VEM SER

HTML e CSS

Aula 04 - Responsividade, Media Queries e animations


calc

calc

calc é uma função do CSS que nos permite realizar cálculos simples.

Podem ser realizados cálculos de:

- Adição;
- Subtração;
- Divisão;
- Multiplicação.


Responsividade

Responsividade

Um site responsivo é aquele cujo conteúdo se adapta ao formato de tela do dispositivo utilizado para a sua visualização.

Ou seja, deve exibir corretamente o *layout* e o conteúdo.

Qual a importância da responsividade?

Mais da metade da população com acesso à internet acessa exclusivamente pelo celular.

Os *smartphones* se popularizaram cada vez mais e o público que acessa a internet pelo celular só cresce a cada dia. Um site responsivo garante que esse público seja devidamente atendido.

https://canaltech.com.br/internet/internet-alcanca-74-dos-brasileiros-e-58-utilizam-a-rede-apenas-pelo-celular-165851/

Curiosidade: *mobile first*

Mobile first é uma metodologia cujo foco de desenvolvimento primário e principal é a usabilidade em dispositivos móveis.

Diferentemente da metodologia de desenvolvimento mais praticada, em que a versão desktop é desenvolvida primeiro e somente depois a *mobile*, quando o *mobile first* é adotado, primeiro é desenvolvida a versão *mobile* e só então são desenvolvidas as versões para outras plataformas.

(*Mobile first, desktop last*)

Tamanhos de tela

Os principais tamanhos de tela atualmente são:

- Celular;
- Tablet;
- Desktop (inclui tanto notebooks quanto monitores).

Breakpoints

Um ***breakpoint*** é justamente a "quebra" entre um tamanho de tela e outro.

Quantos utilizar? É interessante que os sites mais modernos tenham **no mínimo** dois (celular e desktop), mas é recomendado pelo menos três (celular, tablet e desktop).

Para que a maior parcela do público seja atendida, os sites costumam ter de **três a quatro** *breakpoints*. Esse quarto *breakpoint* seria para monitores com baixa resolução de tela.

Desktop

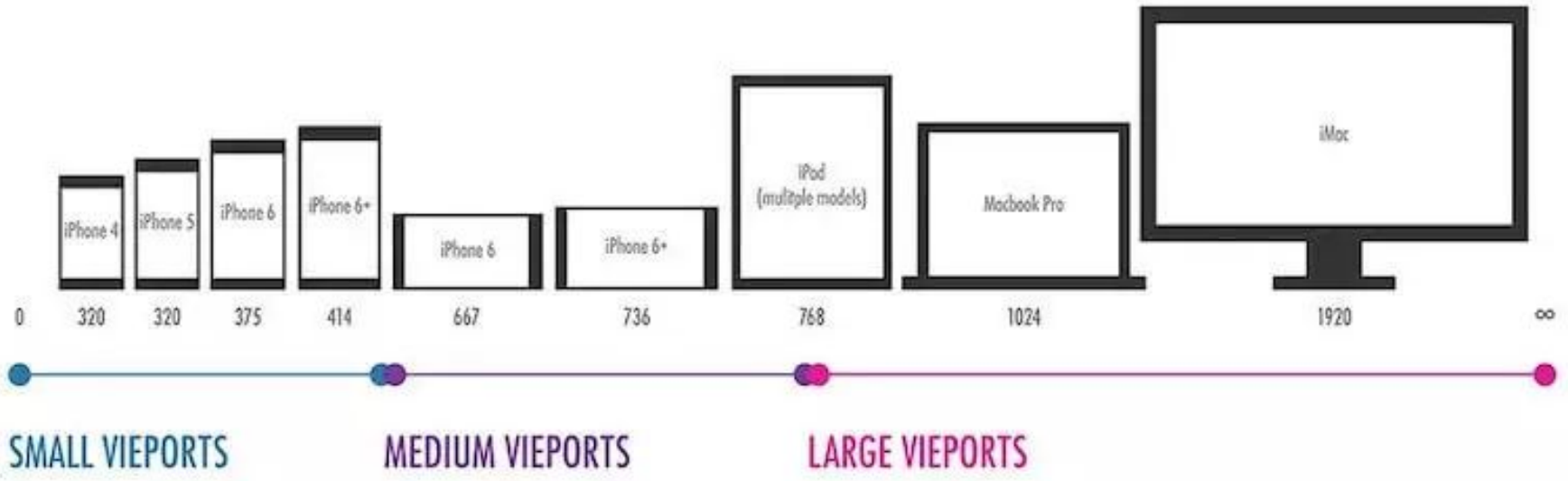
@media screen and
(min-width: 1024px)
{...}

Tablet

@media screen and
(min-width: 768px) and
(max-width: 1023px)
{...}

Smartphone

@media screen and
(max-width: 767px)
{...}


iPhone 4
iPhone 5
iPhone 6
iPhone 6+
iPhone 6
iPhone 6+
iPod
(multiple models)
Macbook Pro
iMac
0
320
320
375
414
667
736
768
1024
1920
∞
SMALL VIEPORTS
MEDIUM VIEPORTS
LARGE VIEPORTS

Media Queries: Responsividade na prática

Formas de usar *media queries*

Dentro do arquivo *style.css*

```css
.container {
  width: 100px;
  height: 100px;
  background-color: blue;
}

@media (max-width: 900px) {
  .container {
    background-color: red;
  }
}
```

Formas de usar *media queries*

Em um arquivo externo

```
<link rel="stylesheet" href="style.css" />
<link rel="stylesheet" href="mobile.css" media="(max-width: 700px)" />
<link
  rel="stylesheet"
  href="tablet.css"
  media="(min-width: 700px) and (max-width: 1024px)"
/>
```


Escondendo
elementos

Display: none

```
1  .container {
2    display: none;
3  }
```

Visibility: hidden

```css
.container {
  visibility: hidden;
}
```

O resultado prático é o “mesmo”.
Então quais são as diferenças entre um e outro?

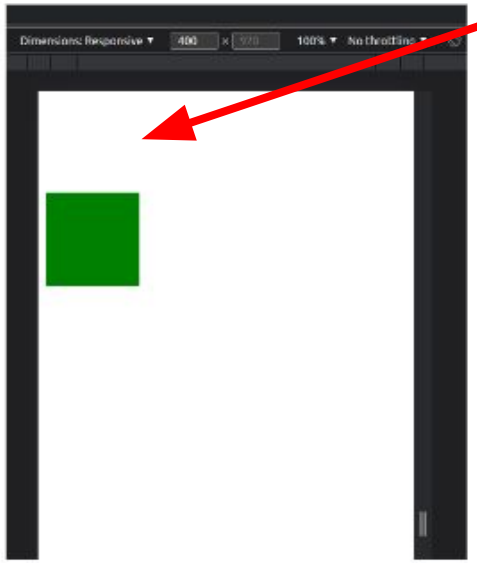
Vamos imaginar que temos duas divs:

```
1  <div class="container"></div>
2  <div class="segundo-container"></div>
```

E agora vamos esconder a primeira div com *visibility: hidden*.

?????

```
.container {
  visibility: hidden;
}
```

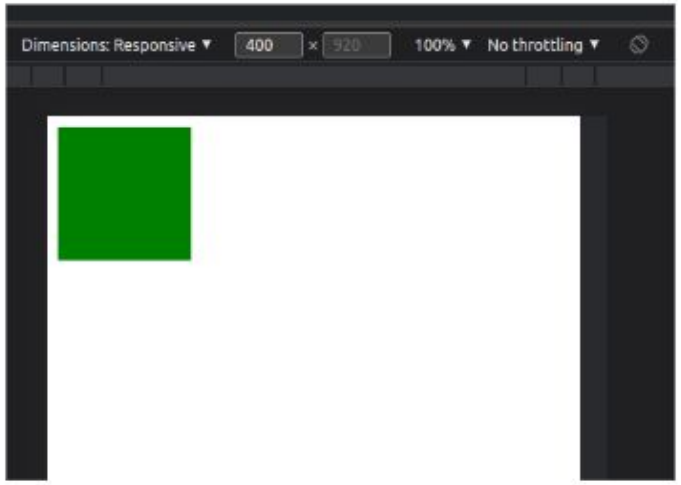


Agora vamos esconder com *display: none*:

```css
.container {
  display: none;
}
```

“Consertando” o *visibility: hidden*

```css
.container {
  visibility: hidden;
  width: 0;
  height: 0;
}
```




Transitions
e Animations

Transition

Uma transição irá intermediar dois estados de um elemento.

A propriedade **transition** é *shorthand* para quatro propriedades:

- transition-property;
- transition-duration;
- transition-timing-function;
- transition-delay.

A utilização de transitions pode deixar a experiência do usuário com a interação de elementos que alteram estados mais fluída e agradável.

Animation

Com o ***CSS Animation*** conseguimos fazer animações relativamente complexas podendo utilizar somente CSS.

Alguns exemplos de animações podem ser: **Loaders**, **menus hambúrguer**, elementos que **mudam de posição na tela** e etc.

Exemplo: https://codepen.io/desenvolvweb/pen/xxGwzqZ

Animation

Cada animação tem ***keyframes,*** que nos ajudam a definir como deve ocorrer a sequência da animação. Ou seja, como a animação deve ser renderizada.

Eles nos ajudam a estabelecer onde a animação deve começar e onde deve terminar (from/to, 50%/100%).

Também podem existir mais de um estado intermediário, por exemplo: 25%/50%/75%/100%.

Animation

A propriedade **animation** é um *shorthand* para as seguintes propriedades:

- **animation-name;**
- **animation-duration;**
- **animation-delay;**
- **animation-timing-function;**
- **animation-iteration-count;**
- animation-direction;
- animation-fill-mode;
- animation-play-state.

Dicas de material

https://www.youtube.com/watch?v=LCEgHntqBps


Let's Tech Up Together