DevWeb

Capítulo 22

Inline Frames

Aposto contigo uma bananada que mesmo já conhecendo HTML há anos, provavelmente você não sabia que a tag `<iframe>` significava *inline frame* ou, no bom e velho Português, algo como *"quadro em linha"*. Nesse capítulo, vamos nos aprofundar um pouco nos *iframes*, que já apareceram aqui no curso durante o capítulo 11, enquanto nós falávamos sobre inserção de vídeos vindos de serviços como YouTube e Vimeo.

Entendendo quadros em linha

Antes de mais nada, é bom deixar claro que essa tradução porca que levou um `iframe` a ser chamado de *quadro em linha* na sua versão brazuca não faz parte da referência oficial do W3C. Esse termo traduzido para o idioma tupiniquim veio única e exclusivamente da minha vontade em fazer você lembrar que aquele "i" vem de *inline* e não tem nenhuma relação a produtos da **Apple** como *iPod*, *iPhone* e *iPad* (juro que um aluno já me perguntou se havia alguma ligação).

Um `iframe` é basicamente uma "janela aberta" dentro da nossa página para apresentar o conteúdo de outras páginas. Esse é um recurso bem antigo, presente desde as versões anteriores da HTML, mas que até hoje são utilizados por alguns sites, o que causa muita discussão em relação a segurança. Vamos abordar esse assunto mais pra frente, mas antes vamos aprender a usar esse recurso.

NÓS JÁ VIMOS ISSO ANTES: No **capítulo 11** aprendemos como incorporar um vídeo do **YouTube** ou **Vimeo** a uma página. Basta voltar lá e constatar que o código fornecido pelo serviço é uma tag `<iframe>` com vários parâmetros pré-configurados e que "magicamente" apresentava um vídeo dentro do nosso HTML. Pois saiba que podemos usar os quadros para apresentar qualquer tipo de conteúdo que esteja em outro site de forma simples.

Como criar um iframe?

Vamos começar inserindo um quadro bem simples:

```
<h1>Teste de iframe</h1>
<p>
    Acesse o site
    <iframe src="https://www.cursoemvideo.com"></iframe>
    e faça muitos cursos!
</p>
```

O código acima cria um título e um parágrafo com um `iframe` no meio da frase. Analisando o resultado, percebemos o motivo do uso do *"i"* antes da palavra *"frame"* no elemento. Ele cria um bloco *inline* (na mesma linha) que não quebra a frase, mesmo que o elemento tenha uma altura muito maior que o texto. Veja só a imagem a seguir:

Teste de iframe

Acesse o site 

e faça muitos cursos!

O site do **CursoemVideo** foi aberto dentro da nossa página, permitindo inclusive que o visitante faça a rolagem pelo conteúdo.

Podemos ainda configurar detalhes visuais e comportamentais como o tamanho do *frame,* sua borda, a forma de rolagem e até mesmo tratar eventuais incompatibilidades com alguns navegadores:

```
<p>
    Acesse o site
    <iframe src="https://www.cursoemvideo.com"
    frameborder="0" width="500" height="500"
    scrolling="auto">
        [ Seu navegador não suporta iframes ]
    </iframe>
    e faça muitos cursos!
</p>
```

O código acima remove a borda 3D que é criada por padrão (podemos ainda personalizá-la com CSS mais pra frente) e também configura um *frame* quadrado, com 500 pixels de largura (`width`) e de altura (`height`).

A propriedade `scrolling` aceita três valores:

- `no` - não permite a rolagem da página, mesmo que ela seja maior que o tamanho estabelecido para o *frame*.
- `yes` - habilita a rolagem do conteúdo, mesmo que seu conteúdo não seja grande o suficiente para preencher o *frame*.
- `auto` - (padrão) a rolagem é habilitada apenas se o conteúdo for maior que o tamanho do quadro estabelecido.

Por fim, note que acrescentamos uma frase entre `<iframe>` e `</iframe>` para que seja exibida caso o navegador do usuário não suporte o uso desse tipo de recurso. Alguns celulares mais antigos possuem navegadores que bloqueiam a exibição de quadros por questões relacionadas ao tamanho da tela, uso de memória e até segurança.

Talvez seja até interessante adicionar um link `<a>` a essa mensagem de erro para que o usuário possa visitar a página caso seu navegador não seja compatível com `iframe`.

Páginas locais dentro do iframe

Claro que não precisamos apenas carregar sites externos dentro dos quadros que vão compor nosso site. Também podemos ter páginas locais carregadas no meio do nosso conteúdo.

```
<iframe src="outra-pagina.html" width="600" height="400">
    <p>Seu navegador não é compatível com a exibição desse recurso.</p>
</iframe>
```

É claro que para o elemento acima funcionar, devemos ter um arquivo chamado `outra-pagina.html` dentro da mesma pasta e o conteúdo será exibido em um quadro de dimensões `600x400`, conforme as configurações estabelecidas.

Podemos até ter mais de um `iframe` em uma mesma página, lembrando que por padrão eles serão exibidos no formato `inline-block` (isso pode até ser mudado via CSS). Apenas considere que cada quadro aberto dentro da página é semelhante a uma nova instância do navegador aberta. Sendo assim, sobrecarregar um site com muitos *iframes* pode deixar o dispositivo do seu visitante mais lento, consumindo um pouco mais de memória.

IFRAMES VÃO DEIXAR O COMPUTADOR MAIS LENTO? A melhor resposta para essa pergunta é: **DEPENDE**. Acabamos de ver que um `iframe` é como uma nova instância do navegador abrindo em um quadro. De forma grosseira, é como se abríssemos uma nova aba do navegador para ver o conteúdo. Abrir uma nova aba deixa seu computador muito mais lento? Pode ser que sim, pode ser que não. É a mesma linha de raciocínio quando usamos *quadros em linha*.

Gerando conteúdo dentro do iframe

No lugar de usar o parâmetro `src` para indicar a **origem** (*souce*) do conteúdo de um quadro, podemos usar o parâmetro `srcdoc` para criar um conteúdo simples estaticamente dentro do *iframe*.

```
<iframe srcdoc="<h2>Teste de título</h2><p>Esse documento foi gerado
dinamicamente</p><img src='cursoemvideo.png'>" frameborder="0">
    <p>Seu navegador não é compatível com a exibição desse recurso.</p>
</iframe>
```

A estrutura acima vai criar um quadro de tamanho padrão (`300x150`) e vai colocar um título, um parágrafo e uma imagem dentro dele. Não existe nenhum arquivo adicional ou site externo para carregar o conteúdo.

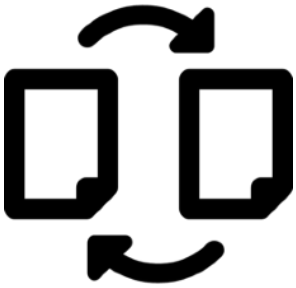
Personalizando um quadro via CSS

Já que todo `iframe` é uma caixa em linha (mais informações no **capítulo 16**), ele pode ser personalizado usando folhas de estilo sem problema algum. O princípio é exatamente o mesmo e o comportamento também.

```
iframe {
    width: 500px;
    height: 500px;
    border: 5px solid black;
    border-radius: 15px;
    box-shadow: 3px 3px 10px #0000006c;
}
```

UMA QUESTÃO DE PRIORIDADES: O código acima define o tamanho do `iframe` para uma resolução de 500x500 pixels. Se por acaso o código HTML tivesse definido para outro tamanho (800x600, por exemplo), qual configuração fica valendo? Pois saiba que as folhas de estilo **SEMPRE** vão sobrescrever toda e qualquer pré-configuração feita pela HTML. O estilo nesse caso é soberano. Lembre-se disso.

Direcionando novo conteúdo para um iframe



Podemos dar ordens a um quadro para trocar o seu conteúdo através de *links* ou códigos *JavaScript*. O primeiro passo é dar um nome a ele através do parâmetro `name`.

```
<iframe name="quadro" src="pagina01.html">
    <p>Seu navegador não é compatível
        com a exibição desse recurso.</p>
</iframe>
```

No código acima, o quadro terá `300x150` pixels de tamanho e vai exibir o conteúdo do arquivo `pagina01.html` (se for compatível, claro). A diferença aqui é que atribuímos um nome a ele, que agora se chama **quadro**.

A qualquer lugar do documento original (o que contém o *frame*), podemos criar um link da seguinte forma:

```
<a href="pagina02.html" target="quadro">Abrir segundo site</a>
```

O *link* que criamos vai abrir a `pagina02.html` dentro do *iframe* sempre que o usuário clicar sobre ele, substituindo o conteúdo anterior imediatamente.

Porém, nem tudo são flores...

Quando escolhemos usar *iframes* em nosso conteúdo, devemos estar cientes de que essa opção traz consequências. Muita gente diz que usar códigos de `<iframe>` é algo ultrapassado e não recomendado. Não compartilho da mesma opinião, apenas quero te deixar ciente do que pode acontecer, e por isso resolvi enumerar alguns tópicos e falar brevemente sobre eles.

Não confunda frames e iframes

Minha avó sempre dizia: *"- não confunda alhos com bugalhos"* e isso me fez aprender o que são bugalhos. Pesquise aí também se quiser aprender. A aula aqui é de *iframe*!

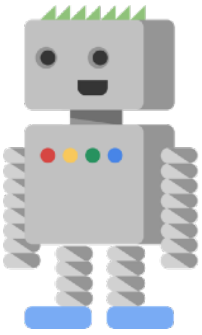
E sem fugir mais do assunto, muita gente confunde *iframes* com os antigos *frames* da versão 4 da HTML e por isso ficam dizendo que todos são ultrapassados. Antigamente, os sites usavam as tags `<frameset>` e `<frame>` para organizar o layout do site. Ainda hoje, alguns sites ainda os usam, mas para criar uma barra fixa superior enquanto mostra outro conteúdo na parte de baixo da tela.

Os *frames* não se tornaram aquelas *tags* ditas **obsoletas**, mas estão em um caminho acelerado para isso. Hoje existem técnicas mais modernas e convenientes para que não seja necessário utilizá-los.

Os *iframes* são mais modernos e muitos sites ainda os utilizam, como veremos mais adiante. Então não cometa o engano de confundir os dois ou colocá-los em um "mesmo saco".

Porém, como o próprio título dessa sessão sugere, nem tudo são flores e mesmo sendo modernos, optar por usar *iframes* pode trazer algumas consequências indesejadas em algumas situações.

Problemas com mecanismos de busca



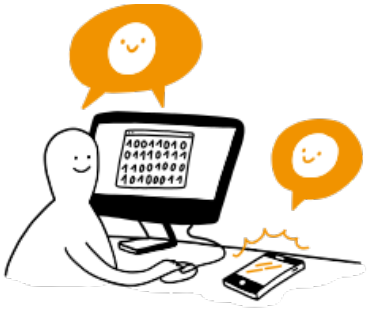
Quando criamos sites, uma das coisas que mais vamos querer é que ele apareça nos resultados de busca do **Google**, certo? Pois saiba que existe um algoritmo que varre o conteúdo dos sites - um robô batizado de *Googlebot* - que analisa todo o conteúdo de um HTML e indexa seu conteúdo.

UMA OLHADA NAS GUIDELINES: Ao acessar a documentação oficial do Google, encontramos referências a esse problema de indexação de conteúdo pelo robô. Dê uma breve conferida no link a seguir para mais informações.

Google Search Console Help: https://support.google.com/webmasters/answer/72746?hl=pt-br

Segundo o próprio **Google**, podem acontecer problemas com a indexação de conteúdos contidos dentro de *iframes* e talvez não haja uma boa relação entre o que está fora (na página principal) e o que está dentro (na parte interna do *iframe*).

A usabilidade mandou um abraço



Ao decidir usar *iframes* para apresentar conteúdos em nosso site, saiba que existem alguns incômodos relativos à usabilidade e acessibilidade desse conteúdo.

Para quem é *deficiente visual*, alguns leitores de tela simplesmente se perdem completamente quando o conteúdo está dentro de um quadro desse tipo. E além disso, alguns outros problemas pontuais:

- O botão de **voltar** para a página anterior no navegador fica meio perdido, sem saber se quer voltar no site externo ou dentro do *iframe*.
- Confunde os usuários quando um site que está dentro de um *iframe* tenta abrir uma página em uma nova janela.
- Se o site a ser aberto não for responsivo, provavelmente vai ficar visualmente quebrado ou esquisito dentro de um *iframe* na tela de um celular.

Não estou escrevendo tudo isso pra te desmotivar, mas também acho falta de consideração esconder esses inconvenientes de você. Os iframes são ótimos, muito práticos e fáceis de usar, mas podem dar dor de cabeça em alguns navegadores/dispositivos.

Cuidado com quem você chama pra sua casa

E por fim, mas não menos importante (na verdade a coisa mais importante de todas), a questão da **segurança** é muito discutida em relação ao uso de *iframes*.

Porém, em relação a esse assunto, tenho uma revelação importante a fazer: **em parte, isso pode ser responsabilidade SUA**! Isso porque devemos tomar todo o cuidado na hora de colocar um quadro no nosso site, abrindo um site de terceiros.

Você por acaso deixa aberta a porta da frente da sua casa pra quem quiser entrar e, eventualmente, beber um copo d'água? É uma atitude muito arriscada, não é? Pois esse conselho também serve para o seu site.

Ao usar *iframe* em projetos que vão efetivamente ao ar, devemos tomar o cuidado de dificultar as ações cruzadas e as relações entre o nosso site e aquele apresentado dentro do quadro. Isso pode deixar seu site vulnerável a ataques do tipo *cross-site* e seus visitantes suscetíveis a ataques XSS.

Mas nem tudo está perdido e podemos proteger um pouco mais nosso site usando configurações de *sandbox* e definindo *políticas de referência* mais restritivas. E é sobre isso que falaremos nos próximos tópicos desse capítulo.

Tornando as coisas mais seguras

Aprendendo a usar a caixa de areia

Cientes dos problemas que um `iframe` mal configurado pode trazer, a HTML5 implementou o novo recurso de *sandbox*. Uma caixa de areia (*sandbox*) é um ambiente controlado que permite exibir outros sites para seu visitante sem que ele possa "tomar controle" do site principal e aos dados do visitante.

Quando habilitamos a `sandbox` em nosso iframe, automaticamente o site que está dentro do quadro perde algumas funcionalidades, dentre elas:

- Não pode mais enviar dados de formulários
- Não pode mais executar scripts
- Desabilita todo tipo de API, janelas modais e *popups*
- Desabilita todo tipo de plugin com `<embed>`, `<object>`, `<applet>`
- Evita que o site dentro do iframe assuma a navegação *top level* do navegador
- Bloqueia recursos como autoplay e foco automático em elementos de formulário

Para carregar um site dentro de um quadro *iframe* em ambiente protegido, basta declarar:

```
<iframe src="https://www.cursoemvideo.com" frameborder="0" width="400"
height="400" scrolling="yes" sandbox="sandbox"> </iframe>
```

Usar a propriedade `sandbox` com o valor `sandbox` (entre aspas) vai ligar todos os bloqueios citados acima.

Caso queira abrir alguma excessão e desbloquear algum dos recursos pontualmente, poderá usar um ou mais valores da lista a seguir:

- `allow-forms`
- `allow-scripts`
- `allow-modals`
- `allow-popups`
- `allow-same-origin`
- `allow-top-navigation`

REFERÊNCIA COMPLETA: Para uma lista completa de todos os valores suportados pela propriedade `sandbox`, acesse o site.

W3Schools: https://www.w3schools.com/tags/att_iframe_sandbox.asp

Definindo a política de referência

Outra maneira de limitar a ação de sites que carregarmos em nossos iframes é configurando a **política de referência** (*referral policy*). Sem entrar muito em termos

técnicos, normalmente os sites conseguem monitorar nossos rastros de navegação através de um campo no cabeçalho http chamado *Referer Header*. Ele sempre indica onde o visitante estava quando decidiu ir para o seu site. Esse é um dos recursos que as redes sociais como Twitter, Facebook e Instagram e sites como Google e YouTube usam para entender nosso padrão de navegação e mostrar propagandas. Mas não é só pra isso que serve.

Ao configurar a propriedade `referrerpolicy` em um *iframe*, conseguimos definir quantos dados serão compartilhados com o site de destino. A maneira de manter a navegação mais "anônima" é usar o valor `no-referer` para esse atributo.

Existem outros valores, como:

- `no-referer-when-downgrade` (padrão)
- `origin`
- `origin-when-cross-origin`
- `same-origin`
- `strict-origin`
- `strict-origin-when-cross-origin`
- `unsafe-url`

REFERÊNCIA COMPLETA: Para uma lista completa de todos os valores suportados pela propriedade `referrerpolicy`, acesse o site.

W3C: https://www.w3.org/TR/referrer-policy/

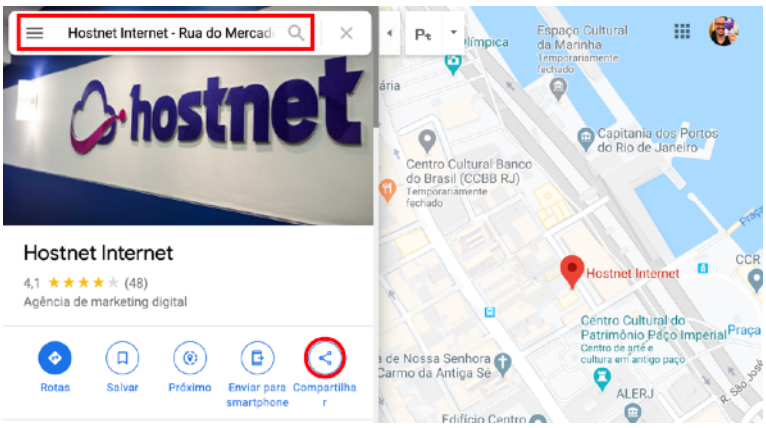
Outros exemplos do uso de iframes

Além do **YouTube** e do **Vimeo**, que já usamos para incorporar vídeos aos nossos sites, também podemos usar outros serviços para acrescentar ainda mais conteúdos aos nossos projetos com HTML.

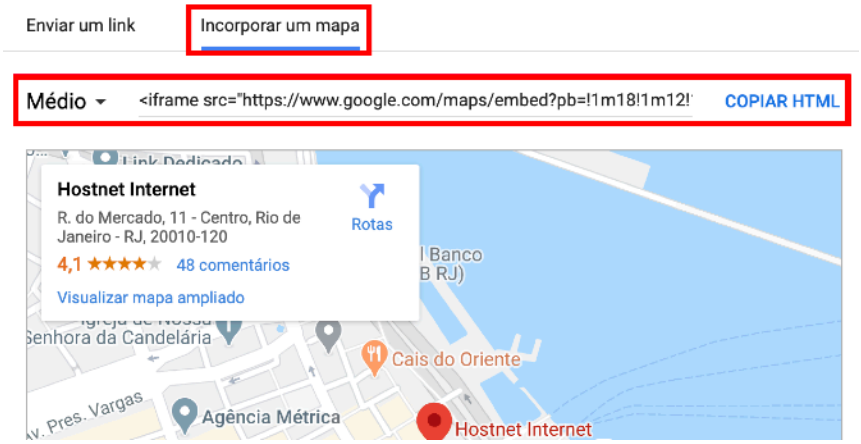
NADA DE AREIA AQUI, RAPAZINHO: Nenhum desses recursos, incluindo vídeos de YouTube e Vimeo, funcionam corretamente se habilitarmos o recurso de *sandbox* que vimos anteriormente.

Google Maps

Podemos também adicionar mapas do Google Maps ao seu site. Para isso, basta acessar o site `maps.google.com` e fazer uma busca de local previamente cadastrado no serviço ou colocar um endereço específico. Logo em seguida, clique sobre o botão redondo com a identificação **Compartilhar** (veja a imagem a seguir)

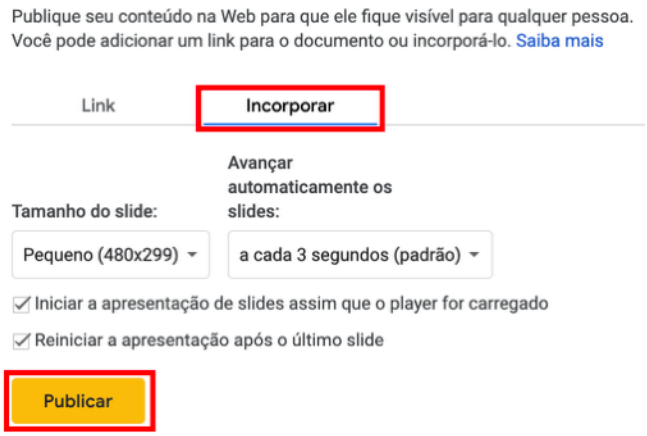


Ao pressionar o botão de compartilhamento, na tela a seguir, escolha a opção **Incorporar um mapa** e escolha um dos tamanhos disponíveis. Para finalizar, clique sobre o botão **COPIAR HTML** e o código do `iframe` será colocado na sua área de transferência. Agora vá até seu código HTML no **VSCode** e pressione `Ctrl+V` para colar o elemento copiado.



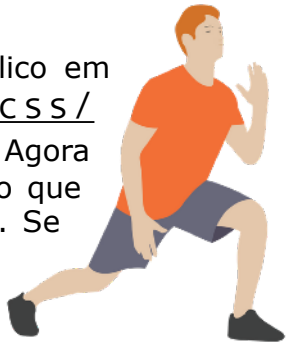
Google Docs

A suite de documentos do **Google** também permite incorporar conteúdos diretamente em sites através de *iframes*. Para isso, acesse sua conta no Google Docs e crie um documento novo (pode ser texto, planilha ou apresentação). Assim que terminar, clique na opção **Arquivo > Publicar na Web** que fica na parte superior esquerda da aplicação (cuidado pra não confundir com o menu do seu navegador).



A tela que vai aparecer tem a opção de **Incorporar** o documento e - dependendo do tipo de documento - você fará algumas configurações básicas e pode clicar no botão **Publicar**. O código do *iframe* será disponibilizado e você já pode colocá-lo no seu HTML.

Hora de exercitar



Chegou a hora de acessar o endereço do nosso repositório público em https://gustavoguanabara.github.io/html-css/desfios/ e executar os **exercícios 024** no seu computador. Agora tente atingir esse mesmo resultado em casa, sem copiar o código que eu criei. Nesse momento, a prática é algo que você mais precisa. Se por acaso ficar difícil, pode acessar o repositório público de HTML e CSS e dar uma olhada nos comandos, mas **EVITE COPIAR**.

Tenho desafios pra você!

Lá no repositório, além do material em PDF e dos códigos dos exercícios 100% disponíveis, também disponibilizamos alguns **desafios** que devem ser resolvidos. Esses desafios não incluem o código original e você deve tentar chegar à resposta sem copiar nenhum código.

Com todo o conteúdo que vimos até essa aula, você já pode resolver o **desafio d013**. Acesse o repositório público, abra a área do curso de HTML+CSS e clique no link de acesso aos desafios. Manda ver! Só não fica pedindo a resposta! Você consegue resolver isso sozinho(a)!

Repositório em: https://gustavoguanabara.github.io

Eu já falei sobre isso no YouTube?

Eu sei que às vezes as pessoas gostam mais de assistir vídeos do que ler livros, e é por isso que eu lanço há anos materiais no canal Curso em Vídeo no YouTube. O link que vou compartilhar contigo tem o conteúdo explicado como você leu aqui, só que de forma mais ilustrada. Reserve um tempo dos seus estudos para assistir esse vídeo todo.

Curso em Vídeo: https://www.youtube.com/playlist?list=PLHz_AreHm4dlAnJ_jJtV29RFxnPHDuk9o