

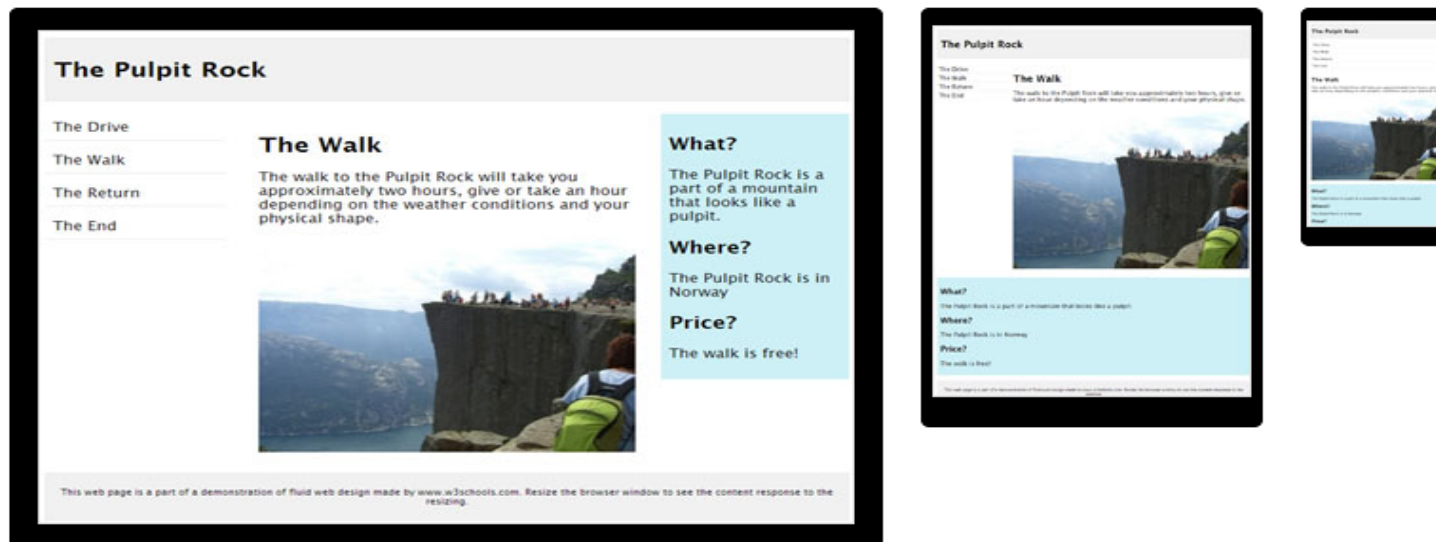
Programação II



Introdução ao Design Responsivo

Web Design Responsivo

- Design capaz de “responder” às características do dispositivo ao qual é servido.
- Um layout responsivo é capaz de expandir ou contrair com a finalidade de se acomodar de maneira usável e acessível à área onde é visualizado ou, mais genericamente, ao contexto onde é renderizado, seja um smartphone, um tablet, um desktop, um leitor de tela, um mecanismo de busca etc.
 - Compare em: <http://www.liquidapsive.com/>

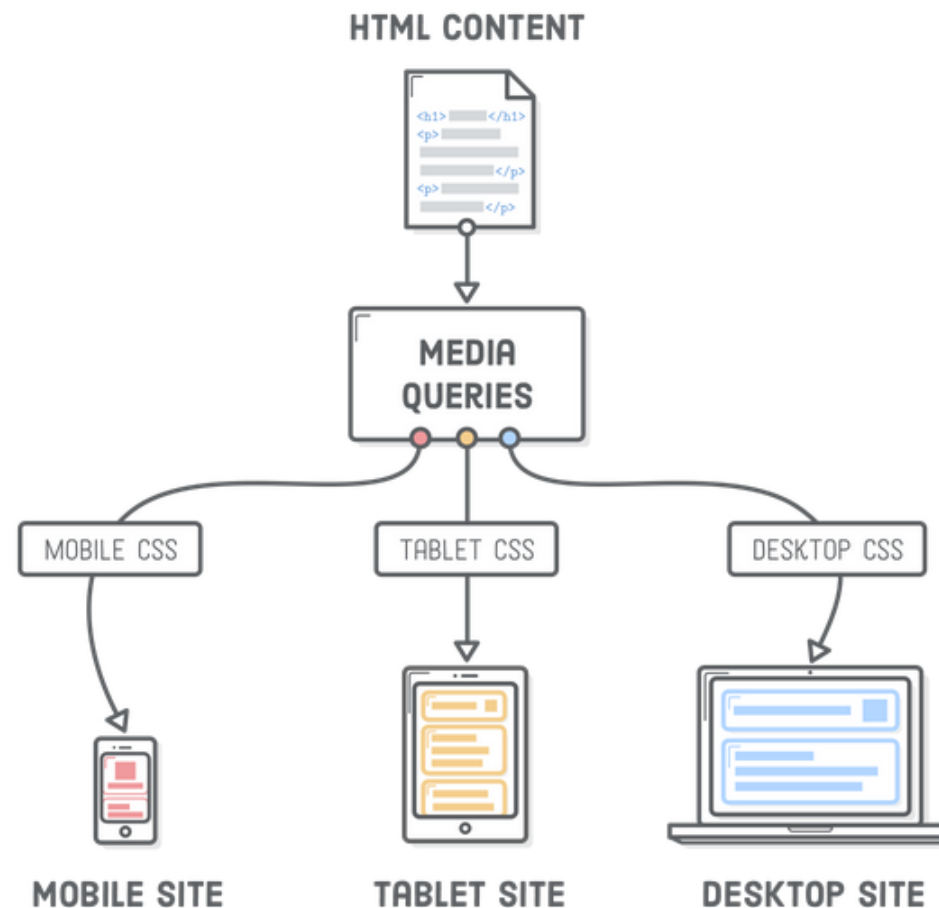


Ingredientes do RWD

- O conceito surgiu em 2010, quando Ethan Marcotte escreveu uma matéria chamada **Responsive Web Design** no site AlistApart.
- Ethan cita os seguintes “ingredientes técnicos” como componentes fundamentais para o desenvolvimento de design responsivo:
 - **Layout fluido**, utilizando predominantemente medidas relativas;
 - **Imagens e mídias flexíveis**, capazes de contrair ou expandir suas dimensões em função do contexto onde são renderizadas;
 - **Media queries**, tecnologia CSS que permite servir folhas de estilos baseadas em características do dispositivo ao qual o layout será servido.
- Podemos atingir estes 3 requisitos com HTML5, CSS3 e JavaScript (para eventuais compatibilidades)

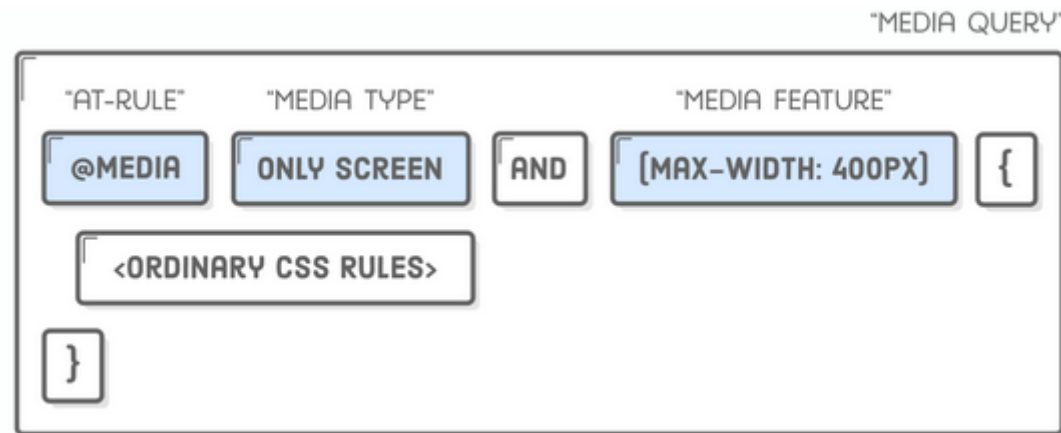
Media Queries

- Constituem um meio de aplicar regras CSS condicionalmente, definindo expressões lógicas (condições) para que um bloco de regras CSS seja utilizado em mídias específicas.
- Deste modo, pode-se exibir o mesmo conteúdo HTML em diferentes layouts CSS.



Media Queries

- Uma media query sempre inicia com `@media`, seguido de um *media type* e uma ou mais expressões booleanas envolvendo *media features* (características de mídia).



- O resultado da query é verdadeiro se o *media type* especificado é condizente com o dispositivo que está acessando o documento e se todas as expressões são verdadeiras.
 - Neste caso, os estilos especificada entre { } serão aplicados ao documento.

Media Queries

Tipo de mídia	Descrição
all	Todos os dispositivos de mídia
screen	Telas de computadores, tablets, smartphones
speech	Leitores de tela
print	Impressoras

Operador	Descrição
and	Operador de concatenação, retorna true se as condições à sua esquerda e direita são verdadeiras
only	Palavra-chave usada para esconder folhas de estilo dos navegadores antigos que não suportam media query
not	Operador de negação, retorna true se a condição não for satisfeita

O operador e a expressão são facultativos.

A não ser que se use os operadores not ou only, o media type é opcional e o tipo **all** é assumido.

Media Queries

- O exemplo ao lado altera a cor do plano de fundo de uma página de acordo com a largura da tela:
 - telas de no máximo 400px terão plano de fundo vermelho
 - telas entre 401px e 960px terão plano de fundo amarelo
 - telas com 961px ou mais terão plano de fundo azul
- O efeito pode ser verificado redimensionando-se a janela, ou através do **modo responsivo** presente nas ferramentas do desenvolvedor dos navegadores (tecla F12).

```
/* Mobile Styles */
```

```
body {  
    background-color: #F09A9D; /* Red */  
}
```

```
/* Tablet Styles */
```

```
@media only screen and (min-width: 401px) and  
(max-width: 960px) {  
    body {  
        background-color: #F5CF8E; /* Yellow */  
    }  
}
```

```
/* Desktop Styles */
```

```
@media only screen and (min-width: 961px) {  
    body {  
        background-color: #B2D6FF; /* Blue */  
    }  
}
```

Mobile First

- Conceito segundo o qual devemos criar nossos sites primeiramente para dispositivos móveis, e gradativamente incrementá-lo até a versão desktop (normalmente mais complexa e detalhada).
- Começamos o código CSS com os estilos *mobile* e os que serão padrão para todos os layouts, mantendo-os fora das *media queries*.
- Depois, definimos *media queries* para outros layouts, as quais sobrescrevem as regras previamente definidas, ou ainda, acrescentam regras específicas.
 - Observe como isso é feito no exemplo do slide anterior
- O desenvolvimento *mobile first* maximiza o reaproveitamento de código e contribui para manter o foco nos dados e ações que são realmente fundamentais, evitando a retirada de itens necessários ou a inclusão de elementos supérfluos.

Breakpoints

- Breakpoints são pontos nos quais o layout se readapta para se ajustar a uma nova largura de janela/tela.
 - Em outras palavras, são os valores utilizados como min-width e max-width.
- Considerando o exemplo de uma página com 3 colunas, podemos incluir a seguinte media query a fim de que, quando a janela atingir 600px de largura, todas as colunas (que no exemplo teriam nomes de classes iniciando por “col-”) sejam expandidas para 100%, transformando-o em um layout de coluna única.



```
@media only screen and (max-width: 600px) {  
  [class*="col-"] {  
    width: 100%;  
  }  
}
```



Orientação

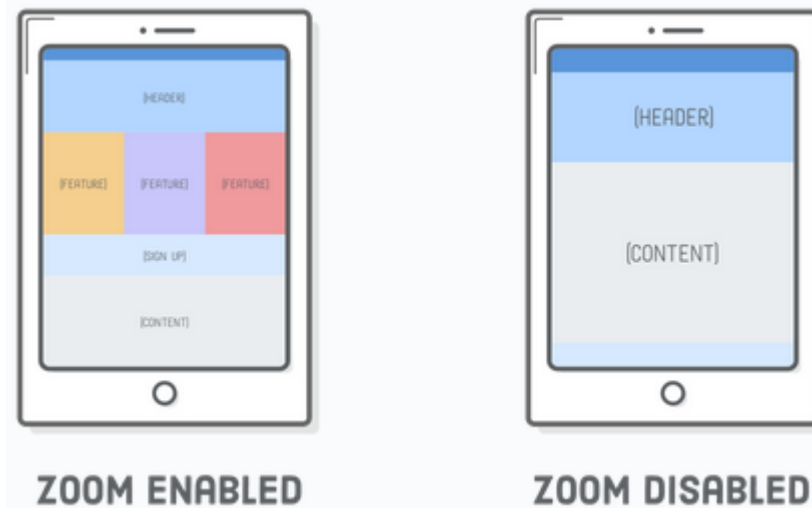
- É possível modificar o layout dependendo da orientação do dispositivo:
 - retrato (*portrait*, quando a largura é menor do que a altura);
 - paisagem (*landscape*, quando a largura é maior do que a altura).

```
body {  
    background-color: blue;  
}  
  
@media only screen and (orientation: portrait) {  
    body {  
        background-color: red;  
    }  
}
```

Viewport

- Alguns dispositivos aplicam, por padrão, um zoom às páginas, fazendo com que seja exibida uma versão miniatura da página desktop, ao invés da versão mobile que desenvolvemos.
- Para prevenir este comportamento, acrescente a metatag abaixo à área **<head>** da página:

`<meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0">`



Imagens flexíveis

- Diz-se que uma imagem é responsiva quando ela se adapta em resposta às diferentes características do dispositivo do usuário.
- Para tratar a responsividade do layout, nos preocupamos apenas com a largura do dispositivo (*device width*). Já em relação a imagens, é preciso considerar o *device width*, as dimensões da imagem e a resolução da tela do dispositivo.
- Há 2 preocupações principais:
 - Fazer com que as imagens se adaptem aos layouts mobile, respeitando ao mesmo tempo seu tamanho intrínseco para que não haja distorções;
 - Evitar que o usuário precise fazer o download de grandes arquivos de imagem desnecessariamente.
- O primeiro ponto pode ser alcançado através de mudanças nas dimensões da imagem (propriedades *width* e *max-width*). Já o segundo pode ser alcançado através de diferentes versões do arquivo da imagem, respeitando em cada caso as características do dispositivo (*media features*).

Imagens flexíveis

- width:
 - Define a largura da imagem. A altura será calculada proporcionalmente.
 - Ao definir a largura de uma imagem como 100%, por exemplo, esta ocupará sempre a largura total de seu container.
 - Se este for maior do que a imagem, a mesma extrapolará seu tamanho original.
- max-width:
 - Define a largura máxima que a imagem poderá atingir.
 - A imagem é mostrada integralmente, com dimensões que se adaptam ao tamanho da tela, mantendo a proporção original entre largura e altura.
 - Porém, em telas maiores do que a largura original da imagem, esta não crescerá em largura além de sua dimensão original.

Imagens flexíveis

- Embora funcionem no sentido de flexibilizar as dimensões das imagens para que se adequem às dimensões da *viewport*, as definições em porcentagem não resolvem totalmente o problema.
- Uma imagem de 300KB, por exemplo, continuará sendo carregada com este peso tanto em um desktop com tela grande e conexão banda larga quanto em um smartphone com tela reduzida e conexão precária
 - No smartphone, a imagem será carregada e depois redimensionada

"Art direction"

- Através de *media queries*, podemos servir diferentes imagens ou versões de uma imagem para diferentes dispositivos – técnica também conhecida como “*art direction*”.
- Se a imagem estiver como plano de fundo (o que é indicado para imagens meramente decorativas e que serão ignoradas por mecanismos de busca e tecnologias assistivas em geral), basta alterar as propriedades de background:

```
/* dispositivos menores do que 400px */
#div1 {
    background-image: url('imgB.jpg');
    background-repeat: no-repeat;
    height: 200px;
}
/* dispositivos acima de 400px */
@media only screen and (min-width: 400px) {
    #div1 {
        background-image: url('imgA.jpg');
        height: 600px;
    }
}
```

"Art direction" com <picture>

- Quando a imagem faz parte do conteúdo, transmitindo ou complementando informações, a inclusão de um elemento <picture> possibilita a indicação de um conjunto de imagens das quais apenas uma será apresentada.
- A tag <picture> encapsula duas outras tags: <source> e
- A tag <source> carrega imagens condicionalmente, baseado em media queries. Atributos:
 - srcset: URL da imagem a ser exibida
 - media: aceita os tipos de media queries válidos na CSS
- A tag funciona como um *fallback*, isto é, seu src carregado em navegadores que não suportam o elemento <picture> ou quando nenhum <source> for aplicado.

"Art direction" com <picture>

```
<picture>
  <source media="(min-width: 800px)" srcset="imgA.jpg">
  <source media="(min-width: 400px)" srcset="imgB.jpg">
  
</picture>
```

- O navegador utilizará os valores dos atributos para carregar a imagem mais apropriada (que será o primeiro <source> cujas condições forem atendidas, sendo os demais ignorados).
- O elemento vem por último dentro de um elemento <picture>
- **Importante:** <picture> não é um elemento criado para substituir o elemento . Quando houver apenas uma imagem a ser mostrada, deve-se usar apenas o elemento como de costume, sendo desnecessário colocá-lo dentro de um elemento <picture>.