

 [gustavoguanabara](#) / [html-css](#) Public

Code Issues 200 Pull requests 13 Actions Projects 1 Security Insights

 master ▼

...

[html-css](#) / [aulas-pdf](#) / 10 - Ligações em toda parte.pdf



gustavoguanabara PDFs atualizados ✓

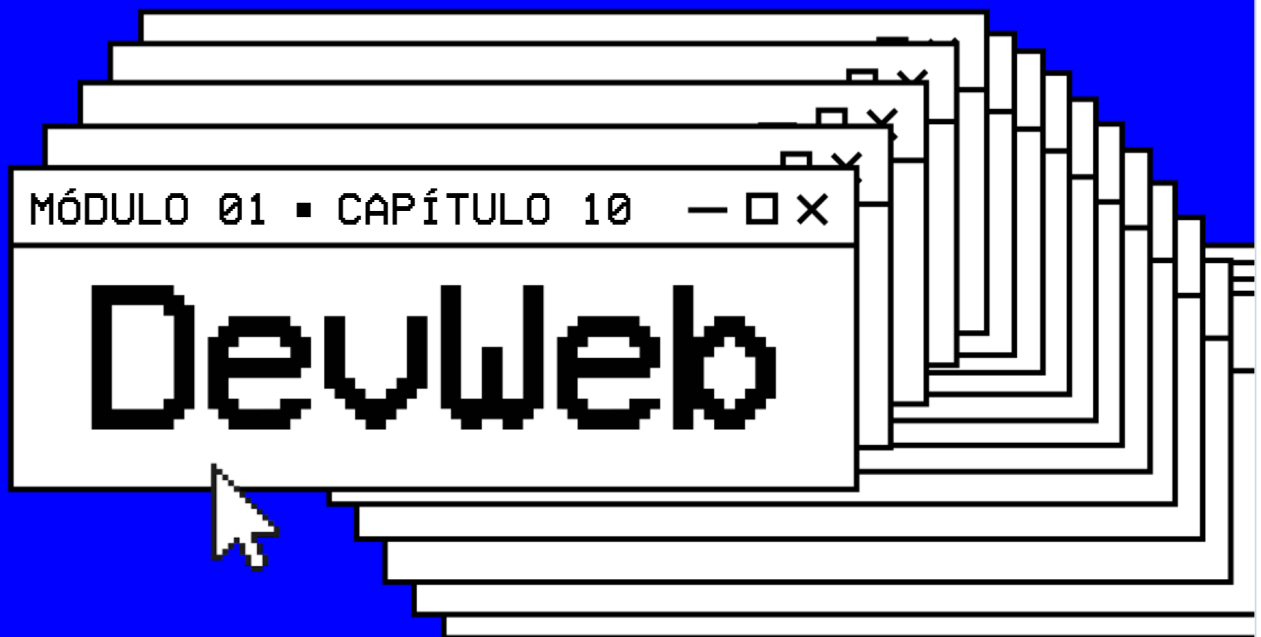


 1 contributor

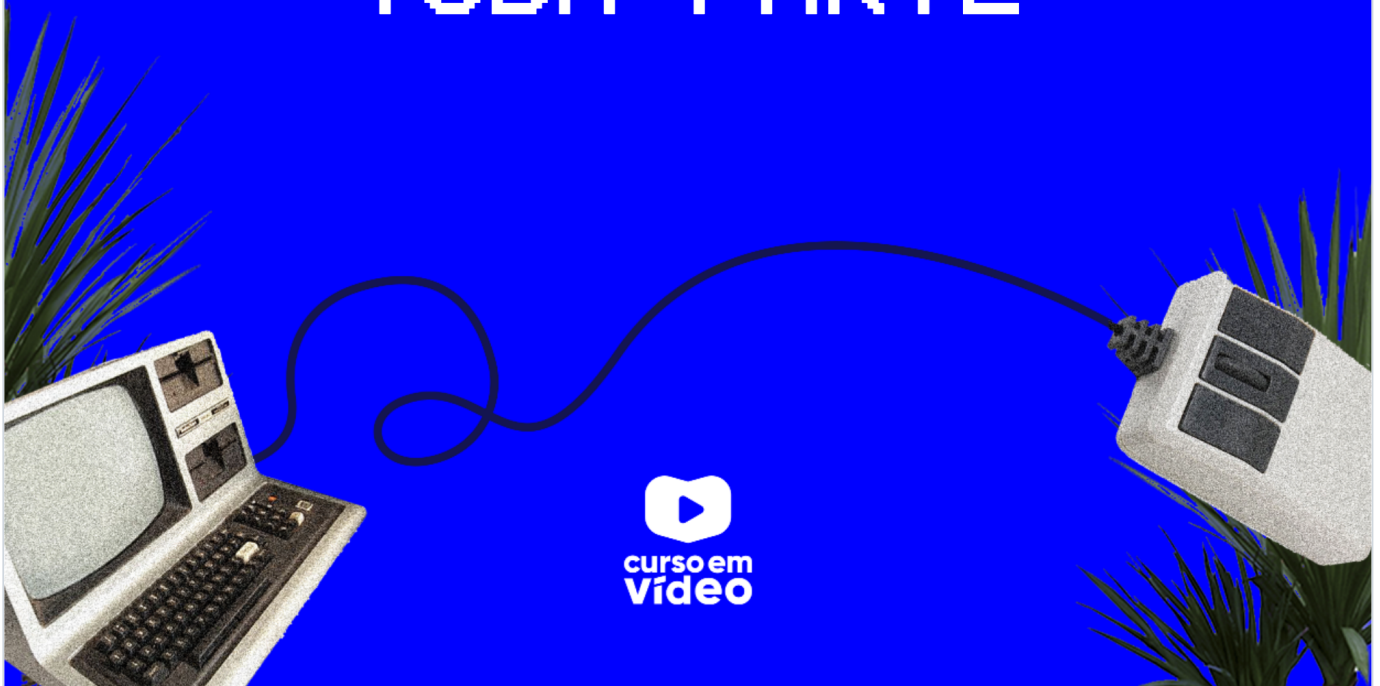
1.66 MB

...

CURSO DE INFORMÁTICA PARA INTERNET



LIGAÇÕES EM TODA PARTE



M01C10

LIGAÇÕES EM TODA PARTE

veja em vídeo



O link ou ligação é a essência fundamental do hipertexto. Ela nos permite ligar documentos entre si e permitir a navegação entre essas páginas. Nesse capítulo vamos aprender a configurar as âncoras para vários comportamentos, desde o mais simples que é criar um link interno até ligações que habilitam downloads de arquivos.



Você tem todo o direito de usar esse material para seu próprio aprendizado. Professores também podem ter acesso a todo o conteúdo e usá-los com seus alunos. Porém todos o que usarem esse material - seja para qual for a finalidade - deverão manter a referência ao material original, criado pelo **Prof. Gustavo Guanabara** e disponível no endereço do seu repositório público <https://github.com/gustavoguanabara/>. Este material não poderá ser utilizado em nenhuma hipótese para ser replicada - integral ou parcialmente - por autores/editoras para criar livros ou apostilas, com finalidade de obter ganho financeiro com ele.

Joga a âncora, marujo

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*.



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.



APRENDA MAIS: Veja com mais detalhes como funcionam os algoritmos fundamentais de busca assistindo esse vídeo do próprio **Google**, onde o engenheiro **Matt Cutts** explica o mecanismo básico da ferramenta mais valiosa do mundo. Habilite as legendas em PT-BR.



Google: <https://youtu.be/BNHR6IQJGZs>

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:

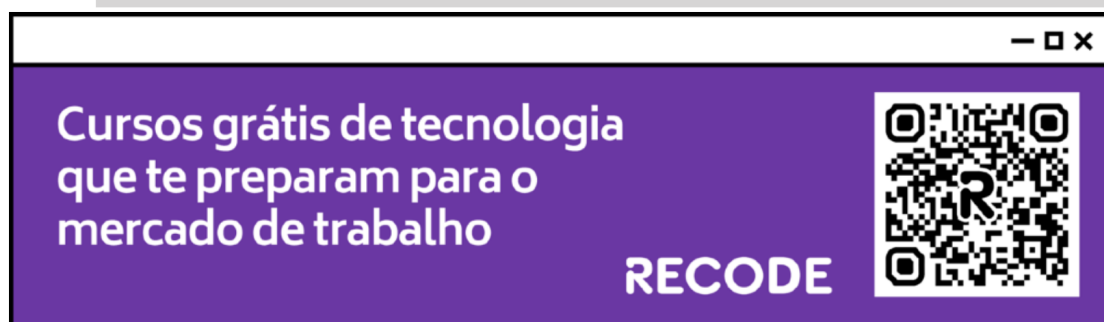
```
<h1>Vamos criar um link</h1>
```

```
<a href="https://gustavoguanabara.github.io">Acesse meu perfil GitHub</a>
```

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



NÃO ENTENDEU? Você sabe o que é uma URL? Pois volte para o nosso **capítulo 02** e vá até a parte onde falamos de domínios e hospedagem. Lá explicamos melhor o que são *Uniform Resource Locators* e quais são os seus principais componentes.



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>
```

Mira no Alvo

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?

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

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://gustavoguanabara.github.com" target="_blank">  
  Abrir perfil GitHub em 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

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

```
<a href="https://gustavoguanabara.github.com" target="_blank" rel="external">  
  Abrir perfil GitHub em nova janela  
</a>
```

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.



NAVEGANDO POR PASTAS LOCAIS: Não sei se você conhece um pouco do mundo **Linux**, mas servidores Web normalmente rodam esse sistema operacional. Se estiver se referindo à pasta atual do servidor, pode usar `./` antes do nome do arquivo. Se quiser se referir à pasta imediatamente superior na hierarquia, use `../` para voltar um nível para a pasta mais externa. E por favor, **aprenda Linux** 🐱

E para fazer 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:



