

Popups nativos no navegador (Popover API)

Resumo da aula

- Popups estão em toda parte. Eles são amplamente utilizados para diversos fins, como exibir anúncios, captar leads (através de formulários), fornecer notificações ou oferecer interações rápidas com o usuário.
- Embora possam ser úteis para chamar a atenção ou aumentar a conversão, seu uso excessivo ou intrusivo pode prejudicar a experiência do usuário, tornando importante o equilíbrio entre relevância e frequência.
- Implementar um popup não é nenhuma tarefa de outro mundo, mas também não é tão trivial quanto gostaríamos. É necessário criar a interface com HTML, customizar sua aparência com CSS e lidar com os diferentes casos de interação com o JS.
- É por isso mesmo que geralmente usamos bibliotecas como o SweetAlert2 (que tem aula aqui no canal) para fazer isso pra gente. Mas e se eu te disser que já existe uma API nativa para fazer isso? Estou falando da API popover! E sabe o melhor? Não precisamos nem de JavaScript!
- Para exemplificar o seu funcionamento, nós faremos uma pequena janelinha para que o usuário se cadastre na nossa newsletter.
- A primeira coisa que precisaremos é de um botão que vai acionar a nossa caixinha:

```
<!DOCTYPE html>
<html lang="pt-br">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Popover API</title>
</head>

<body>
  <button>
    Inscreva-se na nossa newsletter!
  </button>
</body>

</html>
```

- Agora vamos atrelar este comportamento com a janela que desejamos que apareça. Para isso, usamos o atributo `popover target`. Veja:

```
<!DOCTYPE html>
<html lang="pt-br">

<head>
  <meta charset="UTF-8">
```

```

<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Popover API</title>
<style>
  #newsletter-box::backdrop {
    background: rgba(0, 0, 0, 0.6);
  }
</style>
</head>

<body>
  <button popovertarget="newsletter-box">
    Inscreva-se na nossa newsletter!
  </button>

  <div id="newsletter-box" popover>
    O conteúdo fica aqui
  </div>

</body>

</html>

```

- Olha só que legal o resultado!

fazendo um modalzinho de boas sem dores de cabeça

- **Conflito:** Mas e se não ficar claro para o usuário que ele deve clicar fora do modal ou usar a tecla `ESC` para sair?
- Neste caso, podemos alterar o atributo `popover` do nosso modal para ser manual!

```

<div id="newsletter-box" popover="manual">
  O conteúdo fica aqui
</div>

```

- Só que agora está impossível sair do modal!
- Vamos adicionar um botão para acionar o comportamento de fechar.

```

<div id="newsletter-box" popover="manual">
  <button class="close-btn" popovertarget="newsletter-box"
    popovertargetaction="hide">
    <span aria-hidden="true"> </span>
  </button>
  <p>O conteúdo fica aqui</p>
</div>

```

- Pronto! Inclusive agora podemos tirar o valor `"manual"` do atributo `popover` e ter os dois comportamentos!

Repositório

<https://github.com/Professor-DiegoPinho/popover-api>