



Estácio

CSS

Web Responsiva

Desenvolvimento de páginas conforme características do dispositivo.



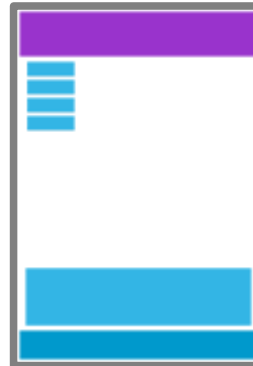
- **Web Responsiva**

- Página Web com “boa” apresentação em qualquer dispositivo
- Páginas Web podem ser visualizadas em diferentes dispositivos: laptops, tablets, smartphones, ...
- A página Web deve adaptar-se para cada tipo de dispositivo
- Utiliza apenas HTML e CSS

- **Web Responsiva**



Desktop



Tablet



Smartphone

CSS – Web Responsiva

- Viewport

- Área visível de uma página Web
- Dimensões variam com o dispositivo
- Antes dos *tablets* e *smartphones* era comum termos páginas com elementos de tamanhos fixos
- Navegando nestas páginas em dispositivos móveis não era uma experiência agradável
- Então naqueles dispositivos os navegadores reduziam toda a página para caber na tela, simulando no dispositivo uma largura maior.
- HTML5 introduziu um método para deixar web designers ter controle sobre o *viewport* , através da tag `< meta>`

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

- `width=device-width` → define a largura da página para seguir a largura do dispositivo
- `initial-scale=1.0` → define o zoom inicial para 1.0, ou seja, o dispositivo não irá reduzir o zoom para alterar a apresentação.
- Você deve incluir em todas as suas páginas web

CSS – Web Responsiva

- Viewport - Exemplos

Sem controle do viewport

```
<!DOCTYPE html>
<html>
<head><meta charset="UTF-8"></head>
<body>
  
<p>
  Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue dui dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim placerat facer possim assum. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim placerat facer possim assum.
</p>
</body>
</html>
```

Com controle do viewport

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<style>
  img { max-width:100%; }
</style>
</head>
<body>
  
<p>
  Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue dui dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim placerat facer possim assum. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim placerat facer possim assum.
</p>
</body>
</html>
```

CSS – Web Responsiva

- Viewport - Exemplos

Sem controle do viewport



Com controle do viewport



- **Grid-View**

- Muitas páginas são baseadas em *grids*, significa que são divididas em colunas.
- Utilizar *grids* facilita bastante o projeto de páginas de uma aplicação, ajudando no posicionamento dos elementos
- Um grid-view responsivo tem, na maioria das implementações, 12 colunas e ocupa toda a largura do *viewport*
- Para construir um grid-view responsivo primeiro temos que garantir que todos os elementos incluem na definição de largura o *padding* e a borda
 - Regra CSS
 - * { box-sizing: border-box; }

CSS – Web Responsiva

- **Grid-View**

- Para construir uma página com 12 colunas:

- Primeiro devemos calcular a porcentagem da largura para cada coluna: $100\% / 12 \text{ columns} = 8.33\%$.
 - Então definimos classes que representam as possíveis larguras dos elementos

- `.col-1 {width: 8.33%;}`
 - `.col-2 {width: 16.66%;}`
 - `.col-3 {width: 25%;}`
 - `.col-4 {width: 33.33%;}`
 - `.col-5 {width: 41.66%;}`
 - `.col-6 {width: 50%;}`
 - `.col-7 {width: 58.33%;}`
 - `.col-8 {width: 66.66%;}`
 - `.col-9 {width: 75%;}`
 - `.col-10 {width: 83.33%;}`
 - `.col-11 {width: 91.66%;}`
 - `.col-12 {width: 100%;}`

- Todas estas colunas devem estar flutuando a esquerda

- ```
[class*="col-"] {
 float: left;
}
```



# CSS – Web Responsiva

- **Grid-View**

- Cada linha será representada por 1 elemento, por exemplo uma `<div>`, e deve conter exatamente as 12 colunas

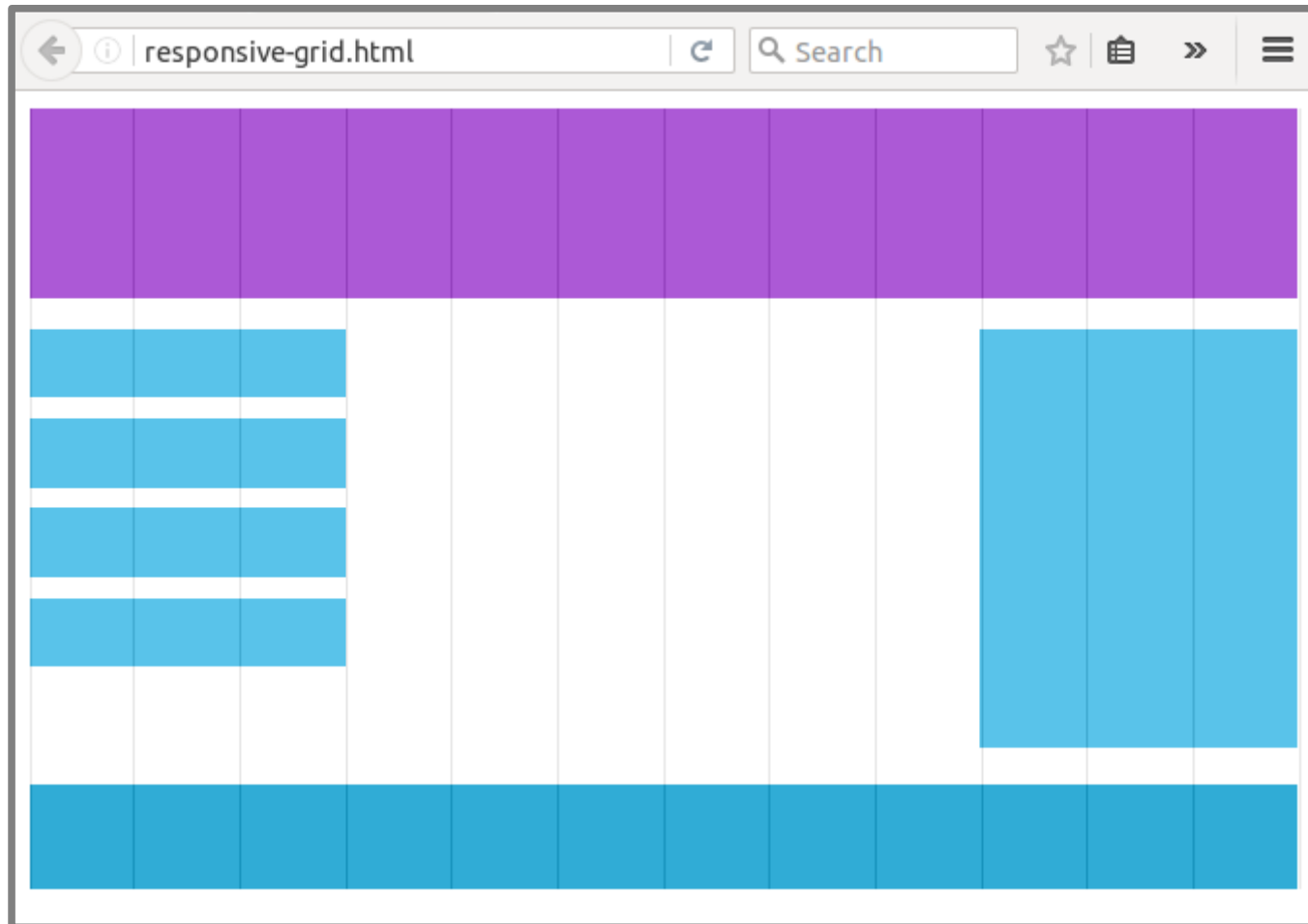
```
<div class="row">
 <div class="col-3">...</div>
 <div class="col-9">...</div>
</div>
```

- As colunas dentro de uma linha estão flutuando, portanto, são levados para fora do fluxo normal da página. Assim os próximos elementos serão colocados como se essas colunas não existissem, para evitar isso temos esta solução:

```
.row::after {
 content: "";
 clear: both;
 display: block;
}
```

# CSS – Web Responsiva

- **Grid-View**
  - **Exemplo:**



# CSS – Web Responsiva

- **Media Query**

- Media Query é uma técnica introduzida no CSS3
- É utilizada para indicar um conjunto de regras CSS somente se certas condições forem verdadeiras

- Ex.:

```
@media only screen and (max-width: 500px) {
 body {
 background-color: lightblue;
 }
}
```

- Media queries permitem adicionar comportamentos diferentes para certas partes do layout baseados em determinadas condições, tais como:
  - altura e largura do viewport;
  - altura e largura do dispositivo;
  - orientação ( tablets e smartphones ); e
  - resolução.

- **Media Query**

- Media queries in CSS3 ampliam a ideia presente no CSS2: Em vez de olhar para o dispositivo, vamos olhar para as características do dispositivo.
- Uma media query consiste de uma tipo de mídia e pode conter uma ou mais expressões:

```
@media not|only mediatype and (expressions) {
 CSS-Code;
}
```

```
<link rel="stylesheet" media="mediatype and|not|only (expressions)" href="print.css">
```

- Mais informações em : [http://www.w3schools.com/css/css3\\_mediaqueries.asp](http://www.w3schools.com/css/css3_mediaqueries.asp)

# CSS – Web Responsiva

- CSS3 Media Types

Valor	Descrição
all	Utilizado para todos os tipos de mídia
print	Utilizado para impressoras
screen	Utilizado para monitores, tablets, smartphones, etc
speech	Utilizado para leitores de tela que lêem a página em voz alta

- Características dos Media Types

- [http://www.w3schools.com/cssref/css3\\_pr\\_mediaquery.asp](http://www.w3schools.com/cssref/css3_pr_mediaquery.asp)

# CSS – Web Responsiva

- **Media Query**

- Exemplos:

- Alterar a cor de fundo dependendo da largura da tela

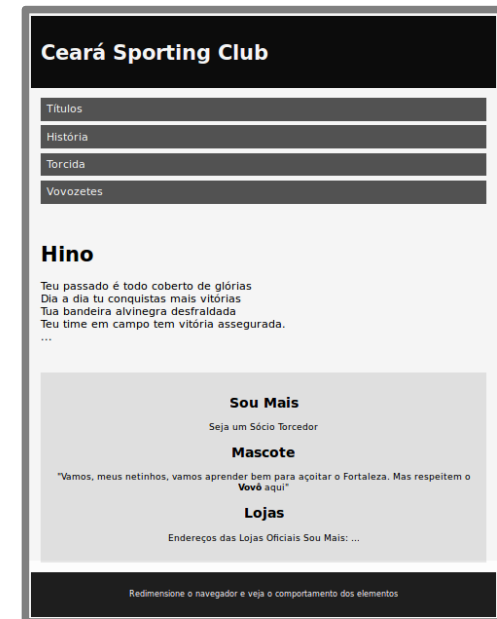
```
<style>
body {
 background-color:lightgreen;
}
@media only screen and (max-width: 500px) {
 body {
 background-color:lightblue;
 }
}
</style>
```

- Alterar a largura das colunas em um grid dependendo da largura da tela

```
@media only screen and (max-width: 768px) {
 /* For mobile phones: */
 [class*="col-"] {
 width: 100%;
 }
}
```

# CSS – Web Responsiva

- Media Query
  - Exemplos:



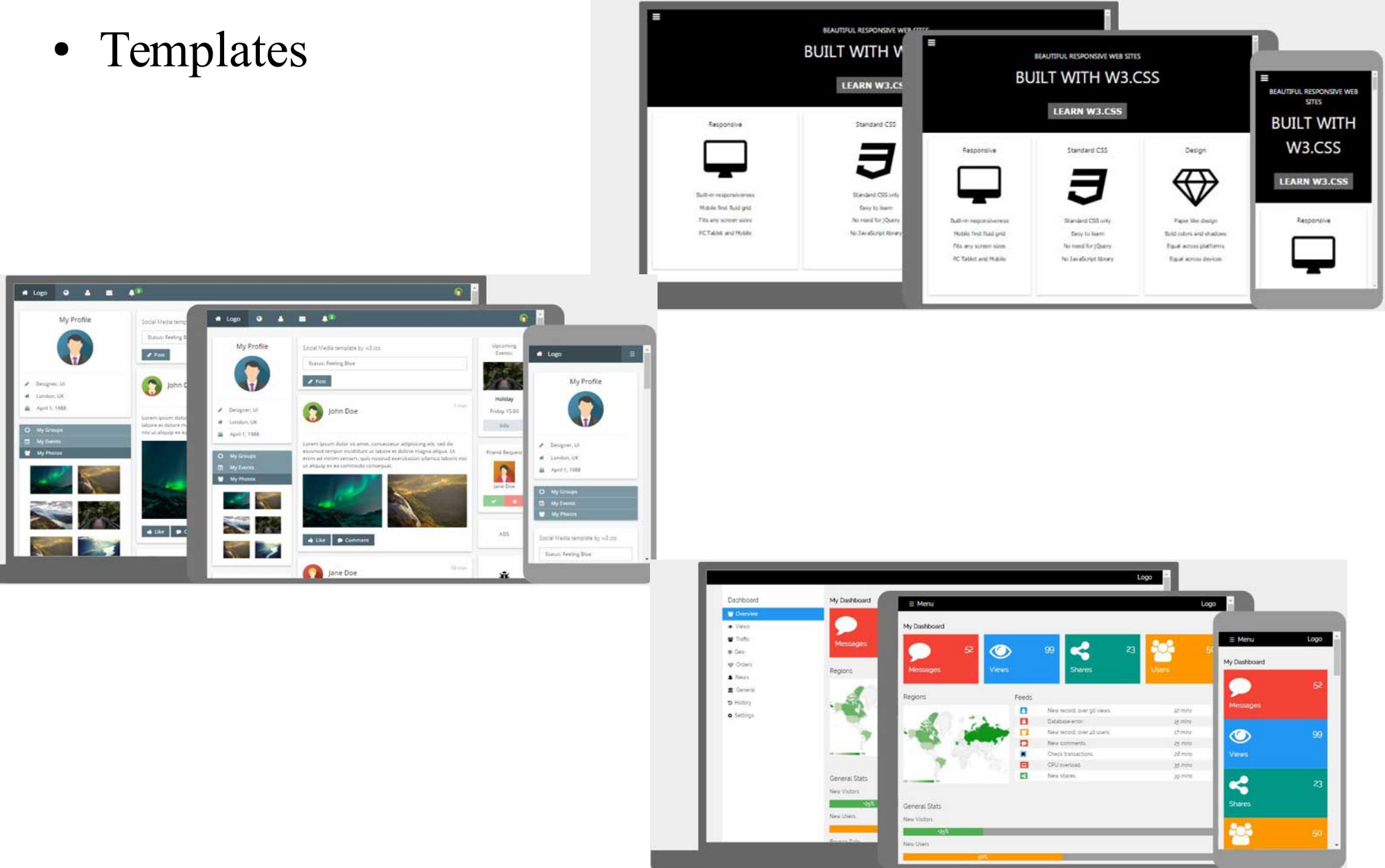
# CSS – Web Responsiva

- Frameworks para desenvolvimento responsivo
  - W3.CSS → <http://www.w3schools.com/w3css/default.asp>
  - Bootstrap → <http://www.w3schools.com/bootstrap/default.asp>
  - Skeleton → <http://getskeleton.com/>



# CSS – Web Responsiva

- Templates



# CSS - Exercícios

- Escreva Páginas Web Responsivas
- Estude:
  - [http://www.w3schools.com/css/css3\\_mediaqueries\\_ex.asp](http://www.w3schools.com/css/css3_mediaqueries_ex.asp)

- Referências de Estudo
  - Material da Estácio
  - <http://www.w3schools.com/css/>
  - <http://www.w3schools.com/css/exercise.asp>
  - Apostila css-web.pdf