

- Back-End (<https://imasters.com.br/back-end>)
- Mobile (<https://imasters.com.br/mobile>)
- Front End (<https://imasters.com.br/front-end>)
- DevSecOps (<https://imasters.com.br/devsecops>)
- Design & UX (<https://imasters.com.br/design-ux>)
- Data (<https://imasters.com.br/data>)
- APIs e Microsserviços (<https://imasters.com.br/apis-microsservicos>)
- IoT e Makers (<https://imasters.com.br/iot-makers>)

ACESSIBILIDADE

PATROCINADOR:

5 JAN, 2016

Melhorando a acessibilidade no canvas

100 visualizações

 (<https://www.facebook.com/sharer?u=https://imasters.com.br/acesibilidade/melhorando-acesibilidade-no-canvas>)

 (<https://twitter.com/share?url=https://imasters.com.br/acesibilidade/melhorando-acesibilidade-no-canvas>)

 (<https://www.linkedin.com/shareArticle?url=https://imasters.com.br/acesibilidade/melhorando-acesibilidade-no-canvas>)

COMPARTILHE!

REINALDO FERRAZ
([HTTPS://IMASTERS.COM.BR/PERFIL/REINALDONIC-BR](https://imasters.com.br/perfil/reinaldonic-br))
Tem 34 artigos publicados com 37433 visualizações desde 2012



REINALDO FERRAZ ([HTTPS://IMASTERS.COM.BR/PERFIL/REINALDONIC-BR](https://imasters.com.br/perfil/reinaldonic-br))

34 

Formado em desenho e computação gráfica e pós graduado em design de hipermídia pela Universidade Anhembi Morumbi em São Paulo. Trabalha com desenvolvimento web desde 1998. Coordena as iniciativas de acessibilidade na Web do NIC.br e projetos relacionados a Open Web Platform, Digital Publishing e Web das Coisas. Representante do NIC.br em grupos de trabalho do W3C internacional em Acessibilidade na Web, Digital Publishing e Web das Coisas. Apaixonado por acessibilidade, usabilidade, padrões web, HTML, CSS e café sem açúcar.

LEIA MAIS ([HTTPS://IMASTERS.COM.BR/PERFIL/REINALDONIC-BR](https://imasters.com.br/perfil/reinaldonic-br))

26 NOV, 2018

Thing description: um manifesto para aplicações de Internet das Coisas (<https://imasters.com.br/iot-makers/thing-description-um-manifesto-para-aplicacoes-de-internet-das-coisas>)

9 OUT, 2018

Recursos de acessibilidade na Web para designers, desenvolvedores e produtores de conteúdo da WAI (<https://imasters.com.br/acesibilidade/recursos-de-acesibilidade-na-web-para-designers-desenvolvedores-e-produtores-de-conteudo-da-wai>)

10 ABR, 2018

Acessibilidade vai muito além de marcações semânticas no HTML. É possível garantir que aplicações dentro de elementos <canvas> transmitam informação a usuários de tecnologia assistiva ou até permitam o controle de uma aplicação complexa. O objetivo deste texto é mostrar como pequenas intervenções no código podem melhorar a acessibilidade de aplicações dentro do <canvas>.

O elemento <canvas> é um container que pode ser usado para renderizar gráficos de jogos, artes, ou outras imagens visuais em tempo real. Além da simples renderização de bitmaps, controles da aplicação atuam dentro desse container. E esses controles podem (e devem) ser acessíveis ao usuário.

Vamos usar como exemplo uma aplicação simples com <canvas> que permite desenhar uma estrela ao clicar no elemento. A princípio, sua marcação no HTML (excluindo os scripts para fazê-lo funcionar) funcionaria desta forma:

```
<canvas id="x" ></canvas>
```

Uma das principais diretrizes de acessibilidade é possibilitar o acesso via teclado ao conteúdo da página. Sendo assim, um texto descrevendo o que a aplicação vai executar dentro de um elemento de âncora (que pode executar a mesma função do clique do mouse) pode melhorar a acessibilidade desse elemento da seguinte forma:

```
<canvas id="x" >
<a href="#"> Desenho de uma estrela amarela feita ao clicar no elemento </a>
</canvas>
```

A acessibilidade no <canvas> também pode explorar muito mais do que uma simples descrição da aplicação. É possível fazer com que a aplicação seja controlada por ações baseadas na navegação por teclado.

No segundo exemplo, consideramos o desenvolvimento de um jogo ou aplicação no qual o usuário deve mover objetos no <canvas>. Todo esse controle pode ser feito na camada por trás do meu container, possibilitando a navegação via teclado e especialmente o acesso à tecnologia assistiva.

```
<canvas id="x" >
<h2>Controles</h2>
<a href="#" id="up">Para cima</a>
< a href="#" id="down">Para baixo</a>
</canvas>
```

Dessa forma, os controles da aplicação não ficam visíveis para o usuário, mas estão disponíveis para a navegação via teclado. A aplicação continua sendo acessível por mouse/toque no container, e também via teclado/movimentos.

Podemos também fazer com que conteúdo atualizado dentro do <canvas> seja exposto ao usuário de tecnologia assistiva. Nesse caso, utilizamos WAI-ARIA para garantir que as atualizações de status sejam informadas ao usuário, mesmo que seja fora do <canvas>, como no exemplo a seguir:

```
<canvas id="x" >
<h2>Controles</h2>
<a href="#" id="up">Para cima</a>
< a href="#" id="down">Para baixo</a>
</canvas>
```

```
<p aria-live="polite">Posição: movendo para <span id="position">cima</span></p>
```

Assim, quando ocorrer uma atualização de status, o usuário de tecnologia assistiva será informado imediatamente, e poderá continuar utilizando a aplicação devidamente informado sobre as notificações anteriores.

O ponto mais importante dos exemplos mostrados é a simplicidade da sua implementação em aplicações mais complexas e elaboradas. Vale lembrar que garantir acessibilidade a uma aplicação vai muito além de filantropia. A Web está evoluindo da mesma forma que nossos sentidos, e não podemos ser impedidos de acessar um conteúdo por não utilizar um computador de última geração ou por alguma limitação física. Não precisamos de muito para fazer a Web ser inclusiva. Precisamos, antes de qualquer conhecimento técnico, de vontade de fazer a diferença.

