

RESPONSIVIDADE

É fundamental que o site que você criou seja responsivo da maneira correta. Por exemplo, atualmente sites podem ser visualizados em Tv's, celulares, Desktops e afins. Os conteúdos que mais 'sofrem' para essa adaptação são as imagens, uma vez que, uma televisão pode suportar um tamanho de imagem enorme comparado à um celular. Nessa situação você se vê obrigado a reduzir ela através do CSS, porém, essa prática não fara a redução em bytes, somente visual, e isto tornará seu site lento.



CUIDADO! Sites lentos diminuem a **taxa de retenção** dos usuários, que ficam menos tempo acessando e podem prejudicar a indexação da sua página em mecanismos de busca como o **Google**. Veja no vídeo a seguir, um especialista em SEO (otimização para mecanismos de busca) falando sobre lentidão de sites, principalmente via 3G e 4G do celular.

Portal SEO: <https://youtu.be/jWnMfvSdo1E>

IMAGENS FLEXÍVEIS



Nosso primeiro passo no caminho de adaptar nosso conteúdo ao tamanho da tela vai ser aprender a gerar imagens de tamanho diferentes e a fazer o navegador carregar a imagem certa para cada situação. Para isso, devemos conhecer as tags `<picture>` e `<source>`.

Para esse exemplo, criamos as três imagens ao lado: a menor tem 300x300px, a média tem 700x700px e a maior tem 1000x1000px. Usamos o programa **GIMP**, que é um editor de imagens 100% gratuito. Essas imagens serão carregadas pelo navegador de acordo com o tamanho da janela atual. Para isso, criamos o seguinte código base:

```
<picture>
  
</picture>
```

Nota-se que a tag **** foi inserida normalmente dentro de **<Picture>**, a diferença agora é que essa nova tag abrigará todas as outras imagens, sendo carregado por padrão a foto-g (1000x1000px).

Quando a tela do navegador chegar perto dos 1000px a imagem não caberá mais lá, e aí que vem a próxima solução o **<source>**

```
<picture>
  <source media="(max-width: 1050px)" srcset="foto-m.png" type="image/png">
  
</picture>
```

A tag `<source>` é usada para especificar vários recursos de mídia para elementos de mídia como: **<video>** **<audio>** **<picture>** [Saiba tudo sobre a tag source](#).

Note que a tag `<source>` possui três atributos:

- ▶ `type` vai indicar o *media type* da imagem que usamos (veja mais informações sobre *media types* no capítulo 10)
- ▶ `srcset` vai configurar o nome da imagem que será carregada quando o tamanho indicado for atingido
- ▶ `media` indica o tamanho máximo a ser considerado para carregar a imagem indicada no atributo `srcset`.

```
<picture>
  <source media="(max-width: 750px)" srcset="foto-p.png" type="image/png">
  <source media="(max-width: 1050px)" srcset="foto-m.png" type="image/png">
  
</picture>
```

É necessário respeitar uma ordem dentro da tag **<Picture>**, a última linha de código deve ser a `img` padrão e a ordem dos **<source>** dentro dela deve ser crescente.

AÚDIO

Colocar qualquer áudio em seu site se tornou simples graças ao HTML5, basta colocar a tag `<áudio>` e alguns `<source>` e pronto;

Definição e uso: A tag `<áudio>` suporta uma ou mais `<source>` com diferentes fontes de áudio. O navegador escolherá a primeira fonte compatível.

Os textos entre as tag `<áudio>` só serão exibidas caso o navegador não suporte nenhum dos arquivos dentro das `sources`.

Atributos

Autoplay: Especifica que o áudio começará a tocar assim que estiver pronto;

Controls: Especifica que os controles de áudio devem ser exibidos (como um botão de reprodução/pausa, etc.);

Loop: Especifica que o áudio recomeçará sempre que terminar;

Muted: Especifica que a saída de áudio deve ser silenciada;

Preload: Especifica se e como o autor acha que o áudio deve ser carregado quando a página for carregada;

Src: Especifica a URL do arquivo de áudio;

A tag também suporta [atributos globais](#) e [atributos de eventos](#)

```
<audio preload="metadata" controls autoplay loop>
  <source src="midia/guanacast-33.mp3" type="audio/mpeg">
  <source src="midia/guanacast-33.ogg" type="audio/ogg">
  <source src="midia/guanacast-33.wav" type="audio/wav">
  <p>Seu navegador não suporta áudio <a href="midia/guanacast-33.mp3"
    download="guanacast-33.mp3" type="audio/mpeg"></a></p>
</audio>
```

Vamos analisar os principais atributos da tag <audio> antes de mais nada:

- ▶ O atributo `preload` indica se o áudio será pré-carregado ou não e aceita três valores:
 - `metadata` vai carregar apenas as informações sobre o arquivo (tamanho, tempo, informações de direitos, etc)
 - `none` não vai carregar absolutamente nada até que o usuário clique no botão play ou um script inicie a reprodução
 - `auto` (padrão) vai carregar o arquivo de áudio inteiro assim que a página for carregada, mesmo que o usuário nunca aperte o play
- ▶ O atributo `controls` vai apresentar o player na tela. Caso não seja colocado na tag <audio>, o controle será transparente e o usuário não poderá interagir com ele.
- ▶ O atributo `autoplay`, quando inserido, vai iniciar a reprodução do áudio assim que a página for carregada.
- ▶ O atributo `loop` vai fazer com que o áudio seja repetido eternamente assim que terminar a sua reprodução.

Dentro da tag <audio>, adicionamos vários <source> com formatos diferentes do mesmo áudio. Coloque na parte de cima o seu formato favorito. Os demais só serão carregados caso o de cima falhe. Caso todos falhem, criamos um parágrafo que permite o download do arquivo MP3 para ouvir no player padrão do dispositivo.

VÍDEO

A tag vídeo funciona idealmente similar à tag áudio.

```
<video width="600" poster="thumb.jpg" controls autoplay>
  <source src="meu-video.webm" type="video/webm">
  <source src="meu-video.mp4" type="video/mp4">
  <source src="meu-video.ogv" type="video/ogg">
  <p>Infelizmente seu navegador não conseguiu carregar o vídeo.</p>
</video>
```

Antes de mais nada, vamos criar a tag <video> e configurar alguns atributos importantes:

- ▶ `width` vai indicar a largura que o vídeo vai ter na tela. Nesse exemplo, 600px.
- ▶ `poster` configura uma imagem que vai aparecer como uma capa, enquanto o visitante não aperta o play para reproduzir o vídeo
- ▶ `controls` vai configurar se os controles do vídeo vão aparecer na parte inferior da mídia. Por padrão, os controles não aparecerão, mas basta colocar a palavra `controls` na tag <video>.
- ▶ `autoplay` diz para o navegador se o vídeo vai começar a tocar automaticamente, assim que a página for carregada.

Atributos Opcionais

Attribute	Value	Description
<u>autoplay</u>	autoplay	Specifies that the video will start playing as soon as it is ready
<u>controls</u>	controls	Specifies that video controls should be displayed (such as a play/pause button etc).
<u>height</u>	<i>pixels</i>	Sets the height of the video player
<u>loop</u>	loop	Specifies that the video will start over again, every time it is finished
<u>muted</u>	muted	Specifies that the audio output of the video should be muted
<u>poster</u>	URL	Specifies an image to be shown while the video is downloading, or until the user hits the play button
<u>preload</u>	auto metadata none	Specifies if and how the author thinks the video should be loaded when the page loads
<u>src</u>	URL	Specifies the URL of the video file
<u>width</u>	<i>pixels</i>	Sets the width of the video player

Vale ressaltar que a tag aceita os [atributos globais](#) e [atributos de eventos](#). Para saber tudo sobre a tag <vídeo> acesse o [documento referencial](#).

Diferente dos arquivos de imagem e áudio os arquivos de vídeo são um pouco mais complexos no seu formato. Os formatos suportados são MPEG, WEBM e OGG, sendo os dois primeiros os mais recomendáveis e mais compatíveis.

Tome muito cuidado ao optar por hospedar algum vídeo em seu próprio site, isto porque pode ser muito perigoso. Isso pode deixar seu site lento e consumir muito do seu tráfego contratado com a hospedagem.

Vamos fazer uma conta simples: um vídeo simples, com poucos minutos, em formato mp4 com codec padrão deve ocupar uns 150MB com facilidade. Agora imagine que você lance seu site e 200 visitantes (um número super possível) acessem seu site e reproduzam o vídeo. Pronto, você acabou de utilizar 29GB de tráfego! Imagine o quanto isso pode deixar seu site lento ou consumir sua franquia de hospedagem, caso não seja ilimitada.



Euuuuuu! Ou melhor, o **YouTube** ou o **Vimeo**! Esses são serviços para a hospedagem de vídeos que vai evitar consumir nossos próprios recursos de host contratado. Cada um tem suas vantagens e desvantagens:



O **YouTube** é o serviço de hospedagem de vídeos mais popular do mundo e é gerenciado pelo Google. Sua principal vantagem é que seus servidores são ultra rápidos. Por outro lado, a ideia do Google é deixar todos os vídeos públicos e disponíveis, o que pode ser uma dor de cabeça caso você queira limitar quem vai ter acesso a determinado vídeo (uma escola online, por exemplo, onde queremos que apenas os alunos matriculados possam assistir).

O **Vimeo** resolve o problema que apontamos anteriormente. Ele permite limitar quem vai poder ver o vídeo, o que é especialmente vantajoso para quem quer criar produtos em forma de vídeo, entregues por demanda dentro de um site personalizado. Como desvantagens desse serviço, ele é pago por uma taxa anual e seus algoritmos não são tão eficientes quanto os do YouTube, logo os vídeos apresentam pequenos travamentos às vezes.



Esses dois sites lhe permitem incorporar via HTML seu conteúdo em seu site. Na parte 'compartilhar' tem a opção incorporar, basta selecioná-la e copiar o código que lá aparece, então, por fim, colocar entre sua tag <vídeo>

