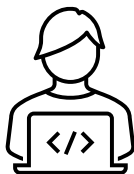


Prof. Lucas Teixeira

Repositório de aulas:
bit.ly/4bHLaad



Front-End



Misturando as Coisas

Podemos também criar listas mistas, configurando listas dentro de outras listas.

```
<h2>Minhas linguagens</h2>
<ol>
  <li>Antigas</li>
  <ol type="a">
    <li>Clipper</li>
    <li>Visual Basic</li>
    <li>Fortran</li>
    <li>Delphi</li>
  </ol>
  <li>Novas</li>
  <ol type="a" start="5">
    <li>PHP</li>
    <li>Python</li>
    <li>Javascript</li>
    <li>Kotlin</li>
  </ol>
</ol>
```

Lista de definições

É como se fosse um dicionário, temos os termos e as suas descrições. É uma lista sem demarcadores, mas bem útil em alguns casos.

Toda lista de definições está dentro de uma tag `<dl> </dl>` (definition list).

Cada termo é um `<dt>` (definition term) e cada descrição é um `<dd>` (definition description).

Assim como os itens da lista, essas duas últimas tags possuem fechamento opcional, segundo a referência oficial da HTML5.

Exemplo Prático

```
<h2>Lista de definição</h2>
<dl>
  <dt>HTML</dt>
  <dd>Linguagem de marcação para a criação de conteúdo de um site</dd>
  <dt>CSS</dt>
  <dd>Linguagem de marcação para a criação do design de um site.</dd>
  <dt>Javascript</dt>
  <dd>Linguagem de programação para a criação de interatividade de um site.</dd>
</dl>
```

Links

Os hyperlinks são um dos conceitos mais antigos da história da linguagem HTML. Eles permitem que você ligue um ponto a outro na World Wide Web. Toda vez que você está acessando um site e clica em um local para ir para outra página, outro site ou até para baixar um arquivo, você está interagindo com um hyperlink



Links

Até os mecanismos de busca se utilizam dos hyperlinks de um site. O Google, por exemplo, para achar um determinado site, fica vasculhando constantemente todos os outros sites da Internet procurando por links para descobrir novos conteúdos. Por isso é tão importante conseguir links válidos de outros sites para o nosso próprio site.



Links

Para criar um hyperlink, devemos criar âncoras através da tag `<a>`. O principal atributo dessa tag é o **href**, que cria uma referência hipertexto. Vamos ver um exemplo simples:



```
<h2>Links</h2>  
<a href="https://google.com">Clique Aqui</a>
```

Note que dentro do atributo **href**, o que colocamos foi uma URL completa para outro site.

Links

Outro atributo bem útil da tag de âncora é o **hreflang**, que permite indicar qual é o idioma principal do site para onde o link está desviando o fluxo de navegação. Isso vai permitir avisar ao navegador e a softwares de tradução como lidar caso o visitante opte por traduzir automaticamente os conteúdos.

```
<a href="https://www.w3schools.com/html/" hreflang="en">  
  Site da W3schools (Em Inglês)  
</a>
```



Tipos de links

Por padrão, sempre que um visitante clique em um hyperlink, o site de destino abre na mesma janela do site que continha esse link. Ou seja, o conteúdo anterior vai deixar de ser exibido para mostrar o novo conteúdo.

Esse é um comportamento desejado quando o visitante vai continuar a visitar o nosso site, apenas mudando de um documento para outro. Mas e quando um clique leva o visitante para outro site e provavelmente ele nunca mais voltará ao nosso?



Tipos de links

Para poder controlar onde o site de destino vai abrir, podemos usar o atributo target, que suporta os seguintes valores:

- **_blank** vai abrir o link em uma nova janela em branco
- **_self** vai abrir o link na janela ou frame atual (padrão)
- **_top** vai desfazer todos os frames e abrir o destino no navegador completo
- **_parent** similar ao uso do _top em uma referência à janela mãe
- **nome-do-frame** caso esteja usando frames, indicar o nome da janela a abrir



Tipos de links

Como o uso de frames é uma técnica quase em desuso, vamos nos basear apenas nas duas primeiras opções `_blank` e `_self`.

```
<a href="pagina2.html" target="_self">  
    Continuar navegando no site  
</a>
```

```
<a href="https://google.com" target="_blank">  
    Acessar o google em uma nova janela  
</a>
```

Esse link é seu ou dos outros?

Existe um recurso bem interessante para links que é indicar qual é a natureza do destino usando o atributo rel. Esse atributo aceita vários valores, entre eles vou citar:

- **next** indica que o link é para a próxima parte do documento atual
- **prev** indica que o link é para a parte anterior do documento atual
- **author** indica que é um link para o site do autor do artigo atual
- **external** indica que é um link para outro site que não faz parte do site atual
- **nofollow** indica que é um link para um site não endossado, como um link pago

Esse link é seu ou dos outros?

```
<a href="pagina2.html" target="_self" rel="next">  
    Continuar navegando no site  
</a>
```

```
<a href="https://google.com" target="_blank" rel="external">  
    Acessar o google em uma nova janela  
</a>
```

Esse link é seu ou dos outros?

No código anterior, o primeiro link é o que chamamos de link local ou link interno, já que ele leva o visitante a outra página dentro do nosso próprio site. Note que não é necessário nem indicar a URL completa nesses casos.

Já o segundo link vai nos levar para um outro site, o que chamamos de link externo. Nestes casos, devemos indicar a URL completa, incluindo o protocolo **http://** ou **https://** e o caminho que leve à uma página específica, se for necessário.

Links Para Downloads

Outra coisa que aparece bastante em sites são os links para efetuar download de algum material em PDF, ou de um arquivo ZIP qualquer. A partir da versão HTML5, as âncoras receberam atributos especiais para isso. Basta fazer o link diretamente para o arquivo que se deseja efetuar o download e adicionar o atributo download com o valor configurado para o nome do arquivo a ser baixado e o atributo **type** para indicar ao navegador que tipo de arquivo está sendo baixado. Vamos ver um exemplo:



Links Para Downloads

```
<a href="livro.pdf" download="livro.pdf" type="application/pdf">  
  Baixe aqui o livro em PDF  
</a>
```


Links Para Downloads

Aqui vão alguns **media types** bem usados no nosso dia-a-dia:

- application/zip
 - text/html
 - text/css
 - text/javascript
 - video/mp4
 - video/H264
 - video/JPEG
 - audio/aac
 - audio/mpeg
 - font/ttf
 - image/jpeg
 - image/png
- 

Imagens Dinâmicas, Áudio e Vídeo

Fotos, áudios e vídeos são essenciais para a construção dos sites hoje em dia. Ilustrar bem o seu conteúdo é imprescindível para deixar clara a mensagem que queremos passar. Mas é preciso tomar cuidado com formatos e tamanhos, pois muita gente abrirá os sites em seus celulares e dispositivos móveis. Vamos falar sobre esse assunto no capítulo que iniciamos agora.

Imagens Dinâmicas, Áudio e Vídeo

Fotos, áudios e vídeos são essenciais para a construção dos sites hoje em dia. Ilustrar bem o seu conteúdo é imprescindível para deixar clara a mensagem que queremos passar. Mas é preciso tomar cuidado com formatos e tamanhos, pois muita gente abrirá os sites em seus celulares e dispositivos móveis. Vamos falar sobre esse assunto no capítulo que iniciamos agora.

Imagens Adaptáveis

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>`.

Imagens Adaptáveis

Para esse exemplo, criamos as três imagens ao lado: a menor tem 300x300px, a média tem 700x700px e a maior tem 1000x1000px.

Essas imagens serão carregadas pelo navegador de acordo com o tamanho da janela atual. Para isso, criamos o seguinte código base:

canva.com

Imagens Adaptáveis

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

Imagens Adaptáveis

Note que colocamos a tag `` exatamente como aprendemos. A novidade aqui é que inserimos essa imagem dentro da tag `<picture>`, que vai concentrar as outras fontes de imagem. Por padrão, a imagem foto.png (1000x1000px) será carregada.

Imagens Adaptáveis

O problema vai começar a surgir quando a janela do navegador chegar perto dos 1000 pixels de largura, pois a foto não vai mais caber lá. Vamos agora adicionar uma linha para resolver esse problema:

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


Imagens Adaptáveis

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

- **type** vai indicar o media type da imagem que usamos.
- **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.

Imagens Adaptáveis

Vamos continuar e acrescentar mais um `<source>` à nossa imagem:

```
<picture>
  <source media="(max-width: 750px)" srcset="foto2.png" type="image/png">
  <source media="(max-width: 1050px)" srcset="foto.png" type="image/png">
  
</picture>
```

Imagens Adaptáveis

É importante que existe uma ordem entre os `<source>`, e nessa nossa configuração, os itens mais acima sejam os menores tamanhos para `max-width` e que os seguintes sejam maiores, de forma crescente. O último item dentro de `<picture>` deve ser a imagem padrão.

Áudios no HTML

O HTML5 trouxe diversas novidades para a web, e entre elas está a tag **<audio>**, que permite incorporar e reproduzir arquivos de áudio em páginas HTML de forma simples e direta. Nesta aula, você aprenderá tudo o que precisa saber para usar essa tag com maestria e criar experiências sonoras incríveis em seus sites.

A Tag `<audio>`

A tag `<audio>` é responsável por inserir um player de áudio na página. Ela possui diversos atributos que permitem controlar a reprodução do áudio, como:

- **src:** Define o caminho do arquivo de áudio a ser reproduzido.
- **controls:** Exibe os controles de reprodução padrão do navegador (play/pause, volume, etc.).
- **autoplay:** Inicia a reprodução automaticamente quando a página carregar.

A Tag `<audio>`

- **loop**: Reproduz o áudio em loop, sem parar.
- **muted**: Inicia o áudio mudo.
- **preload**: Define como o áudio deve ser pré-carregado (nenhum, metadados ou automático).

Exemplo de Uso

```
<audio src="audio.mp3" controls>  
</audio>
```

Especificando Formatos de Áudio

É importante especificar os formatos de áudio compatíveis com o navegador, utilizando a tag `<source>`. Isso garante que o áudio seja reproduzido em diferentes navegadores, mesmo que nem todos suportem o mesmo formato.

```
<audio controls>
  <source src="meu_audio.ogg" type="audio/ogg">
  <source src="meu_audio.mp3" type="audio/mpeg">
  Seu navegador não suporta a tag `audio`.
</audio>
```


Vídeos em HTML

O HTML5 introduziu a tag `<video>`, que permite incorporar e reproduzir vídeos em páginas HTML de maneira simples e direta. Seja para fins informativos, educativos ou de entretenimento, os vídeos enriquecem a experiência do usuário e tornam seus sites mais dinâmicos e atraentes. Nesta aula completa, você aprenderá tudo o que precisa saber para usar essa tag com maestria e adicionar vídeos incríveis em seus projetos web.

A Tag `<video>`

A tag `<video>` é responsável por inserir um player de vídeo na página. Ela possui diversos atributos que controlam a reprodução do vídeo, como:

- **src:** Define o caminho do arquivo de vídeo a ser reproduzido.
- **controls:** Exibe os controles de reprodução padrão do navegador (play/pause, volume, tela cheia, etc.).
- **autoplay:** Inicia a reprodução automaticamente quando a página carregar.

A Tag **<video>**

- **loop**: Reproduz o vídeo em loop, sem parar.
- **muted**: Inicia o vídeo mudo.
- **poster**: Define uma imagem a ser exibida enquanto o vídeo carrega ou está pausado.
- **width**: Define a largura do player de vídeo.
- **height**: Define a altura do player de vídeo.

Exemplo de uso

```
<video src="meu_video.mp4" controls>  
  Seu navegador não suporta a tag `video`.  
</video>
```

Especificando Formatos de Vídeo

Assim como no áudio, é importante especificar os formatos de vídeo compatíveis com o navegador, utilizando a tag `<source>`. Isso garante que o vídeo seja reproduzido em diferentes navegadores, mesmo que nem todos suportem o mesmo formato.

```
<video controls>
  <source src="meu_video.mp4" type="video/mp4">
  <source src="meu_video.ogv" type="video/ogg">
  Seu navegador não suporta a tag 'video'.
</video>
```

Especificando Formatos de Vídeo

Formatos suportados por navegadores:

Navegador	Arquivos compatíveis
Microsoft Edge	.mp4 .m4v
Apple Safari	.mp4 .m4v
Google Chrome	.mp4 .m4v .webm .ogv
Mozilla Firefox	.webm .ogv
Opera	.webm .ogv

Hospedar seus próprios vídeos pode ser caro

Quando colocamos vídeos no nosso próprio servidor, podemos passar por problemas com alto consumo de banda, site lento e incompatibilidades com alguns navegadores por conta dos codecs. E geralmente só percebemos esses problemas quando colocamos nosso projeto no ar e lançamos oficialmente.

Hospedar seus próprios vídeos pode ser caro

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.

Hospedar seus próprios vídeos pode ser caro

E é claro que seu site possivelmente não vai ter apenas um vídeo, não vai ter apenas 200 visitantes em um dia e aí a conta só aumenta. É preciso tomar muito cuidado quando decidimos guardar nossos próprios vídeos.

E agora? Como resolver esse problema?

Você pode usar serviços especificamente para hospedar os seus vídeos como por exemplo o **YouTube** ou o **Vimeo**. Estes serviços evitam consumir nossos próprios recursos de host contratado. Cada um tem suas vantagens e desvantagens:

YouTube

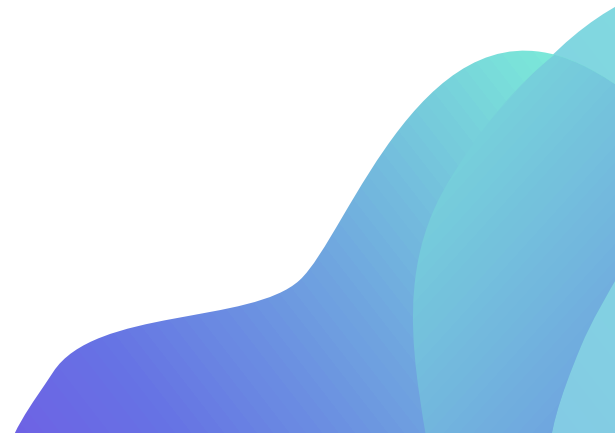


O YouTube é o serviço de hospedagem de vídeos mais popular do mundo e é gerenciado pelo Google. Sua principal vantagem é que seus servidores são ultra rápidos. Por outro lado, a ideia do Google é deixar todos os vídeos públicos e disponíveis, o que pode ser uma dor de cabeça caso você queira limitar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

Vimeo



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.



Incorporando Vídeos no Seu Site

Para incorporar vídeos que você subiu no YouTube ou Vimeo, existem recursos que te dão o código pronto em HTML5. No YouTube, abra o vídeo que você quer incorporar e clique no link COMPARTILHAR que fica abaixo do título (veja imagem a seguir).



Incorporando Vídeos no Seu Site

Ao clicar no link indicado anteriormente, uma janela vai aparecer, te dando as opções de compartilhamento. Escolha o item INCORPORAR.



Incorporando Vídeos no Seu Site

O código HTML personalizado vai aparecer em uma nova janela de contexto, incluindo um botão que permite **COPIAR** o código com a tag do `<iframe>` que vai aparecer diretamente na sua página. Volte ao seu editor de código e cole a tag no seu arquivo HTML que vai apresentar o vídeo.

Vídeo incorporado

```
<iframe width="560" height="315"
src="https://www.youtube.com/embed/
zYQsSWgootw?si=s52u0la7raedBWjk"
title="YouTube video player"
frameborder="0"
allow="accelerometer; autoplay;
clipboard-write; encrypted-media;
gyroscope; picture-in-picture; web-
share" referrerpolicy="strict-
origin-when-cross-origin">
```

Copiar

Imagens Como Links

images.google.com.br

A tag `<a>` utilizada para a criação de links também pode ser utilizada entorno de imagem.

Exemplo:

```
<a href="https://google.com" target="_blank">  
    
</a>
```


Listas Como Links

A tag `<a>` utilizada para a criação de links também pode ser utilizada entorno de itens de uma lista.
Exemplo:

```
<ol>  
  <a href="https://google.com">  
    <li>Google</li>  
  </a>  
</ol>
```

Bibliografia

Online

Mozilla: <https://developer.mozilla.org/pt-BR/docs/Web/HTML>

W3C Schools:
<https://www.w3schools.com/html/>

Livros

Html 5: Entendendo e Executando:

<https://a.co/d/9o3XsHS>

CSS Cookbook: <https://a.co/d/6eU6USL>

JavaScript: O Guia Definitivo: <https://a.co/d/342B0rJ>

HTML5 e CSS3: Guia Prático e Visual:

<https://a.co/d/jg3ccdP>

Use a Cabeça! HTML e CSS: <https://a.co/d/66caZxW>

A psicologia das cores: <https://a.co/d/1jVHsnO>



Por Hoje é Só

Continuamos na próxima aula

Repositório de aulas:
<https://bit.ly/4bHLaad>