

BRUNNA CROCHES

PARTE 4

HTML



Artist Image :Descourtilz, Jean-Théodore
| Dates:179?-1855

Guia iniciante: HTML

ABOUT ME



Brunna Croches

Developer Full Stack

Brunna Croches é Dev FullStack, advogada e empreendedora.

Apaixonada por tech, vem adquirindo vasto conhecimento na área.

Desenvolveu projetos ricos em diversidade, buscando captar as próximas tendências e necessidades do mercado.

Neste e-book você aprenderá ou recapitulará de forma simplificada e otimizados conceitos de programação feito por ela.

let's share

SUMMARY



O QUE É < PATH ?>

13

SVG + CSS + HTML

14

*Usando SVG com CSS3 e
HTML5 (material
Complementar)*

13- SVG PATH

https://www.w3schools.com/graphics/tryit.asp?filename=trysvg_path

```
<!DOCTYPE html>
<html>
<body>

<svg height="210" width="400">
  <path d="M150 0 L75 200 L225 200 Z" />
  Sorry, your browser does not support inline SVG.
</svg>

</body>
</html>
```

path d

O elemento `<path>` é usado para definir um caminho.

14- USANDO SVG COM CSS3 e HTML5:

[Usando SVG com CSS3 e HTML5 - Material Complementar](#)

HTML Content

Usando SVG com CSS3 e HTML5 - Material Complementar

Exemplo de código e extras online para o [livro O'Reilly Media](#) de Amelia Bellamy-Royds, Kurt Cagle e Dudley Storey.

[Book Home](#) [Capítulo 7 Resumo](#) [All Online Extras](#)

Aperfeiçoando caminhos para <textPath>

No SVG 1.1 (e em todos os navegadores existentes), o "caminho" usado por um <textPath> layout deve ser realmente um <path> elemento: você ainda não pode fazer o texto andar em <circle> ou <rect>.

Isso pode ser um tanto frustrante, pois elas (junto com as elipses) são as formas mais comuns que as pessoas desejam envolver o texto. Se você tem uma forma básica e deseja convertê-la em um <path> sem precisar codificar ou desenhar um ponto de cada vez, existem várias ferramentas que você pode usar:

- A maioria dos editores visuais possui ferramentas para converter a forma em um caminho. No Illustrator, use Objeto → Caminho composto → Criar; no Inkscape, use Caminho → Objeto para caminho.
- Em alguns editores, incluindo o Illustrator, mover qualquer ponto em uma forma regular, mesmo que ligeiramente, também o converterá em um caminho.
- Se um círculo ou elipse no sentido anti-horário é tudo que você precisa, [a pequena ferramenta JavaScript de Dan Hansen](#) para pegar os dados de caminho equivalentes.

O texto continuará ao longo do caminho na direção em que o caminho é desenhado no código. Se seus caminhos tiverem uma [ordem de enrolamento](#) no sentido horário, o texto ficará do lado de fora da forma. Se o caminho for no sentido anti-horário, o texto ficará do lado de dentro. Em ambos os casos, é possível acabar com letras de cabeça para baixo. Se o seu caminho estiver na direção errada, o seu editor gráfico pode inverter para você:

- No Adobe Illustrator, há um comando Inverter direção do caminho no painel Atributos do aplicativo (pode ser necessário usar Mostrar opções no canto superior direito do painel para ver os controles de direção do caminho). Isso pressupõe que o caminho é um caminho composto (Objeto → Caminho composto → Criar).
- No Inkscape, selecione o caminho e use a opção Inverter no menu Caminho.
- Se você estiver manipulando SVG com scripts em vez de editores visuais, Mike Kamermans criou um [utilitário JavaScript para fazer a reversão](#).

Finalmente, você pode querer ajustar o ponto inicial do texto para uma posição diferente do ponto inicial do path code. Infelizmente, `startOffset` não é suficiente aqui: embora desloque o texto ao redor da forma, não altera o fato de que o texto é cortado quando atinge os pontos inicial e final do caminho. (SVG 2 propõe mudar isso, mas os navegadores ainda não estão a bordo.)

As ferramentas para alterar o ponto inicial de um caminho não são tão comuns quanto para revertê-lo, mas na maioria dos editores gráficos você pode forçar a questão quebrando e unindo novamente seu caminho no ponto escolhido.

Como alternativa, você pode evitar o problema do texto de corte duplicando a string de dados do caminho dentro do <path> elemento, de modo que o texto se enrole e continue ao longo de uma segunda "volta" do caminho. Lembre-se de que agora seu caminho total é duas vezes mais longo, portanto, os `startOffset` valores percentuais precisam ser divididos pela metade.

Gorjeta Para formas simétricas, como um círculo, você pode usar uma `rotate` transformação no <path> próprio (ou no <text> contendo o <textPath>) para reposicionar o início do texto como desejar.

Com todas essas dicas em mente, estamos prontos para organizar o texto em um caminho fechado. [O Exemplo 7-X3](#) cria uma crista circular a partir do lema "Todos por Um e Um por Todos" dos Três Mosqueteiros. [A Figura 7-X3](#) é a saída.

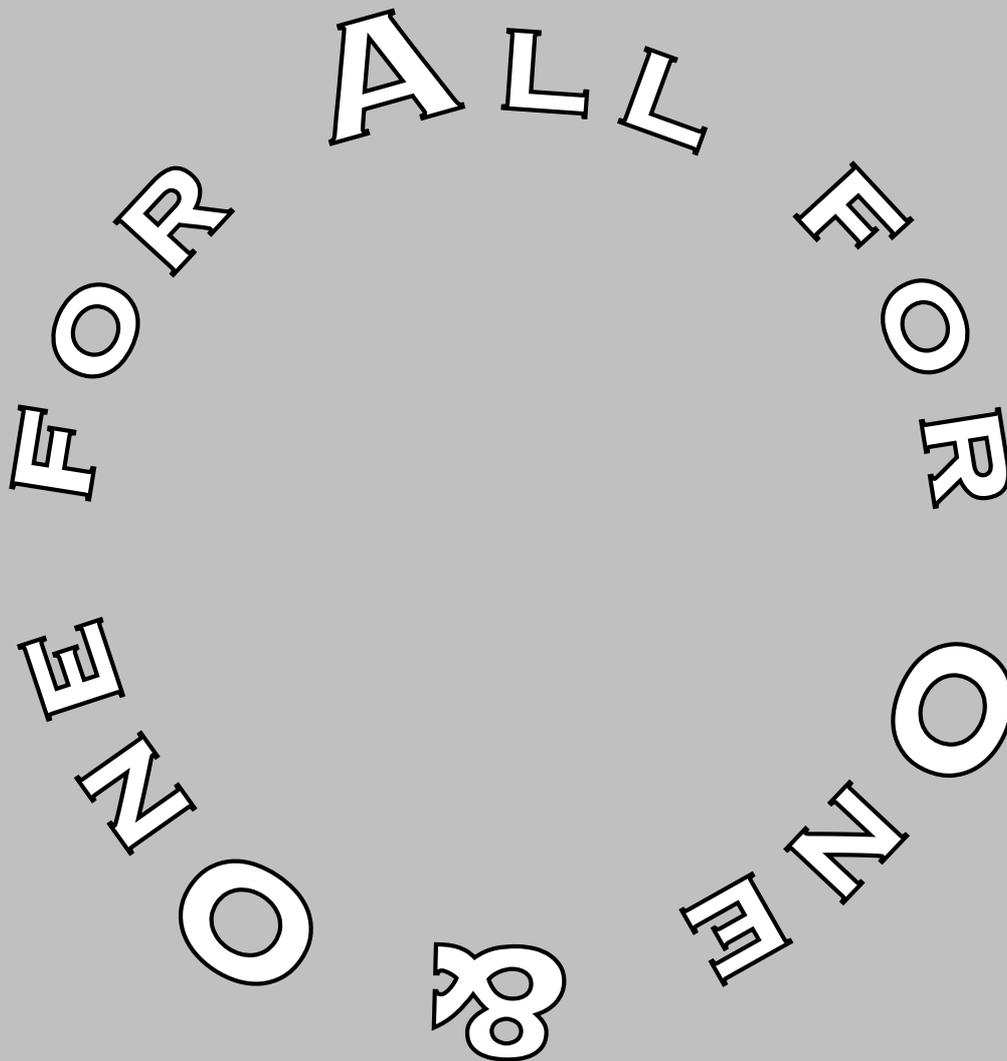


Figura 7-X3. Texto circular usando `<textPath>etextLength`

Exemplo 7-X3. Controlar o comprimento do texto para texto em um caminho fechado

```
<svg xmlns="http://www.w3.org/2000/svg" xml:lang="en"
xmlns:xlink="http://www.w3.org/1999/xlink" viewBox="0 0 500 500">
<title>Caminho de texto circular </title>
<style>
  text {
    font-size: 76px;
    font-family: Copperplate, Copperplate Gothic, serif;
    font-weight: bold;
    font-variant: small-caps;
    font-variant-ligatures: none;
    stroke: black;
    stroke-width: 1.5px;
    fill: white;
  }
</style>
<defs>
  <path id="textcircle"
    d="M250,400
      a150,150 0 0,1 0,-300a150,150 0 0,1 0,300Z"
    transform="rotate(12,250,250)"/>
</defs>
```

```

</defs>
<rect width="100%" height="100%" fill="silver" />
<text>
  <textPath xlink:href="#textcircle"
    aria-label="All for One &amp; One for All"
    textLength="942">&#160;um para todos
    para um &amp;</textPath>
</text>
</svg>

```

O círculo é criado com um `<path>` elemento contendo dois arcos de semicírculo de 150 unidades de raio. Como mencionamos no [Capítulo 6](#), você não pode desenhar um círculo completo com um único segmento de arco, porque ele não tem informações de posicionamento suficientes. O `textLength` atributo restringe o texto para coincidir com a circunferência do círculo (2π vezes o raio) e um rotacional `transform` é usado para ajustar o alinhamento de forma que o e comercial (&) fique centralizado na base do círculo.

Aviso Ou, pelo menos, esse é o resultado final. Criar esse layout foi mais difícil do que o esperado, devido a problemas com `textLength` elementos aninhados, `textLength` combinação com texto centralizado e tratamento inconsistente de espaços em branco. O código que parecia bom em um navegador estava terrivelmente fora de alinhamento em outros. A marcação no [Exemplo 7-X3](#) foi o resultado de muitos ajustes e testes em diferentes navegadores e ainda não está perfeitamente alinhada em todos os lugares.

Como mencionamos em ["Além da horizontal: texto girado e vertical"](#), `textLength` declara o comprimento total que você deseja que o texto tenha e os navegadores ajustam o espaçamento das letras para que se encaixe. Ele *deve* funcionar em qualquer elemento de texto, incluindo `<textPath>`, mas as implementações do navegador apresentam erros de maneiras conflitantes e alguns irão ignorá-lo. `textLength` só pode ser usado de forma confiável para `<text>` elementos sem filhos.

Ao ajustar o `textLength` para corresponder ao caminho, você precisará saber o comprimento do caminho que está usando. O navegador pode calculá-lo para você com o `getTotalLength()` método no `<path>` elemento. O valor retornado por esse método pode ser usado diretamente no `textLength` atributo de a `<textPath>` para fazer com que o texto se ajuste exatamente.

Por exemplo, abrindo o SVG do [Exemplo 7-X3](#) em um navegador e, em seguida, abrindo o console do desenvolvedor, você pode encontrar o comprimento do caminho da seguinte maneira:

```

document.getElementById("textcircle").getTotalLength()
//returns 942.61083984375

```

Isso dá essencialmente o mesmo resultado que calculá-lo a partir da geometria do círculo:

```

150 * 2 * Math.PI
//returns 942.4777960769379

```

Não é uma combinação perfeita, porque o navegador usa aproximações para desenhar caminhos circulares e aproximações para calcular o comprimento dos caminhos. Mas está perto o suficiente para o desenho. E `getTotalLength()` é inestimável para caminhos mais complicados, cujos comprimentos não podem ser convertidos em uma fórmula simples.

No SVG 2 e nas versões mais recentes dos navegadores Chromium, `getTotalLength()` pode ser usado em qualquer elemento de forma, não apenas a `<path>`. Portanto, você pode usá-lo em seu console do Chrome, mesmo que não possa contar com ele em seus scripts. Mas isso não será útil para caminhos de texto até que formas sem caminho possam ser referenciadas pelo `<textPath>` elemento.

Existem algumas outras coisas a serem observadas no [Exemplo 7-X3](#):

- Um `aria-label` atributo substitui o conteúdo de texto de `<textPath>` por uma versão mais amigável para o leitor de tela.
- A `&` entidade é usada para um E comercial (&), enquanto a entidade numérica ` ` é usada para um espaço sem quebra, para evitar que o espaço no início do texto seja reduzido. Como este é um arquivo SVG independente, a entidade HTML equivalente ` ` não está disponível.
- A `font-variant-ligatures` propriedade é usada para desativar ligaduras comuns, que podem causar uma aparência grossa quando o texto é posicionado em curvas fechadas.

Se o caminho for *multicurvo*, o texto pode parecer muito "apertado" na parte interna de uma curva ou muito "solto" na parte externa. Você pode ajustar a relação vertical entre o texto e seu caminho, modificando o valor de um `dy` atributo para o `<text>` elemento pai (ou um `<tspan>` elemento filho). Um valor positivo puxará a linha de base do texto para baixo para ficar mais centralizado no caminho, afrouxando as curvas internas estreitas e estreitando as externas.

Um dia, pode ser possível usar as propriedades relacionadas à linha de base do CSS para ajustar a posição do texto em relação ao caminho, de uma maneira mais intuitiva. Por enquanto, `dy` tem o melhor suporte ao navegador.

Este site é criado e mantido por Amelia Bellamy-Royds, usando material criado pelos três autores do livro.

Visualize o [repositório GitHub](#) para baixar os arquivos, sugerir uma correção ou ver quando uma página foi atualizada pela última vez.

CONTACT



Brunna Croches

Developer Full Stack



brunnacroches.dev



linkedin.com/brunnacroches



github.com/brunnacroches



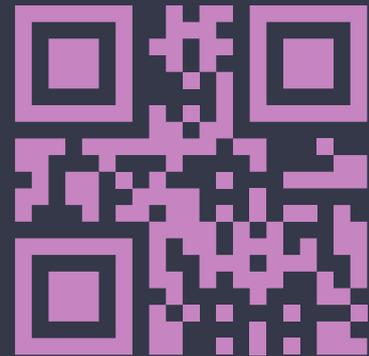
@brunnacroches.dev



discord.com/brunnacroches



brunnacroches@gmail.com



let's share