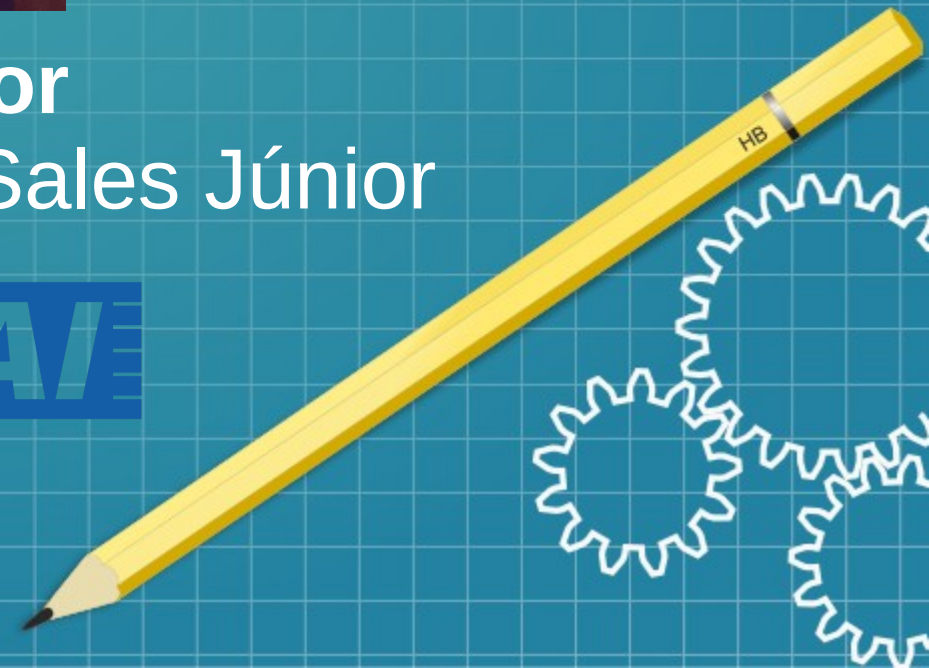


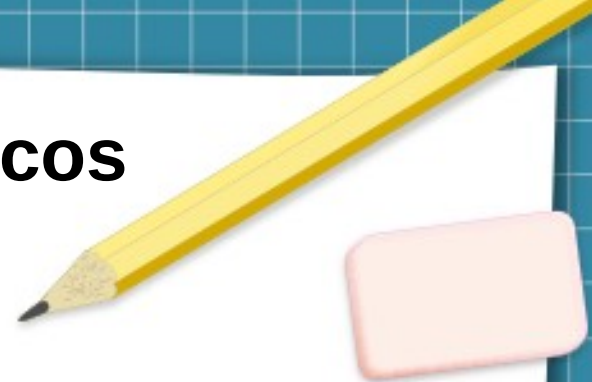
Desenhista de produtos gráficos – Web Design



Professor
Francisco Carlos Sales Júnior



Desenhista de Produtos Gráficos Web Design



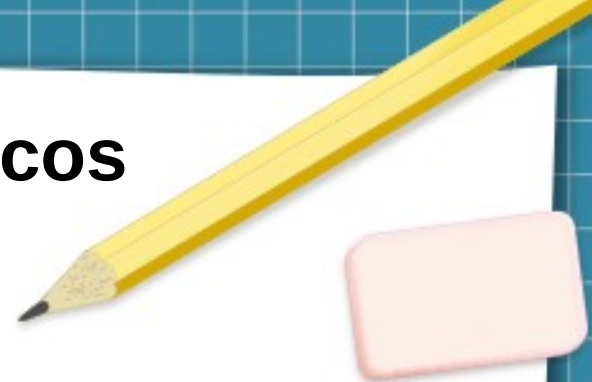
CSS

Elementos arredondados no CSS

Um ponto importante na criação de estilos com CSS é a utilização de elementos arredondados ou redondos. De modo geral, a decisão de como o website irá se comportar visualmente é tomada antes do desenvolvimento. O objetivo é gerar algo visualmente interessante para o usuário, e muitas vezes isso pode não ser possível empregando apenas elementos quadrados ou retangulares. Pensando nisso, nesse artigo veremos o que é necessário para termos elementos com as bordas arredondadas ou totalmente redondos, utilizando somente CSS.

Desenhista de Produtos Gráficos

Web Design

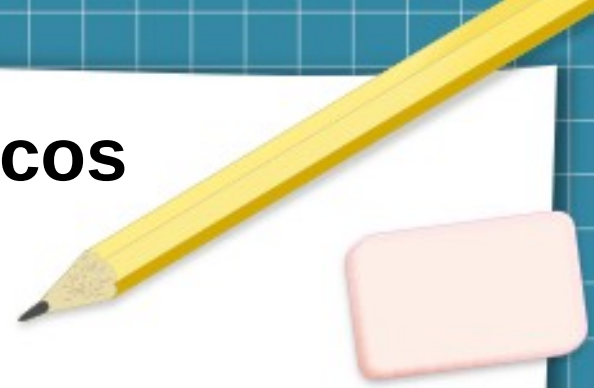


Botão arredondado com CSS

Para entendermos melhor o conceito de “arredondar” elementos utilizando CSS, vamos criar um elemento arredondado. Para isso, iremos definir uma classe CSS capaz de funcionar tanto com o `<button>` quanto com o elemento `<input type=”button”/>`.

A primeira coisa que precisamos especificar a respeito do botão é o que não queremos;

Desenhista de Produtos Gráficos Web Design



CSS

A partir dele, podemos começar a criar os estilos que serão aplicados. Note que não há qualquer definição do que deve ser feito para criar o elemento arredondado que desejamos; apenas algumas propriedades relacionadas a cores e outras informações relevantes.

```
.dvm-button {  
    color: #FFF;  
    background-color: green;  
    border: none;  
    height: 60px;  
    line-height: 30px;  
    padding: 0 20px;  
    text-transform: uppercase;  
    font-weight: bold;  
    cursor: pointer;  
}
```


Desenhista de Produtos Gráficos Web Design



SINTAXE CSS

Linha 01: Definição do nome da classe. Optamos pelo nome “dvm-button”, com um prefixo que remete às iniciais da DevMedia. Esse tipo de nomenclatura é bastante comum em classes CSS;

Linha 02: Propriedade color. Aqui, indicamos que teremos um texto branco (#FFF) no botão;

Linha 03: Propriedade background-color. Especifica que a cor de fundo do botão será verde;

Linha 04: Resetamos a propriedade border, para evitar os valores padrão do elemento <button>;

Linha 05: A altura é definida para o botão (height) em 60 pixels;

Linha 06: Propriedade bastante importante. É ela que centraliza verticalmente o texto do botão. Para esse comportamento, o valor dessa propriedade deve ser exatamente a metade da altura do elemento;

Linha 07: Definição do padding, a margem interna do botão. Nesse caso, optamos por 0 (zero) no eixo y (distância do topo e base) e 20 pixels nas laterais direita e esquerda;

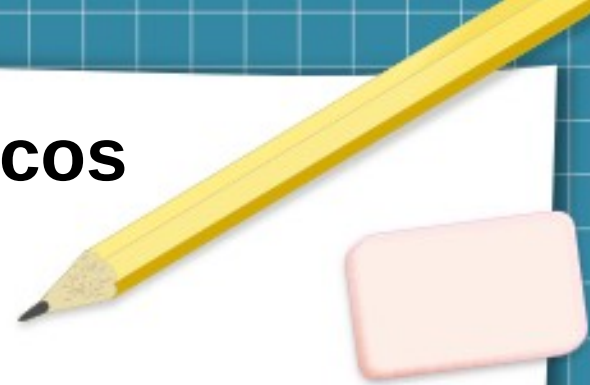
Linha 08: Propriedade text-transform, informa que o texto do botão terá somente letras maiúsculas, independentemente do valor inserido no HTML;

Linha 09: Propriedade font-weight, ou largura da fonte, informa, nesse caso, que temos uma fonte em negrito;

Linha 10: Por meio da propriedade cursor, garante que o botão sempre terá o ponteiro como cursor, ou seja, no momento em que o mouse estiver no botão, teremos esse tipo de cursor.

Desenhista de Produtos Gráficos

Web Design



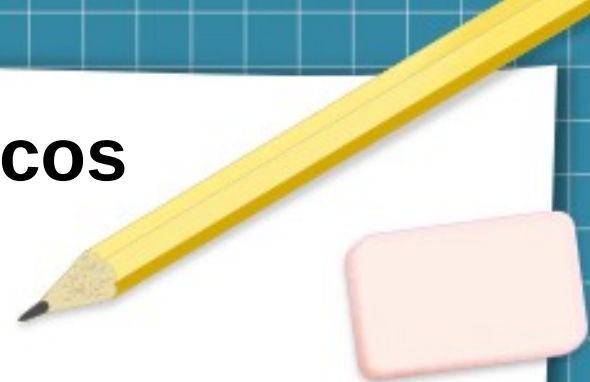
CSS

A parte de arredondamento do elemento, entretanto, ainda não foi criada. Para isso, existe uma propriedade específica do CSS: `border-radius`, ou raio da borda, em uma tradução literal. O raio é uma medida utilizada no círculo (e em outros elementos similares) para indicar a distância da borda em relação ao centro. Assim, ao setarmos o `border-radius`, estamos definindo a distância que essa borda terá em relação a um centro imaginário, arredondando o elemento. Isso fica claro na figura abaixo, que mostra o elemento arredondado que iremos criar.



Desenhista de Produtos Gráficos

Web Design



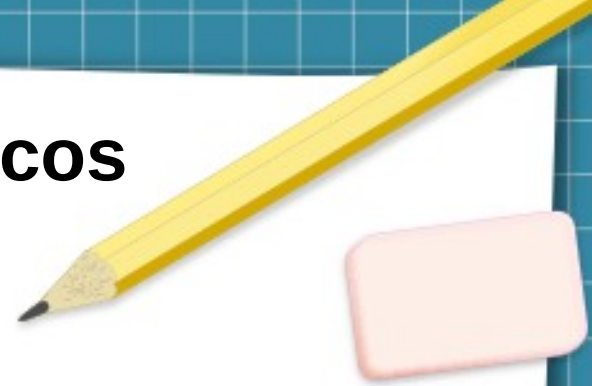
CSS

O código anterior, para definição das demais propriedades do botão, foi abreviado, para focarmos no que realmente interessa neste momento. Ademais, foi adicionada uma ação para o elemento no caso de hover, ou seja, quando o mouse estiver sobre ele.

```
.dvm-button {  
    ...  
    border-radius: 30px;  
}  
  
.dvm-button:hover {  
    transform: scale(1.05);  
}
```

Desenhista de Produtos Gráficos

Web Design



CSS

Linha 03: Definição da propriedade border-radius, responsável pelo arredondamento do elemento que utilizará essa classe (<button>, no caso). Note que temos um raio de 30 pixels, garantindo que o elemento será totalmente arredondado nas laterais, uma vez que o mesmo possui altura de 60 pixels;

Linhas 06 e 07: Classe CSS dvm-button no momento do vhover, ação de manter o cursor sobre o elemento. Nesse caso, apenas utilizamos a propriedade transform para aumentar ligeiramente o botão.

Desenhista de Produtos Gráficos

Web Design



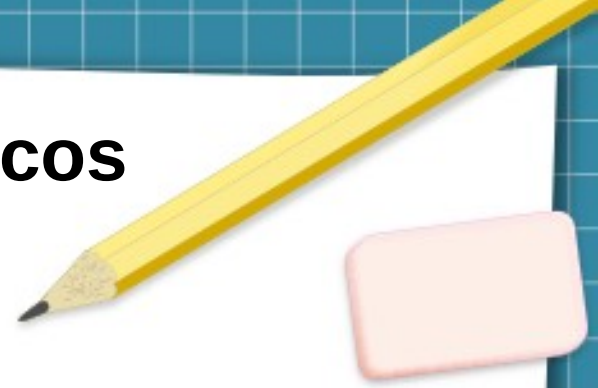
Criando elementos totalmente circulares com CSS

A partir do que vimos em nosso botão arredondado, podemos presumir o que é necessário para termos um elemento totalmente circular utilizando CSS. Note que o botão que criamos, tem as laterais completamente circulares, a partir de um border-radius de 30px em conjunto com uma altura de 60px. A partir disso, é possível concluir que, para termos um elemento totalmente circular, basta configurar um border-radius de 50% da altura (height) e largura (width).

No código da Listagem 3 temos a classe `dvm-circle`. Essa classe segue o mesmo padrão definido anteriormente para a nomenclatura e traz as propriedades necessárias para um elemento circular, como o mostrado na Figura 6.

Desenhista de Produtos Gráficos

Web Design

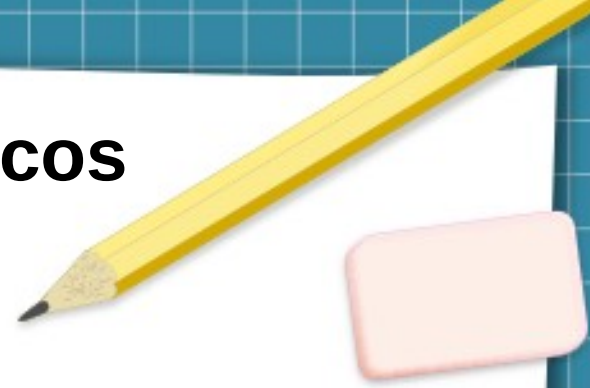


Seletores CSS

```
.dvm-circle {  
    margin: 5px;  
    background-color: black;  
    color: white;  
    padding: 5px;  
    height: 20px;  
    width: 20px;  
    border-radius: 15px;  
}
```

Desenhista de Produtos Gráficos

Web Design



Seletores CSS

Linha 01: Definição da classe `dvm-circle`;

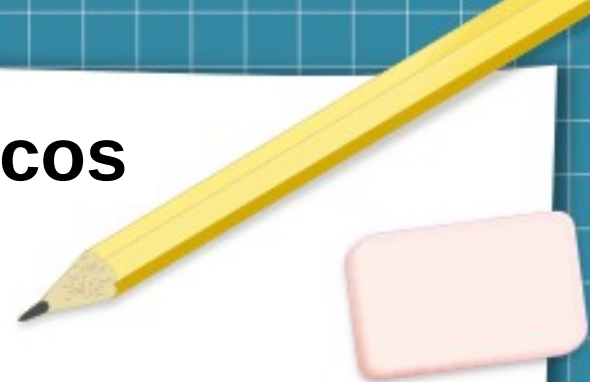
Linhas 02 a 05: Definição das propriedades básicas do elemento. Aqui, informamos que teremos uma margem (`margin`) de 5px em cada lado, bem como o `padding`. As cores de fundo e frente (`background-color` e `color`, respectivamente) também foram especificadas. Note que utilizamos as cores preta para o fundo e branca para os elementos internos, como textos;

Linhas 06 e 07: Especificamos a altura e a largura do elemento circular, ambas de 20px. Somando-se esses valores ao `padding` em cada direção, temos um elemento com altura e largura totais de 30 pixels;

Linha 08: Definição da propriedade `border-radius` baseada na altura total, somando-se o `padding`. Note que o valor é de exatamente 50% da altura e largura do elemento, gerando um visual circular.

Desenhista de Produtos Gráficos

Web Design



Seletores CSS

Apesar de funcionar, a abordagem mostrada na Listagem 3 deve ser evitada, pois é extremamente “enraizada” em um determinado tamanho; 30 pixels, no caso. Para evitar que isso aconteça, podemos informar o valor percentual que comentamos anteriormente – 50% – para o border-radius. Dessa forma, teremos algo completamente independente da altura e largura, o que nos mostra que essas propriedades são dispensáveis no CSS, como podemos confirmar na Listagem 4, que apresenta um código muito similar ao que vimos anteriormente, mas com essas melhorias. O resultado de sua execução será o mesmo

```
.dvm-circle {  
    margin: 5px;  
    background-color: black;  
    color: white;  
    padding: 5px;  
    border-radius: 50%;  
}
```