

ntander Coders 2024.1 | Front-End | #1178

Isaias Soares

Id: 1178008



media queries

Antes de entrarmos nas *media queries*, vale a pena mencionar que existe algo anterior chamado *media type*.

Os *media types* são tipos de dispositivos capazes de renderizar uma página web, como celulares, computadores e até impressoras (sim, podemos customizar o CSS para impressão!).

< Os *media types* têm alguns problemas nos dias de hoje, celulares têm telas muito boas que podem ser consideradas telas de computador, mas são bem menores.

Sendo assim, acabamos por usar um novo *media type*, que nos permite fazer uma consulta sobre as capacidades do dispositivo e tomar decisões de design baseadas na resposta.

Em programação, chamamos consultas de *queries*, logo, essas consultas são os *media queries*.

Só para constar, os *media types* eram:

- **all** Todos os dispositivos.
- **braille** Dispositivos táteis.
- **embossed** Dispositivos que imprimem em Braille.
- **handheld** Dispositivos de mão, (telas pequenas e banda limitada).
- **print** Dispositivos de impressão em papel.
- **projection** Para apresentações, como PowerPoint.
- **screen** Dispositivos com telas coloridas e alta resolução.

- **speech** Sintetizadores de voz. (se tiver interesse no tema consulte: [especificação aural](#)).
- **tty** Dispositivos com grade fixa para exibição de caracteres, como teletypes e terminais.
- **tv** Dispositivos como televisores.

O que é especialmente interessante dos *media types* é que eles podem ser usados no *tag link* para carregar um CSS específico para cada tipo de dispositivo:

```
<link rel="stylesheet" href="estilo.css" type="text/css" media="print" />
```

O exemplo acima carrega o "estilo.css" apenas para impressoras, pois possui `media="print"`.

Isso quer dizer que podemos também carregar um CSS específico usando *media query*.

Consultas mais comuns:

- **width, min-width** e **max-width** verificam o tamanho da tela.
- **orientation** verifica a orientação, pode ser **landscape** ou **portrait**.
- **resolution, max-resolution** e **min-resolution** verifica a resolução da tela.
- **pointer** verifica se há suporte ao mouse.

As consultas do *media query* devem ser tratadas como valores booleanos, ou seja, como valores verdadeiros ou falsos. Para isso, normalmente, comparamos o resultado da consulta como um resultado esperado. Como booleanos, podemos montar consultas mais complexas usando os operadores lógicos **not**, **or**, **and** ou **only**.

Se um *media type* for utilizado na consulta, ele já retornará verdadeiro ou falso. Se quisermos ter um background azul quando a tela tiver menos que 1000px, faríamos o seguinte:

```
@media screen and max-width: 1000px;
```

Com **screen**, garantimos que é um monitor (apesar de celulares modernos também entrarem nessa categoria), e com **max-width: 1000px** garantimos que a tela atual não tem mais do que 1000px de tamanho.

Usamos muito esse modelo de *query* para fazer responsividade.

Outro exemplo:

```
@media (min-width: 600px) and (orientation: landscape);
```

Nesse exemplo, só aplicaremos CSS se a tela tiver mais de 600px e estiver em orientação paisagem.

Essa *query* é usada para adaptar o site ao posicionamento do celular e reagir às mudanças de orientação.

Esses exemplos podem ser usados na propriedade **media** de **link** para carregar o CSS apenas quando as condições forem satisfeitas. Mas também podemos fazer *media queries* diretamente no CSS. Por exemplo:

Dentro de um arquivo .css

```
@media screen and (max-width: 600px) {  
  main {  
    margin: 0;  
  }  
}
```

```
@media screen and (min-width: 601px) and (max-width: 1000px) {  
  main {  
    margin: 0 10px;  
  }  
}  
  
@media screen and (min-width: 1001px) {  
  main {  
    width: 80%;  
    margin: 0 auto;  
  }  
}
```

Observe que basta colocar o CSS dentro da *media query*.

No exemplo, todas as *queries* aplicam-se apenas aos monitores, devido ao **screen**. No primeiro, indicamos que, apenas para telas menores de **600px**, o **main** não terá **margin**; no segundo, indicamos que, para telas entre **601px** e **1000px**, o **main** terá **margin** de **10px** nas laterais. No terceiro, indicamos que, para telas maiores de **1001px**, o **main** terá **margin** automática e ocupará **80%** da tela, o que, na prática, centraliza o conteúdo.

Testando responsividade

Você pode estar pensando: "Como faço para testar todos esses tipos de tela se não tenho um monte de dispositivos?"

É fácil, nas ferramentas de desenvolvedor de praticamente todos os navegadores temos um ícone indicando a possibilidade de simular outros tipos de devices e tamanhos de tela:

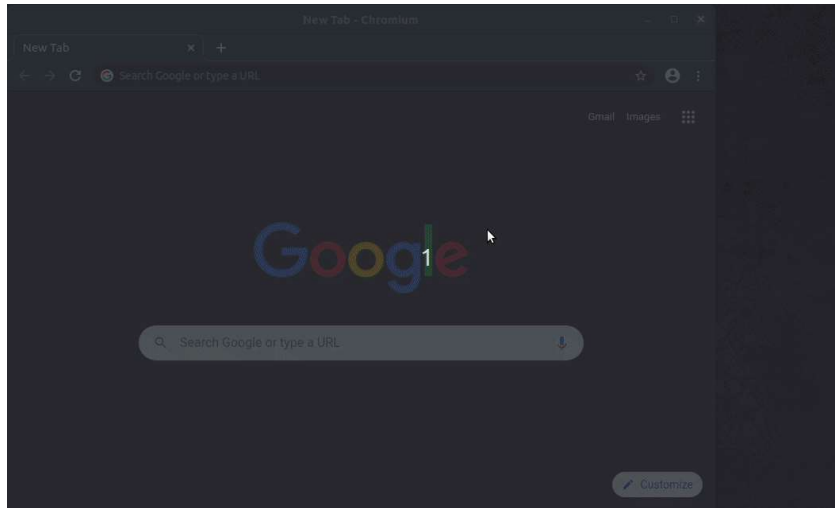


Imagem 1 - Barra de ferramentas do desenvolvedor no Chrome (Fonte da imagem: do autor)

Agora você pode adaptar seu design para diferentes tamanhos de tela, removendo, adicionando ou reposicionando elementos conforme a necessidade.

Recomendamos que aprofunde seus estudos procurando como fazer tamanhos de fonte responsivos usando `clamp()`, `min()` e `max()` e imagens responsivas. Com isso, você terá um ótimo conjunto de técnicas à sua disposição.

Referências e Links úteis

- [Usando Media Queries](#). MDN Web Docs, 08/05/2022. Acesso em: 27/06/2022.

