



PWR

Programação Web Responsiva

AULA 08

Profs. Adriano Milanez, Roberto Leminski e Ricardo Veras

Agenda

- **Seletores de Classe**
- **Seletores de Id**
- **Diferença entre os Seletores Class e Id**
- **As Tags `<div>` e ``**
 - **A tag `<div...>`**
 - **A tag `<span...>`**
- **Definindo múltiplos seletores**
- **Position**

Seletores de Classe

Seletores de Classe

Até agora, utilizamos o **CSS** para formatar **elementos** de um documento, de forma que as **tags** usadas como **seletores** de CSS (**interno ou externo**), definiam a formatação para **todos** os elementos equivalentes, presentes naquela página.

```
p {  
    color: red;  
}  
  
h1 {  
    color: blue;  
    font-size: 42px;  
}
```

Neste caso **todos os parágrafos** (<p>) da página terão cor vermelha, e **todos os cabeçalhos do tipo <h1>** da página terão cor azul e tamanho de fonte 42 pixels.

Seletores de Classe

Imagine agora a situação em que uma determinada página de um site precise ter **duas formatações** diferentes em dois **parágrafos** (<p..>) do documento:

- um parágrafo **alinhado à direita**.
- outro parágrafo **alinhado ao centro**.

Poderíamos até resolver esse *impasse* com o **CSS in-line**.

```
...  
<p style="text-align: right;"> ... Texto 01 ... </p>  
...  
<p style="text-align: center;"> ... Texto 02 ... </p>  
...
```

Seletores de Classe

Mas, e se **vários parágrafos** de uma página tivessem que possuir **uma** das características (p.ex. a direita), e **outros vários parágrafos** possuir **a outra** característica (p.ex. central)?

Mais ainda, e se de repente o "dono da página" (o seu *Cliente*) quisesse trocar essa característica, de todos aqueles parágrafos, por uma outra? Qual a extensão do retrabalho?

Então melhor seria criarmos um **CSS Interno** (ou **Externo**, dependendo da extensão de alcance desta formatação) para este tipo de situação.

Seletores de Classe

Mas *como*, numa mesma página criar **dois parágrafos** com **formatações diferentes**, e descrevê-las em um CSS **Interno**, ou **Externo**?

Neste caso, podemos utilizar o **atributo *class***, o qual permite definir **estilos diferentes** que podem ser aplicados **a um mesmo elemento** (uma mesma *tag*).

Exemplo:

```
...  
<p class="direita"> ... Texto 01 ... </p>  
...  
<p class="central"> ... Texto 02 ... </p>  
...  
<p class="central"> ... Texto 03 ... </p>  
...  
<p class="direita"> ... Texto 04 ... </p>  
...
```

Seletores de Classe

Este recurso diferenciará os parágrafos, identificando-os com o *nome da classe*.

Desta forma podemos **separar**, na declaração CSS, como **cada parágrafo** receberá as **formatações**:

- Especificando a **Tag com uma Classe**:

```
p.direita {text-align:right;}  
p.central {text-align:center;}
```

- Especificando **somente a Classe**:

```
.direita {text-align:right;}  
.central {text-align:center;}
```

Atentem que a especificação de uma **Classe** no CSS é feita com o "**ponto**" (.).

Seletores de Classe

Observem, nos exemplos da página anterior que **no primeiro caso**, a formatação descrita será estendida a **todos os Parágrafos** que possuírem aquela classe definida...

```
/* 1º caso: */  
p.nome_da_classe {propriedade:valor;}  
...
```

...enquanto que **no segundo caso**, a formatação descrita será estendida a **toda e qualquer Tag** que possuir aquela classe definida (independentemente da Tag).

```
/* 2º caso: */  
.nome_da_classe {propriedade:valor;}  
...
```

Seletores de Classe

Um outro exemplo:

```
...  
p.cA {text-align:center;}  
  
.cA {color:red;}  
...
```

Percebam que as duas formatações acima (CSS) **não** são **conflitantes**, e portanto, como relacionam a mesma Classe (classe cA), **elas se somarão**.

Mas em que situação, especificamente, elas se somarão?

Seletores de Classe

Um outro exemplo:

```
...  
p.cA {text-align:center;}  
  
.cA {color:red;}  
...
```

Percebam que as duas formatações acima (CSS) **não** são **conflitantes**, e portanto, como relacionam a mesma Classe (classe cA), **elas se somarão**.

Mas em que situação, especificamente, elas se somarão?

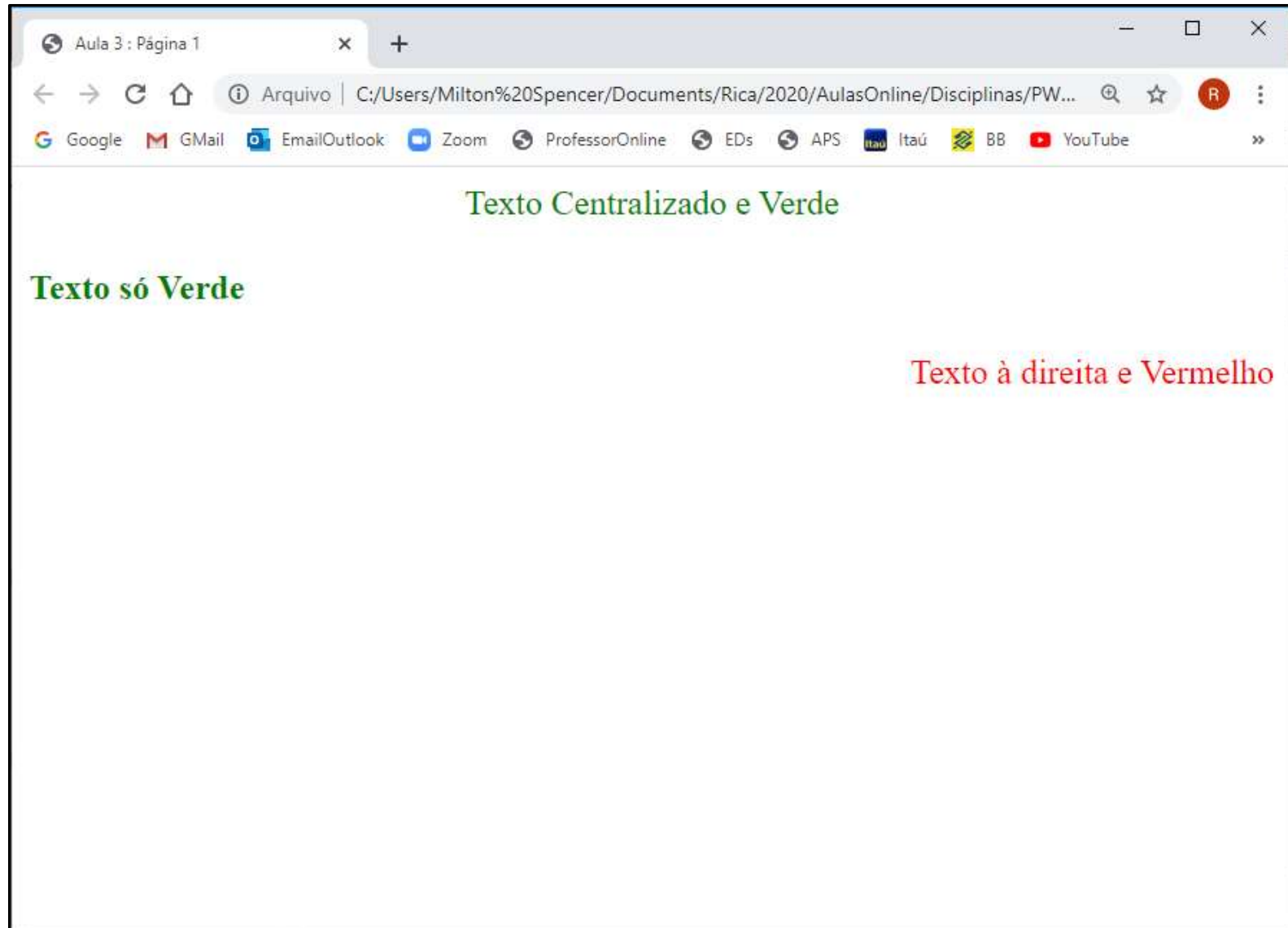
[...a conteúdos de tags <p> com a classe "cA"]

Seletores de Classe

Exemplo (continuação):

```
<html>
<head>
  <style>
    ...
    p.cA {text-align:center;}
    p.cB {text-align:right; color:red;}
    .cA  {color:green;}
    ...
  </style>
</head>
<body>
  ...
  <p class="cA"> Texto Centralizado e Verde </p>
  ...
  <h4 class="cA"> Texto só Verde </h4>
  ...
  <p class="cB"> Texto à direita e Vermelho </p>
  ...
</body>
</html>
```

Seletores de Classe



Seletores de Classe

Observações:

O atributo **class** (como qualquer outro atributo), só pode ser especificado **uma única vez** num dado **elemento**, ou seja, não pode ser algo como :

```
<p class="direita" class="amarelo">
```

No entanto, podemos atribuir **várias Classes** a uma **mesmo elemento**, especificando-os num mesmo atributo class. Para isso separamos as Classes por um espaço:

```
<p class="direita amarelo">
```

Seletores de Id

Seletores de Id

Este recurso, assim como os seletores de classes, diferenciara as tags, **identificando-as** com o **nome do id**.

No entanto **apenas 1(uma) tag** da página podera ter aquele **id**.

```
...  
<p id="identif01"> ... Texto 01 ... </p>  
...  
<p id="identif02"> ... Texto 02 ... </p>  
...  
<p id="identif03"> ... Texto 03 ... </p>  
...  
<p id="identif04"> ... Texto 04 ... </p>  
...
```


Seletores de Id

No CSS, especificamos os *ids* da seguinte forma:

- Especificando a **Tag com um Id**:

```
p#identif01 {color:blue;}  
p#identif02 {color:red; text-align:center;}
```

- Especificando **somente o Id**:

```
#identif01 {color:blue;}  
#identif02 {color:red; text-align:center;}
```

Atentem que a **especificação** de um **Id** no **CSS** é feita com o "*hashtag*" (#).

Diferença entre os Seletores *Class* e *Id*

Diferença entre os Seletores *Class* e *Id*

Utilizando Classes:

- Pode-se usar a **mesma classe** para **vários elementos**;
- Pode-se usar **várias classes** para um **mesmo elemento**;

(Normalmente utilizado para definir formatação **CSS**)

Utilizando Ids:

- **Cada elemento** pode conter **apenas um ID**;
- Cada página pode ter **apenas um elemento com aquele ID**;

(É utilizado mais com **JavaScript**, mas também pode ser utilizado com **CSS**)

**As tags <div> e **

As tags <div> e

A principal função do HTML é dar *significado* ao *conteúdo* (as tags definem um significado – função ou formatação – ao conteúdo interno a elas).

Enquanto a maioria das tags HTML aplicam algum significado (<p> quer dizer parágrafo, <h1> quer dizer cabeçalho, <a...> gera um link, etc.), as tags e <div> não possuem um significado específico, mas *definem uma área* que poderá *receber algum significado*.

No entanto isso se dá com a *utilização conjunta* dos atributos *class*, *id*, ou *style* (atributo do in-line).

A tag <div...>

A tag <div...>

A tag `<div...> ... </div>` é um *elemento de bloco*, e é utilizada para **agrupar blocos maiores de código** (geralmente divisões específicas da página como menus, cabeçalhos, etc.).

Esta tag define uma divisão, ou uma sessão, no documento HTML.

É utilizada **frequentemente com CSS** para **formatar áreas inteiras de um documento**.

Por default (padrão), existe uma **quebra de linha antes e depois** do elemento `<div>`.

A tag <div...>

Exemplo:

```
<div style="background-color:black; color:white;">  
  <p>Isso é maluco!</p>  
  <p>Show!</p>  
</div>
```

Ou ainda:

```
<div style="background-color:lightblue">  
  <h3>Isto é um cabeçalho</h3>  
  <p>Isto é um parágrafo.</p>  
</div>
```


A tag <span...>

A tag <span...>

A tag `<span...> ... ` é um *elemento inline*, e é utilizado para agrupar elementos de texto de áreas menores em um código (geralmente uma palavra, ou um conjunto de palavras de um texto, etc.).

É utilizada frequentemente com CSS para formatar partes de um parágrafo do documento.

Para esta tag, o browser não insere quebra de linha nem antes nem depois do elemento ``.

Obs.: é muito utilizada também como âncora de páginas (apontadas por links direcionados).

A tag <span...>

Exemplo:

```
<p>Minha mãe tem olhos  
<span style="color:blue;"> azuis </span> e meu pai tem  
olhos <span style="color:darkolivegreen; font-  
weight:bold;"> verde escuro </span>.</p>
```

Ou ainda:

```
<p>Isso é <strong><span style="color:red;"> muito maluco  
</span></strong>!</p>
```

As tags <div...> e <span...>

As tags <div> e utilizadas com os atributos *class* e *id*, definem uma área ou partes de texto, que serão formatadas por CSS *Interno* ou *Externo*. Como por exemplo:

```
...  
<div class="d1">  
<p>Minha mãe tem olhos <span class="c1">azuis</span> e  
meu pai tem olhos <span id="i1">verde escuro</span>.</p>  
</div>
```

...

E no CSS, definido que:

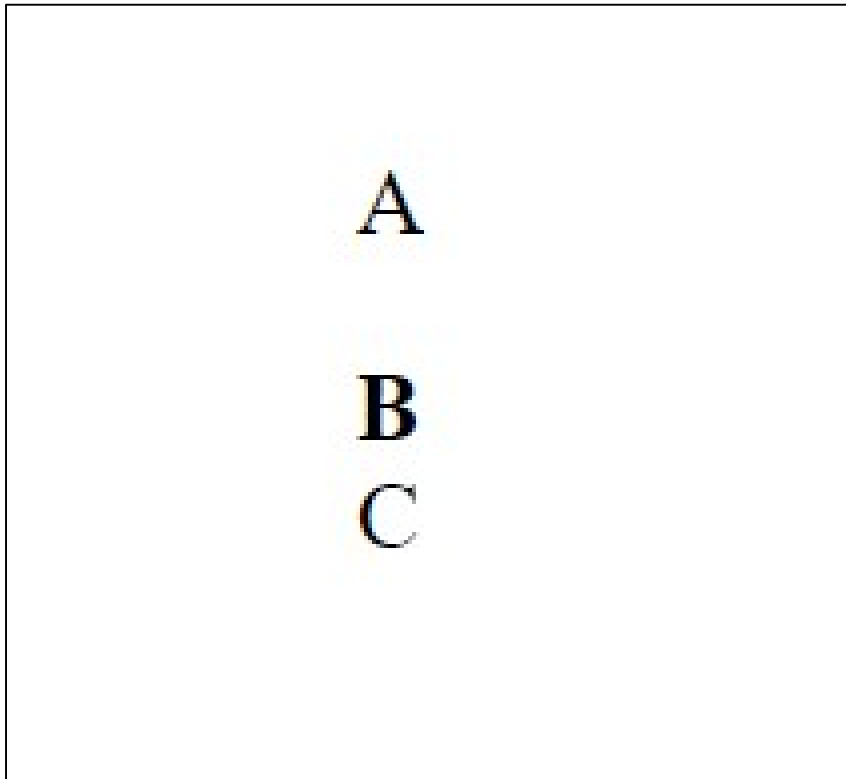
```
...  
.d1 {font-family:verdana;}  
.c1 {color:blue;}  
/* estes dois primeiros referem-se a uma class */  
#i1 {color:darkolivegreen;  
      font-weight:bold;}  
/* este último refere-se a um id */  
...
```

Diferença entre <div...> e <span...>

O exemplo abaixo:

```
...  
<p> A <div style="font-weight: bold;"> B </div> C </p>  
...
```

Resulta em:



Diferença entre <div...> e <span...>

Enquanto que o exemplo abaixo:

```
...  
<p> A <span style="font-weight: bold;"> B </span> C </p>  
...
```

Resulta em:



A B C

Definindo múltiplos seletores

Definindo múltiplos seletores

Quando precisamos **focar** a aplicação de determinada **formatação**, podemos definir **múltiplos seletores** no CSS, de forma a **indicar** exatamente **onde serão aplicadas** aquelas características.

Utilizando vírgula:

```
h1, h2, p, a {color: red;}
```

...significa que **todas aquelas tags** receberão a formatação indicada, ou seja, terão textos em cor vermelha.

Utilizando espaço:

```
table#t1 a img {border:solid 8px yellow;}
```

...significa que a **imagem** que está **dentro** de uma tag de **link**, **dentro** de uma **tabela** cujo **id é "t1"**, terá a borda amarela de linha sólida com 8 pixels.

Position

A Propriedade Position

- Ao trabalhar com a propriedade **position** podemos utilizar as propriedades **top**, **bottom**, **right** e **left** para definir onde o elemento ficará na página.
 - Pode-se dizer que essas outras propriedades definem as "coordenadas" da posição do elemento na página.

Position - Static

position: static;

Posição controlada segundo o "fluxo" normal do documento (em sua posição fixada na página html), e portanto sua posição não será afetada. Este é o valor padrão de posição (caso esta propriedade não seja determinada)

Para este tipo, os estilos top, bottom, right e left simplesmente não são levados em consideração.

Position - Relative

position: relative;

Posição controlada pelas propriedades left, right, top e bottom (ainda dentro do fluxo do documento).

Neste caso haverá um posicionamento apenas do elemento controlado, seguindo as "coordenadas" definidas nas propriedades top, bottom, right e left.

Position - Absolute

position: absolute;

Posição fixa **em uma nova camada**, que não segue o fluxo do documento. Neste caso, todos os outros elementos da página assumem posições como se o elemento controlado não existisse (outra camada).

Um elemento que possui o position como **absolute** terá uma posição relativa a um elemento ancestral (se baseando no elemento pai que possuir um position como relative). Se não houver nenhum elemento antes dele que possua esse tipo de "position", a posição será relativa ao próprio navegador.

Neste caso, se a barra de rolagem for acionada, o elemento controlado sofrerá deslocamento.

Position - Fixed

position: fixed;

Posição fixa na janela, ocupando uma nova camada e que não obedece o deslocamento da barra de rolagem (ele estará sempre visível na tela, na posição definida inicialmente).

Neste caso, todos os outros elementos da página assumem posições como se o elemento controlado não existisse (outra camada).

Position - Sticky

position: sticky;

Posição fixa na janela, que não obedece ao deslocamento da barra de rolagem (ele estará sempre visível na tela, na posição definida inicialmente).

Diferentemente do tipo "fixed", os outros elementos da página assumem posições como se o elemento controlado estivesse ainda na posição original, definida na página html.

Obrigado !