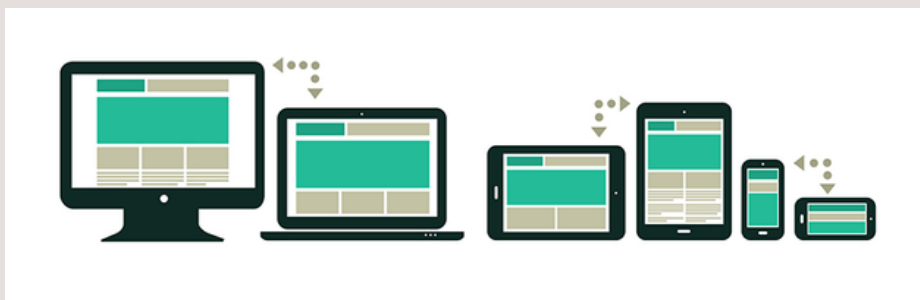


MATERIAL DE APOIO

MEDIA QUERIES

Imagine que você está criando um site. Você quer que ele fique bonito e fácil de usar tanto em um computador grande quanto em um celular pequeno. Para isso, você precisará ajustar o layout do seu site de acordo com o dispositivo que está sendo usado.

Media Queries são como regras especiais no CSS que permitem que você aplique estilos diferentes para diferentes tipos de dispositivos ou condições de exibição. Por exemplo, você pode definir um estilo específico para telas pequenas (como celulares), outro para telas médias (como tablets) e outro para telas grandes (como computadores).



Com o surgimento de smartphones, tablets, laptops e computadores desktop, os sites devem se adaptar a diferentes tamanhos e resoluções de tela. Os Media Queries nos ajudam a criar designs responsivos que se ajustam perfeitamente a esses diferentes dispositivos, proporcionando aos usuários a melhor experiência possível. Os browsers ou as aplicações lêem as expressões definidas na Query, caso o dispositivo em questão se encaixe nessas requisições o CSS será aplicado.

Por que usar Media Queries?

- Design responsivo: Faz com que seu site se adapte automaticamente a diferentes tamanhos de tela, proporcionando uma melhor experiência ao usuário.

- **Otimização para dispositivos móveis:** Garante que seu site seja fácil de navegar e visualizar em smartphones e tablets.
- **Melhora na classificação em mecanismos de busca:** Muitos mecanismos de busca, como o Google, dão preferência a sites responsivos.

Sintaxe Básica

Media Queries consistem na aplicação de um tipo de dispositivo (media type) e uma ou mais expressões que verificam determinadas condições. Por exemplo, podemos especificar que determinados estilos devem ser aplicados somente se a largura da tela estiver acima ou abaixo de um determinado valor.

@media	screen	(min-width: 320px)	and	(max-width: 768px)
AT-RULE	MEDIA TYPE	MEDIA FEATURE	OPERATOR	MEDIA FEATURE

@media

Palavra-chave que indica o início de uma media query.

```
@media [media-type] ([media-feature]){  
  /* Estilos a serem aplicados */  
}
```

Media type

Sua funcionalidade serve para direcionar determinado CSS para um meio específico de dispositivo, ou seja, definem para qual tipo de mídia o CSS que você aplica em seu site será direcionado.

Em muitos (se não na maioria) dos casos, você verá um **screen** usado aqui, já que muitos dos tipos de mídia que estamos tentando combinar são dispositivos com telas anexadas a eles (desktops, tablets, celulares, etc).

```
@media screen {  
  /* Estilos a serem aplicados */  
}
```



Mas as telas não são o único tipo de mídia que podemos atingir, é claro. Temos alguns, incluindo:

- **all:** corresponde à todos os dispositivos;
- **print:** corresponde a documentos exibidos em uma visualização de impressão ou em qualquer mídia que divida o conteúdo em páginas destinadas à impressão;
- **screen:** combina dispositivos com uma tela
- **speech:** corresponde a dispositivos que leem o conteúdo de forma audível, como um leitor de tela;

MEDIA FEATURES

As Media Features são expressões que descrevem características de um dispositivo, como a largura da tela, a altura, a orientação (retrato ou paisagem), a resolução, a densidade de pixels e muito mais. Ao combinar essas features nas suas Media Queries, você pode criar regras CSS extremamente precisas e adaptar seu site para diferentes cenários.

```
/* Estilos para telas com largura mínima de 768px */
@media (min-width: 768px) {
  /* Estilos a serem aplicados */
}

/* Estilos para dispositivos na orientação paisagem */
@media (orientation: landscape) {
  /* Estilos a serem aplicados */
}

/* Estilos para telas com resolução de 1920x1080 pixels */
@media (min-width: 1920px) and (min-height: 1080px) {
  /* Estilos a serem aplicados */
}
```



Abaixo listamos algumas das principais Media Features e suas aplicações:

- **width e height:** Definem a largura e a altura da área de visualização, em pixels.
- **min-width e max-width:** Especificam a largura mínima e máxima da área de visualização, respectivamente.
- **min-height e max-height:** Especificam a altura mínima e máxima da área de visualização, respectivamente.
- **device-width e device-height:** Definem a largura e a altura físicas do dispositivo, em pixels.
- **orientation:** Define a orientação do dispositivo (portrait ou landscape).
- **aspect-ratio:** Define a proporção entre a largura e a altura da área de visualização.
- **resolution:** Define a resolução da tela, em dpi (dots per inch).

OPERADORES

As Media Queries oferecem suporte a operadores lógicos como muitas linguagens de programação, para que possamos corresponder tipos de mídia com base em determinadas condições. Os operadores de Media Queries são como as ferramentas que você usa para combinar e personalizar suas regras CSS, permitindo que você crie layouts ainda mais flexíveis e adaptados a diferentes dispositivos.



Eles são essenciais para criar lógicas complexas e garantir que seu site se adapte perfeitamente a qualquer tela.

```
@media (min-width: 768px) and (orientation: landscape) {  
  /* Seus estilos aqui */  
}
```

Abaixo listamos alguns dos principais operadores e suas aplicações:

- **and:** Combina várias condições. Por exemplo, `@media (min-width: 768px) and (orientation: landscape)` aplica o estilo para telas com largura mínima de 768px e orientação paisagem.
- **or:** Permite escolher entre várias condições. Por exemplo, `@media (min-width: 768px) or (max-width: 480px)` aplica o estilo para telas com largura mínima de 768px ou máxima de 480px.
- **not:** Nega uma condição. Por exemplo, `@media not (orientation: landscape)` aplica o estilo para dispositivos na orientação retrato.
- **only:** Especifica que a regra se aplica apenas a um determinado tipo de mídia. É menos utilizado hoje em dia, mas pode ser útil em alguns casos específicos.





BREAKPOINTS

Em desenvolvimento web, breakpoints são pontos específicos na largura da tela onde você aplica mudanças no layout do seu site. Esses pontos são definidos usando Media Queries e servem para adaptar o design do seu site a diferentes tamanhos de tela, como smartphones, tablets e desktops.

Os breakpoints mais comuns são baseados nas larguras de tela típicas de diferentes dispositivos. No entanto, a escolha dos breakpoints pode variar dependendo do seu projeto específico e do público-alvo.

- **Extra small (XS):** Dispositivos muito pequenos, como alguns smartphones antigos. Largura máxima: 320px.
- **Small (SM):** Smartphones modernos. Largura máxima: 576px.
- **Medium (MD):** Tablets em modo retrato. Largura máxima: 768px.
- **Large (LG):** Tablets em modo paisagem e alguns laptops. Largura mínima: 992px.
- **Extra large (XL):** Desktops e monitores de alta resolução. Largura mínima: 1200px.





```
@media (max-width: 320px) {  
  /* Extra Small (XS) - Estilo para telas com largura máxima de 320px */  
  /* Dispositivos muito pequenos, como alguns smartphones antigos */  
}  
  
@media (max-width: 576px) {  
  /* Small (SM) - Estilo para telas com largura máxima de 576px */  
  /* Smartphones modernos */  
}  
  
@media (max-width: 768px) {  
  /* Medium (MD) - Estilo para telas com largura máxima de 768px */  
  /* Tablets em modo retrato */  
}  
  
@media (min-width: 992px) {  
  /* Large (LG) - Estilo para telas com largura mínima de 992px */  
  /* Tablets em modo paisagem e alguns laptops */  
}  
  
@media (  
: 1200px) {  
  /* Extra Large (XL) - Estilo para telas com largura mínima de 1200px */  
  /* Desktops e monitores de alta resolução */  
}
```

NA PRÁTICA

Vamos então criar um exemplo prático e simples usando Media Queries com o objetivo de demonstrar como criar um menu responsivo simples utilizando HTML e CSS, ajustando a disposição dos elementos de acordo com o tamanho de tela em que o site está sendo executado.



HTML

```
<body>
  <h2>Menu de navegação responsivo</h2>
  <p>
    Redimensione a janela do navegador para ver o efeito: Quando a tela tiver menos
    de 600px, o menu de navegação
    será exibido verticalmente em vez de horizontalmente.
  </p>

  <nav>
    <a href="#">Link</a>
    <a href="#">Link</a>
    <a href="#">Link</a>
  </nav>
</body>
```

CSS

```
* {
  box-sizing: border-box;
}

nav {
  overflow: hidden;
  background-color: #333;
}

nav > a {
  float: left;
  display: block;
  color: #f2f2f2;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
}

nav > a:hover {
  background-color: #ddd;
  color: black;
}
```

```

/* Em telas com 600 px de largura ou menos, faz com que os links do menu sejam
empilhados, em vez de ficarem lado a lado */
@media screen and (max-width: 600px) {

  nav > a {
    float: none;
    width: 100%;
  }
}

```

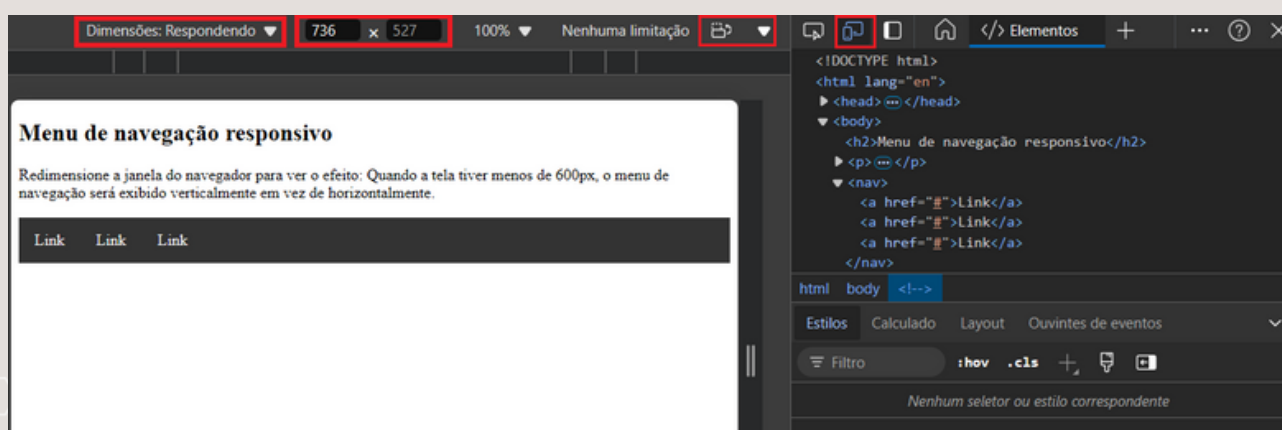
Ao visualizar o resultado do código acima utilizando um navegador moderno, você verá o seguinte conteúdo:

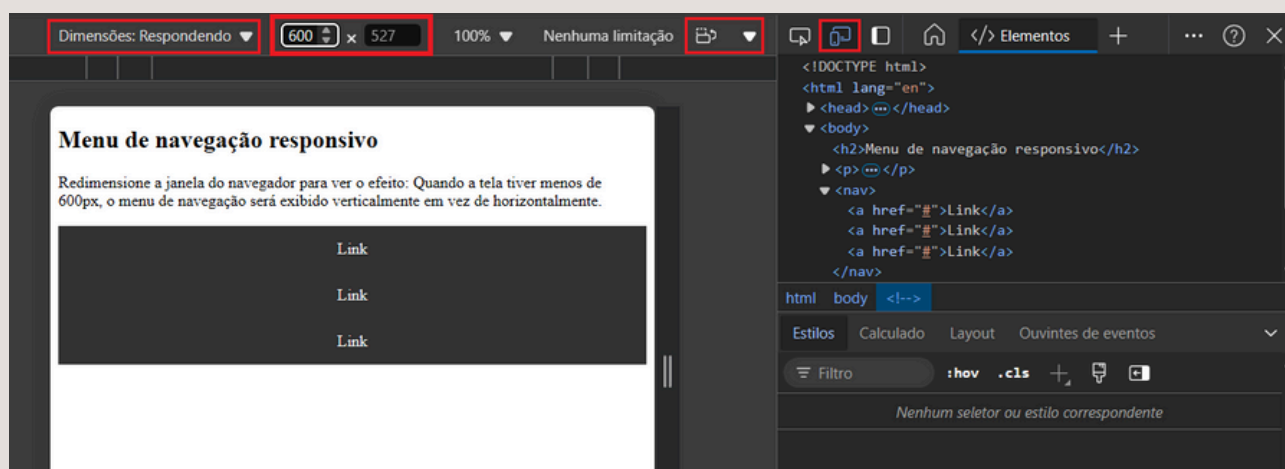
Menu de navegação responsivo

Redimensione a janela do navegador para ver o efeito: Quando a tela tiver menos de 600px, o menu de navegação será exibido verticalmente em vez de horizontalmente.

Link Link Link

Utilize a ferramenta DevTools do seu navegador para utilizar o recurso “Emulação de dispositivo”.





Perceba que assim que a largura de tela é redefinida para 600px, a disposição dos itens do menu muda de “lado a lado” para um “abaixo do outro”.

Utilize esse recurso do seu browser para testar suas próprias definições de Media Queries!

REFERÊNCIAS

- [CSS Media Queries \(w3schools.com\)](https://www.w3schools.com/css/css3_mediaqueries.asp)
- [Usando Media Queries - CSS | MDN \(mozilla.org\)](https://developer.mozilla.org/en-US/docs/Web/CSS/Media_Queries_Using_the_Words)
- [CSS Media Queries Guide | CSS-Tricks](https://css-tricks.com/media-queries/)

BONS ESTUDOS!

