



UNIVERSIDADE
FEDERAL DO CEARÁ

Tecnologias Web

CSS3 - parte 3

UFC – Engenharia da Computação – 2021-2

Prof. Fischer Jônatas Ferreira

Agenda

➡ Responsividade

➡ Media Queries

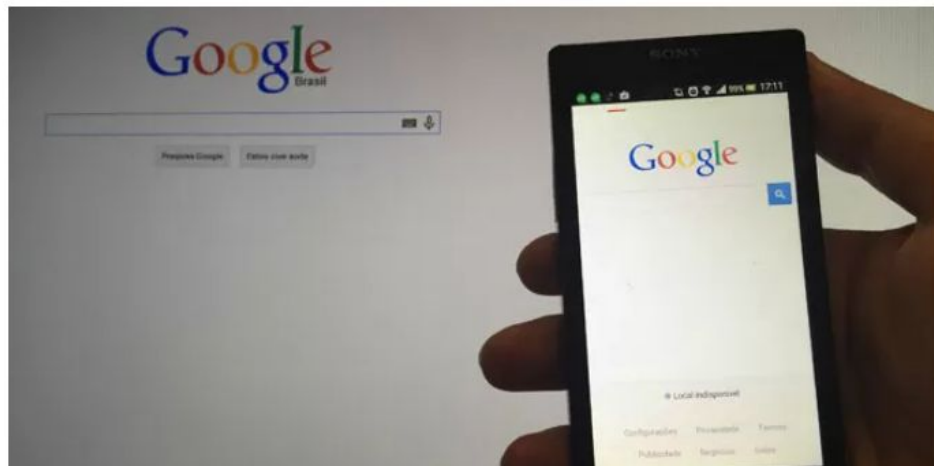


Google passa a 'esconder' sites descalibrados com o mundo móvel

Empresa vai considerar se página funciona bem em smartphones e tablets. Além de design adequado, páginas têm de carregar com internet móvel.

Helton Simões Gomes

Do G1, em São Paulo



As páginas na internet não preparadas para smartphones estão prestes a escorregar para bem longe do topo de pesquisas feitas no Google. A gigante da tecnologia passa a considerar se um site é amigável a dispositivos

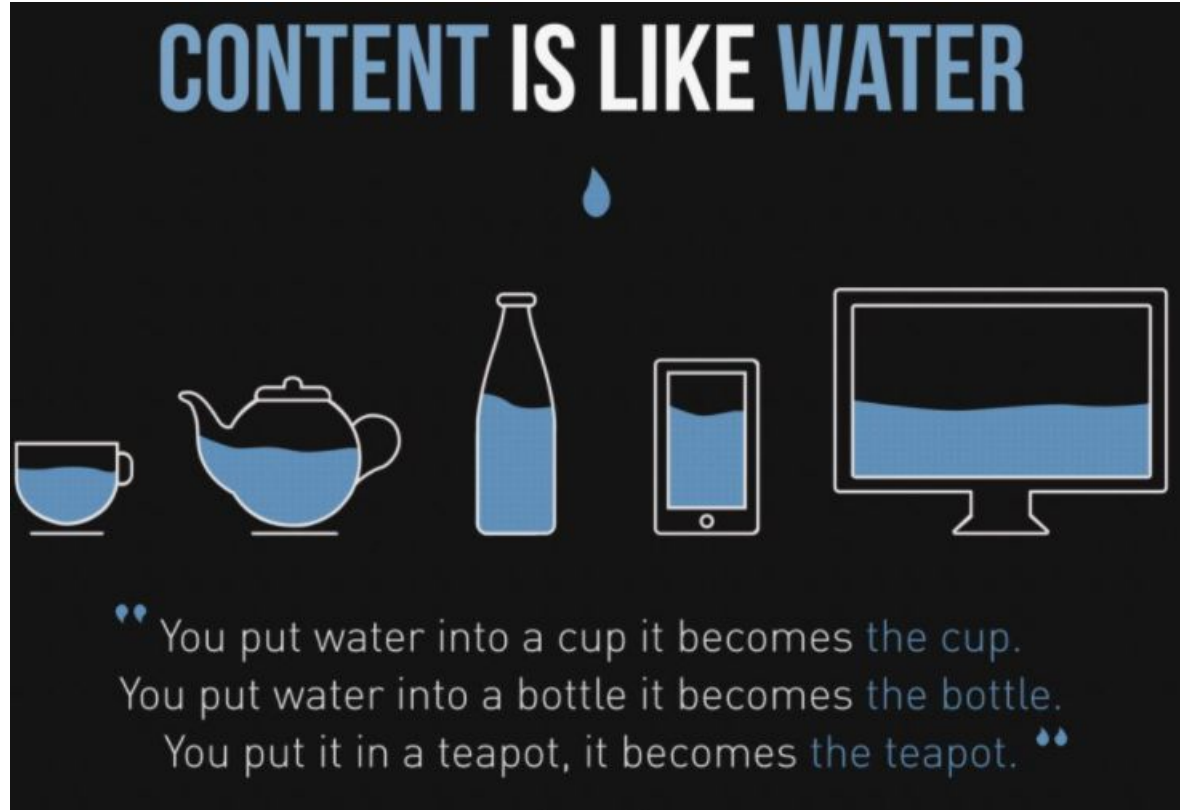
Layouts Web/Mobile e Responsividade

Tipos de layout

- **Layout Líquido:** se ajusta a janela porque seus componentes utilizam valores em % (porcentagem);
- **Fixo:** construído para um tamanho específico e dependendo do dispositivo que é apresentado pode gerar barras de rolagem ou re-escalar a página toda;
- **Responsivo:** usa um estudo de arquitetura de informação para construir um layout inteligente que usa % e PX para mostrar o que é essencial ao usuário em um tamanho proporcional ao dispositivo que esta apresentando.



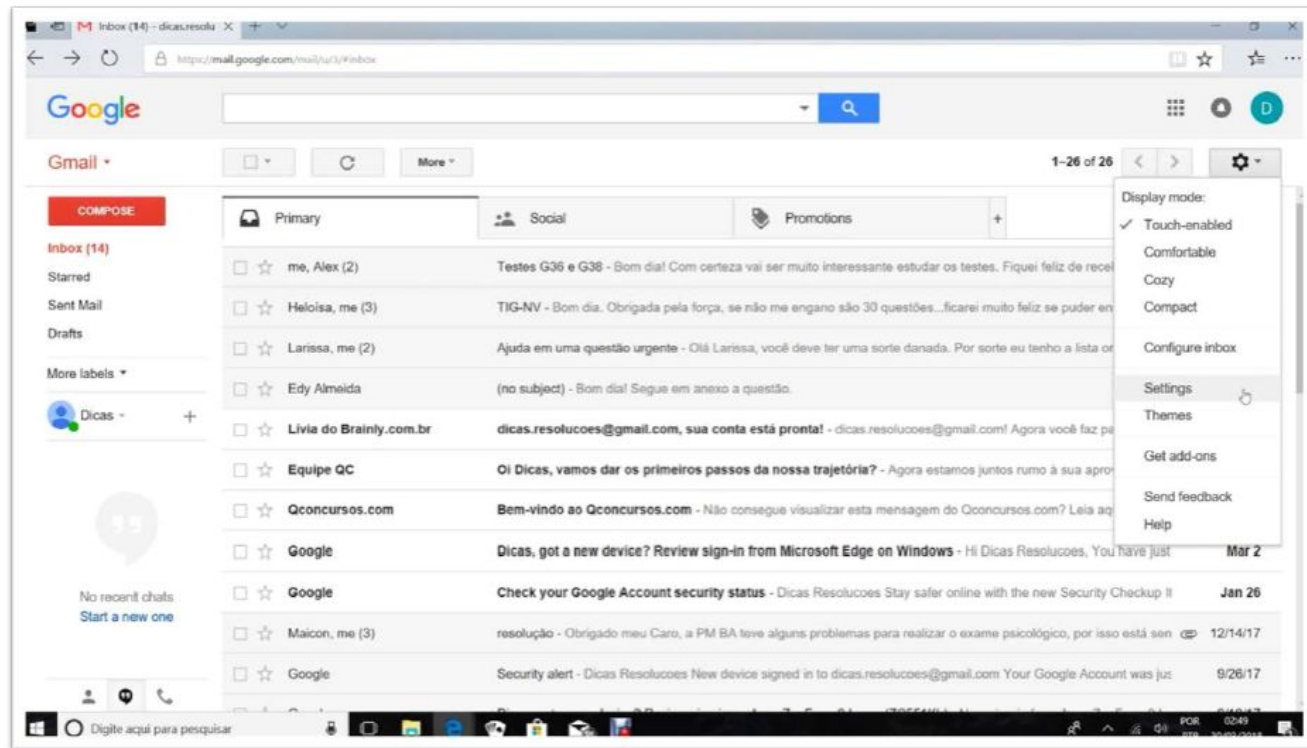
Tipos de Layouts



Tipos de Layouts

Layout Líquido:

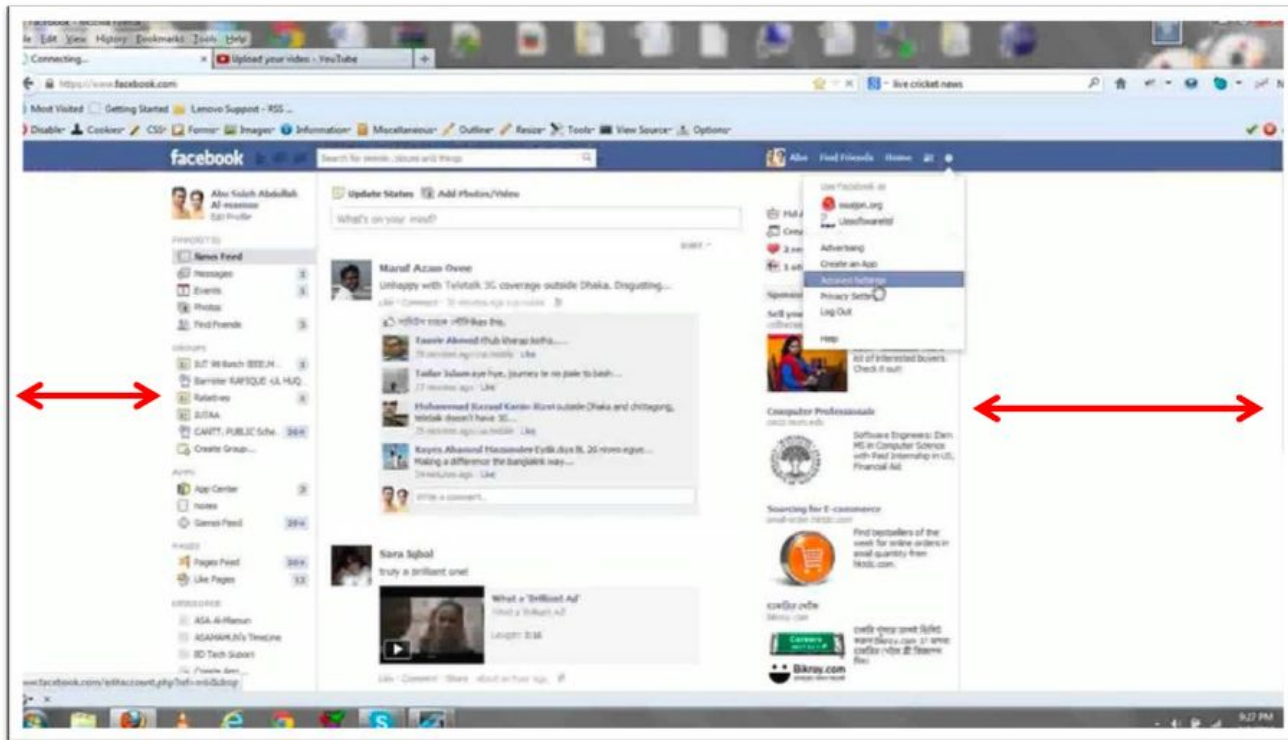
- Componentes somente se encaixam a janela;
- Não é inteligente para mostrar somente o que é útil;
- 70% dos componentes de UI são controlados por %;
- 30% são componentes de tamanho fixo (pixels);



Tipos de Layouts

Fixo:

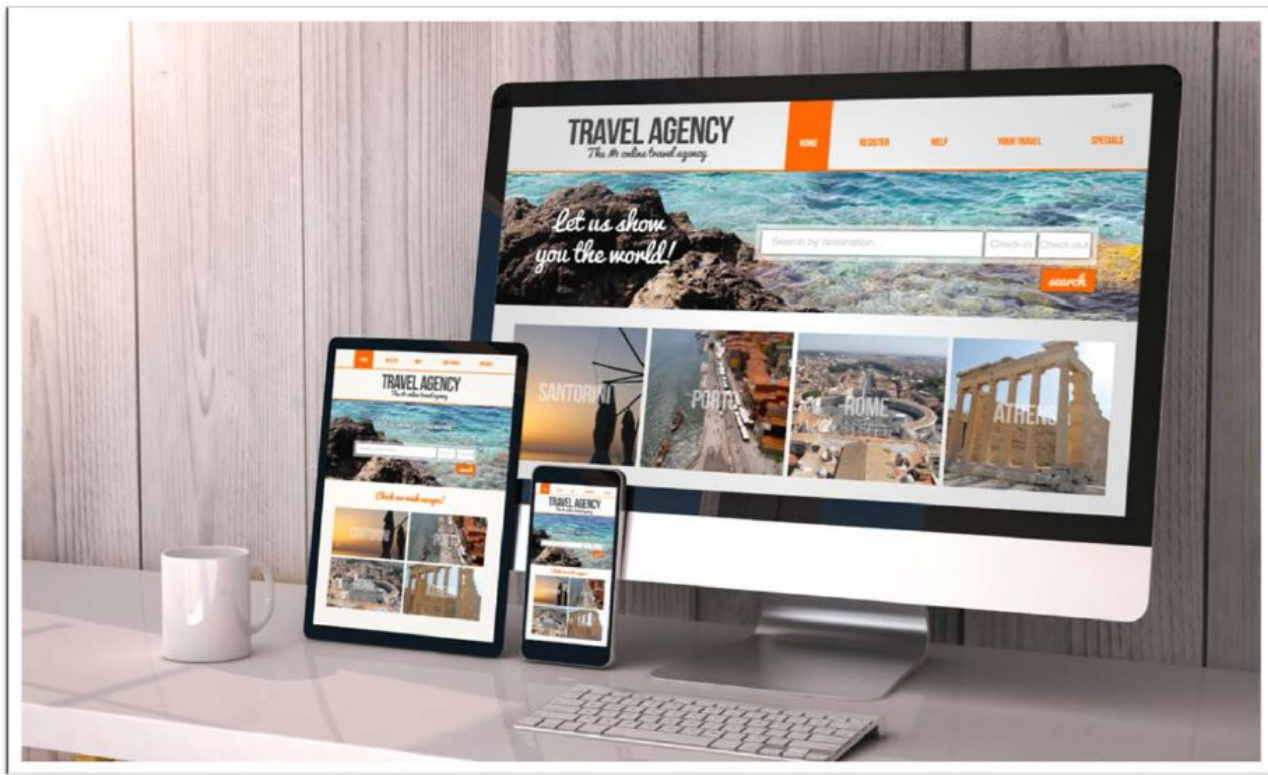
- Construído em um tamanho específico: 768px, 960px, 980px, 1200px, 1600px;
- Componentes controlados por PX;
- Quando uma tela é maior que seu tamanho seu conteúdo é centralizado ou ancorado e um dos lados (esquerda ou direita);
- Quando a tela é menor que seu tamanho ele escala para encaixar ou gera barras de rolagem.



Tipos de Layouts

Responsivo:

- Layout inteligente usa CSS e Javascript para construir uma experiência única ao usuário;
- Trabalho pesado de arquitetura de informação;
- Componentes pode sumir ao aparecer dependendo do *device* que está acessando;
- O mais caro (\$\$\$) dos layouts e mais demorado de ser construído;



O que é responsividade?

É um comportamento inteligente do layout para se adaptar a diferentes tipos de resolução de tela e dispositivos mostrando o que é essencial ao usuário com características que respeitam a arquitetura de informação. Para isso, elementos podem ser escondidos, priorizados ou novos serem inseridos.

New York, NY			
Tuesday, April 15th			
Overcast			
 58 °F		Precipitation: 100%	
		Humidity: 97%	
		Wind: 4 mph SW	
		Pollen Count: 36	
Today		68° 36°	Pollen 36
Wednesday		50° 39°	Pollen 36
Thursday		55° 39°	Pollen 36
Friday		54° 43°	Pollen 36

Quem faz a responsividade?

Conceitualmente:

- Programador de Front-end;
- Arquiteto de Conteúdo;

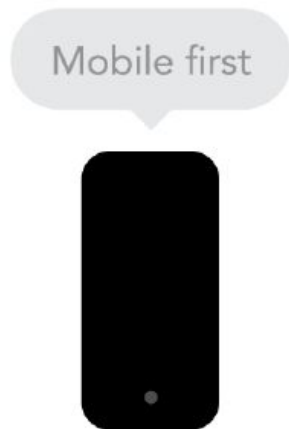
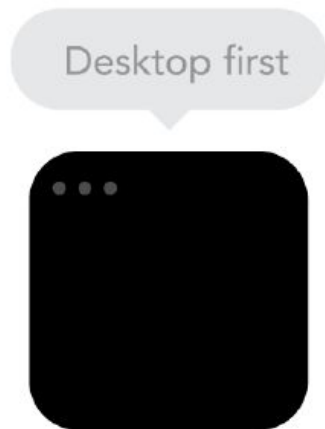
Tecnicamente:

- CSS
- Javascript



Mobile First

Mobile first é um projeto que faz a criação de projetos web e sites primeiro em dispositivos móveis, para depois fazer adaptações para o desktop e outras plataformas, ao contrário do que ainda acontece na maioria das vezes.

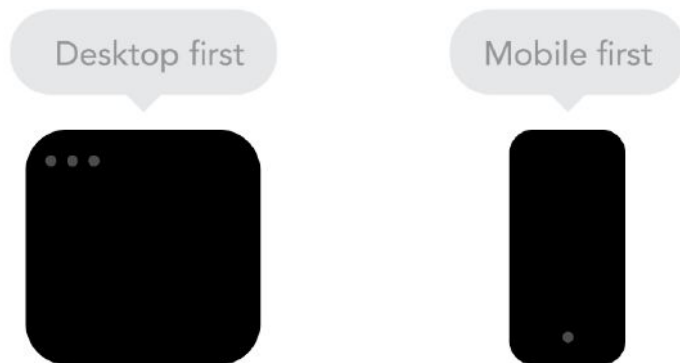


Fonte: <https://www.moblee.com.br/blog/mobile-first-principais-vantagens/>

Mobile First

Vantagens:

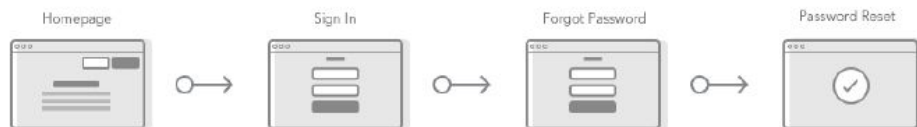
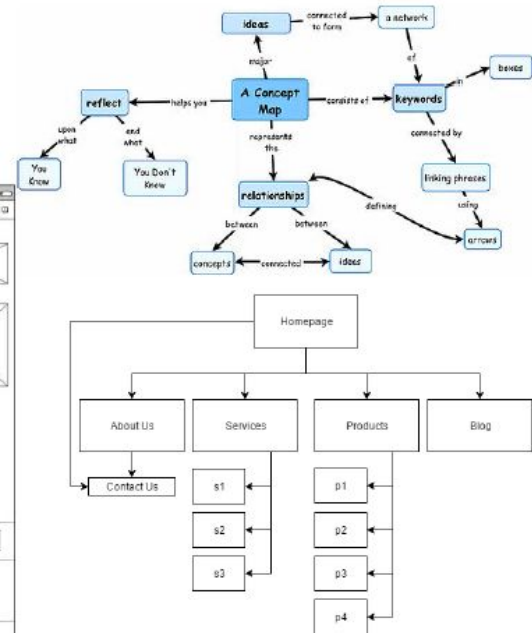
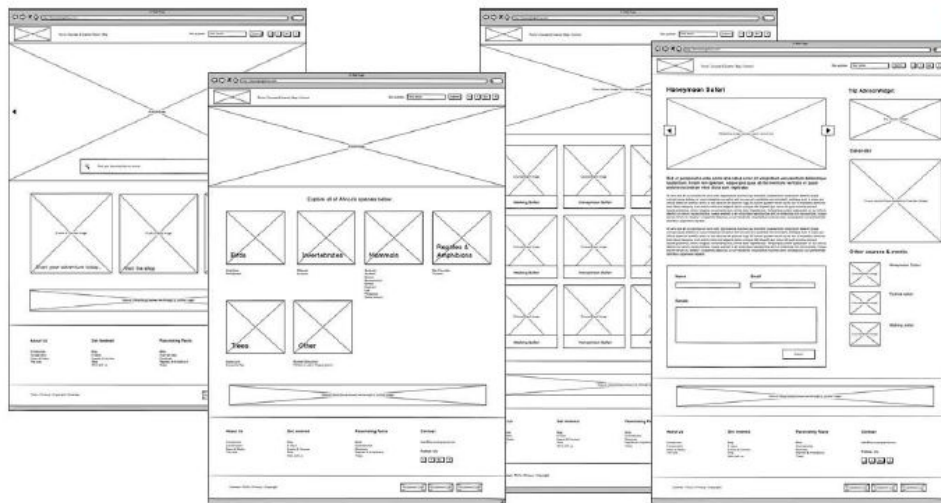

- O acesso via smartphone já é predominante em todo o mundo;
- Proporcionar uma experiência melhor ao usuário (UX);
- Quando o Desktop é construído primeiro muitas "gambiarras" são feitas para "encaixar" o layout ao mobile;
- Ranqueamento no Google (SEO);
- Otimização ao carregamento;



Fonte: <https://www.moblee.com.br/blog/mobile-first-principais-vantagens/>

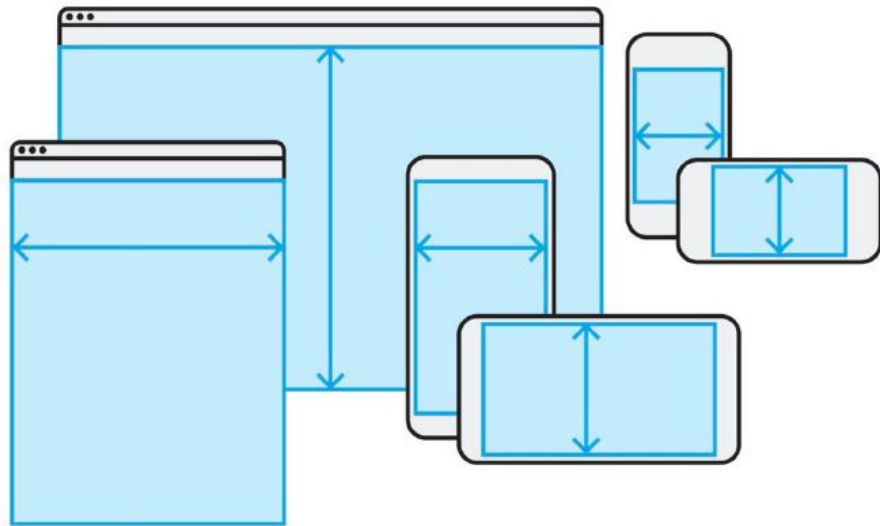
Mapas conceituais:

- Mapa do site;
- Wireframe;
- Taskflow;

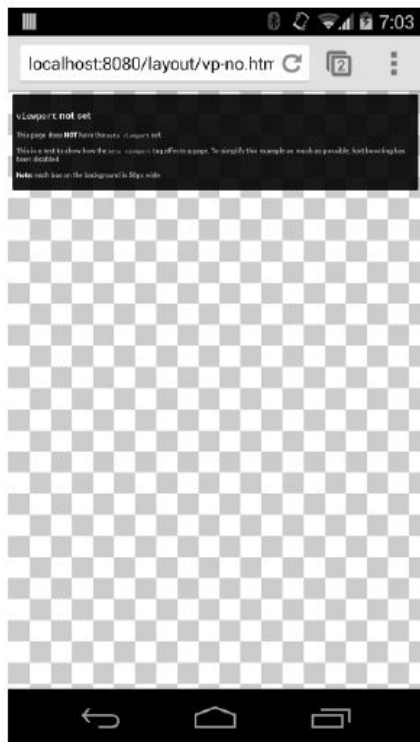


Viewport

- A **viewport** é a área visível do usuário de uma página da web.
- A janela de visualização varia de acordo com o dispositivo e será menor em um smartphone do que na tela do desktop.
- Antes de tablets e smartphones, as páginas da Web eram projetadas apenas para telas de computador, e era comum que as páginas da Web tivessem um design estático e um tamanho fixo.



Viewport



Sem definição do **viewport** o browser tenta encaixar a página na largura completa.



Com o **viewport** definido a página pode corresponder a largura do dispositivo.

Viewport

O tamanho que a **viewport** pode ser definido por uma tag meta e tem algumas propriedades:

```
<meta name="viewport" content="width=X, Y, Z">
```

X = valor fixo em PX ou **device-width**

, Y = initial-scale=1

, Z = minimum-scale / maximum-scale / user-scalable

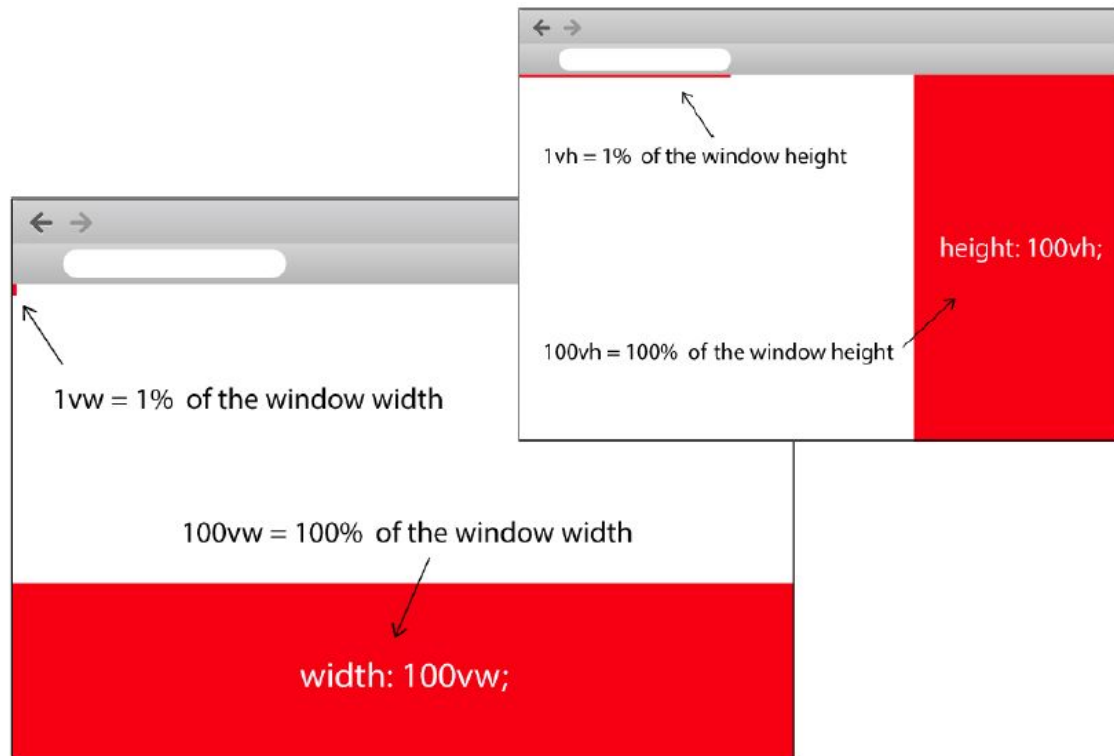
Ex:

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

Viewport

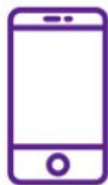
Introdução de tamanhos relativos que podem ser usados no CSS:

- Viewport width (vw)
- Viewport height (vh)
- Altura ou largura mínima da viewport (vmin)
- Altura ou largura máxima da viewport (vmax)



Uma mesma aplicação em várias telas

Mas, e se for necessário adaptar os tamanhos para vários tipos de dispositivos?



0-480

Smaller
smartphones



481-768

Tablets & larger
smartphones



769-1279

Laptops, larger tablets
in landscape, and small
desktops



1280+

Larger desktops
and monitors

Media Query

Media Query ao carregar uma folha de estilo

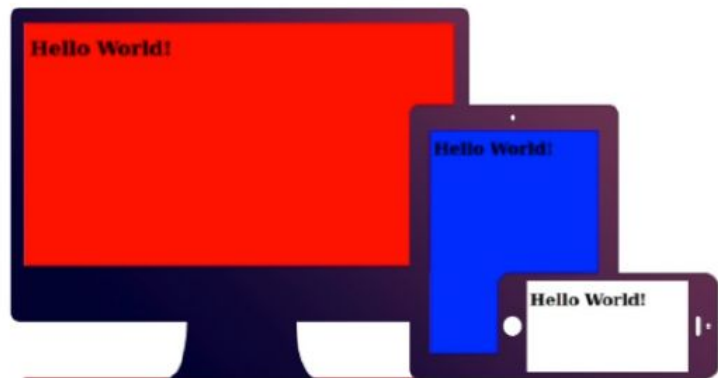
```
<!-- CSS media query em um elemento de link -->  
<link rel="stylesheet" media="(max-width: 800px)"  
href="example.css" />
```

Media Queries

É possível criar seletores CSS que irão operar somente em um determinado tamanho de tela, ou mesmo somente quando o documento estiver em modo impressão. Também é possível descobrir se um smartphone está deitado (paisagem) ou em pé (retrato).

```
@media screen and (min-width:1001px) and (max-width:1400px) {  
  body {  
    background-color: red;  
  }  
}  
@media screen and (min-width:501px) and (max-width:1000px) {  
  body {  
    background-color: blue;  
  }  
}  
@media screen and (min-width:240px) and (max-width:500px) {  
  body {  
    background-color: white;  
  }  
}
```

É possível também usar o NOT para fazer a negação



Media Queries



Os resultados da *query* são verdadeiros se o *media type* especificado na *media query* corresponde ao tipo do documento exibido no dispositivo e todas as expressões na *media query* são verdadeiras.

```
<style>
@media (max-width: 600px)
{
  .facet_sidebar
  {
    display: none;
  }
}
</style>
```

Se verdade aplica os estilo do corpo da @media



Media Queries

➡ É possível utilizar operadores lógicos: and, or, not, only

Se verdade aplica os estilo do corpo da @media

```
@media (min-width: 700px) and (orientation: landscape)
{ ... }
```



Media Queries

A sintaxe:

START WITH
@MEDIA

APPLICABLE
MEDIA TYPES

MEDIA
FEATURES



```
@MEDIA SCREEN AND (MAX-WIDTH: 768PX) {  
  #CONTENTS{ WIDTH:60%; }  
  #SIDEBAR{ WIDTH:40%; }  
}
```

CSS TO APPLY
WHEN CONDITIONS
ARE MET



Medias possíveis:

all
print
screen
speech
braile
projection

Condicionalis:

and
not

Referências

 <https://developer.mozilla.org/en-US/docs/Web/HTML/Element>