos que nos utilizar de imagens de fundo para cada borda, o que resultava em quatro *divs* para serem preenchidos:

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

Nosso HTML ficava muito extenso e confuso, além de deixar a página mais pesada por causa da quantidade de imagens.

Agora, com o CSS3, criamos apenas um elemento em nossa página:

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

No CSS dizemos que as bordas devem ser arredondadas com a propriedade *border-radius*:

```
div {  
  border-radius: 10px;  
}
```

A quantidade de pixels nada mais é do que o raio dessa borda que queremos desenhar:



Podemos também criar um elemento que possua apenas duas bordas arredondadas. É só passar um segundo valor. O primeiro valor se refere às bordas esquerda de cima e direita de baixo e o segundo às outras duas. O "0" significa sem borda alterada.

```
div {  
  border-radius: 10px 0;  
}
```

Veremos que o CSS3 não trouxe apenas essa implementação. Existem outras propriedades que melhoram muito os nossos *layouts*. Elas são recentes. Como será que os navegadores implementaram essas funcionalidades?

Como vimos, o *border-radius* se comporta de uma determinada maneira. Mas e se um navegador específico achar que não se deve fazer assim:

```
border-radius: rx rx rx rx / ry ry ry ry;
```

mas assim:

```
border-radius: rx / ry  
              rx / ry  
              rx / ry  
              rx / ry;
```

É normal que haja discordância na maneira de se implementar um código. Os navegadores podem escolher como acharem melhor.

Mas como desenvolvedores, queremos o suporte de todos os navegadores. Quando escrevemos um CSS que funciona para o *Google Chrome*, não necessariamente ele irá funcionar no *Mozilla Firefox*.

Felizmente os navegadores pensaram nesse problema e desenvolveram um prefixo que facilita nossa vida. No *Chrome*, por exemplo:

```
-webkit-border-radius: rx rx rx rx / ry ry ry ry;
```

Isso significa que todas as funcionalidades que o *Chrome* está testando começam com *-webkit-*

Para o *Firefox*:

```
-moz-border-radius: r r r r;
```

Conseguimos com isso escrever o nosso CSS com os dois *border-radius*:

```
-webkit-border-radius: rx rx rx rx / ry ry ry ry;  
-moz-border-radius: r r r r;
```

Como há comandos desconhecidos, ao ler esse código, será que o CSS não quebra? Não, pois ele irá ignorar o que não conhece. Dessa forma conseguimos testar a mesma funcionalidade em dois navegadores diferentes com sintaxes diferentes.

Todos os navegadores têm seu próprio prefixo:

- *Google Chrome*: *-webkit-*
- *Mozilla Firefox*: *-moz-*
- *Internet Explorer*: *-ms-*
- *Opera*: *-o-*
- *Safari*: *-webkit-*

Por baixo dos panos, isso está mudando. Muitos navegadores já se utilizam do mesmo código e está acontecendo uma padronização. Os três prefixos principais se resumem em: *-webkit-*, *-ms-* e *-moz-*.

É importante ressaltar que estamos utilizando funcionalidades *BETA*, ou seja, ainda estão em fase de teste para que os desenvolvedores desses navegadores recebam *feedbacks* de como elas estão se comportando.

Com isso, podemos colocar todos os prefixos no código para testar em todos os navegadores. Mas ainda não é o suficiente, pois no futuro as funcionalidades não estarão mais em teste e o site precisa continuar funcionando, então também se faz necessária a linha sem os prefixos:

```
-webkit-border-radius: 10px;  
-moz-border-radius: 10px;  
-ms-border-radius: 10px;  
-o-border-radius: 10px;  
border-radius: 10px;
```

Desse modo, seu site funciona no presente e fica ***Future Proof***, para quando as funcionalidades não mais estiverem em *BETA*.

Outra pergunta: mesmo assim, ainda se faz necessário deixar o código cheio desses prefixos? Na verdade não, tratando-se do *border-radius* só é preciso:

```
-webkit-border-radius: 10px;  
-moz-border-radius: 10px;  
border-radius: 10px;
```

Sabemos disso porque existem pessoas que trabalham para testar todas essas funcionalidades e já podem nos dizer o que funciona e o que não funciona por meio de um catálogo. O melhor hoje em dia é o *caniuse.com*, um site onde podemos procurar a funcionalidade que queremos testar e ver se ela precisa de prefixo e em qual navegador precisará.

Quando trabalhamos com CSS3 é importante ter em mente esses prefixos para criarmos sites para o maior número de usuários possível.