



Prof. Taciano Balardin  
tacianobalardin@gmail.com



**E-MAIL:**

**[taciano@ulbra.edu.br](mailto:taciano@ulbra.edu.br)**



**SITE DA DISCIPLINA:**

**<http://www.taciano.pro.br/>**

**SENHA:**

**[@quarta](#)**

CSS

Introdução

Conceitos Básicos

# FUNDAMENTOS PROFISSIONAIS

## AULA 07

# Trabalho Formulários

- Não recebi o trabalho de:
  - Hiury Marques;
  - José Lisboa;
  - Rafael Alves.

# Habilidades e Competências:

- **Competências Tecnológicas e de Gestão**
  - Compreender a dinâmica empresarial decorrente de mercados mais exigentes e conscientes de seus direitos e das novas necessidades sociais, ambientais e econômicas;
  - Participar do desenvolvimento e implantação de novos modelos de competitividade e produtividade nas organizações;
  - Diagnosticar e mapear, com base científica, problemas e pontos de melhoria nas organizações, propondo alternativas de soluções baseadas em sistemas de informações;
  - Planejar e gerenciar os sistemas de forma a alinhá-los aos objetivos estratégicos de negócio das organizações.

# Habilidades e Competências:

- **Competências Tecnológicas e de Gestão**
  - Modelar, especificar, implementar, implantar e validar os sistemas de informações;
  - Auxiliar os profissionais das outras áreas a compreenderem a forma com que os sistemas de informação podem contribuir para as áreas de negócio;
  - Participar do acompanhamento e monitoramento da implementação da estratégia da organização, identificando as possíveis mudanças que podem surgir pela evolução da tecnologia.

# Habilidades e Competências:

## ■ Competências Humanas

- Expressar idéias de forma clara, empregando técnicas de comunicação apropriadas para cada situação;
- Flexibilidade suficiente para se adaptar e absorver as rápidas mudanças do mercado de trabalho e da tecnologia da informação;
- Participar e conduzir processos de negociação para o alcance de objetivos;
- Habilidade de relacionamento interpessoal, favorável ao trabalho em equipe;
- Capacidade de utilização dos conhecimentos obtidos para um desempenho profissional bem sucedido, com base nos **valores éticos** que norteiam a vida em sociedade.

# Plágio / Cópia de Trabalhos

- No **Código Penal Brasileiro**, em vigor, no Título que trata dos **Crimes Contra a Propriedade Intelectual**, nós nos deparamos com a previsão de crime de violação de direito autoral – artigo 184 – que traz o seguinte teor: *Violar direito autoral: **Pena – detenção, de 3 (três) meses a 1 (um) ano, ou multa.*** E os seus parágrafos 1º e 2º, consignam, respectivamente:
- §1º Se a violação consistir em reprodução, por qualquer meio, com intuito de lucro, de obra intelectual, no todo ou em parte, sem autorização expressa do autor ou de quem o represente, (...): Pena – reclusão, de 1 (um) a 4 (quatro) anos, e multa, (...).
- § 2º Na mesma pena do parágrafo anterior incorre quem vende, expõe à venda, aluga, introduz no País, adquire, oculta, empresta, troca ou tem em depósito, com intuito de lucro, original ou cópia de obra intelectual, (...), produzidos ou reproduzidos com violação de direito autoral.

[Leia mais no site do Superior Tribunal de Justiça](#)



# Qual o problema com o HTML?

- Desejamos criar uma aplicação Web, quais seriam os passos?
  - Definimos o conteúdo que vamos querer exibir;
  - Definimos a estruturação do conteúdo na tela;
  - Fazemos uso de tags para definirmos a hierarquia do conteúdo (tags <h>, por exemplo);
  - Definimos a formatação a ser adotada (cores, tamanhos de fonte, espaçamentos, etc).
- Qual o problema desta abordagem?
  - Estamos acoplando o conteúdo a ser exibido com a sua formatação!
  - ... considere um site que todos os anos altera o seu layout

# CSS: A solução

- Separa a marcação do HTML da apresentação do site;
- Agora a responsabilidade do HTML é apenas estruturar o conteúdo, enquanto que o visual é de responsabilidade do CSS;
- HTML marca e estrutura textos, cabeçalhos, parágrafos, links, botões, formulários, imagens e demais elementos da página;
- CSS define cores, posicionamento na tela, estilos de linhas, bordas e tudo mais relacionado a apresentação.

# CSS: A solução

- No início era ... uma confusão!

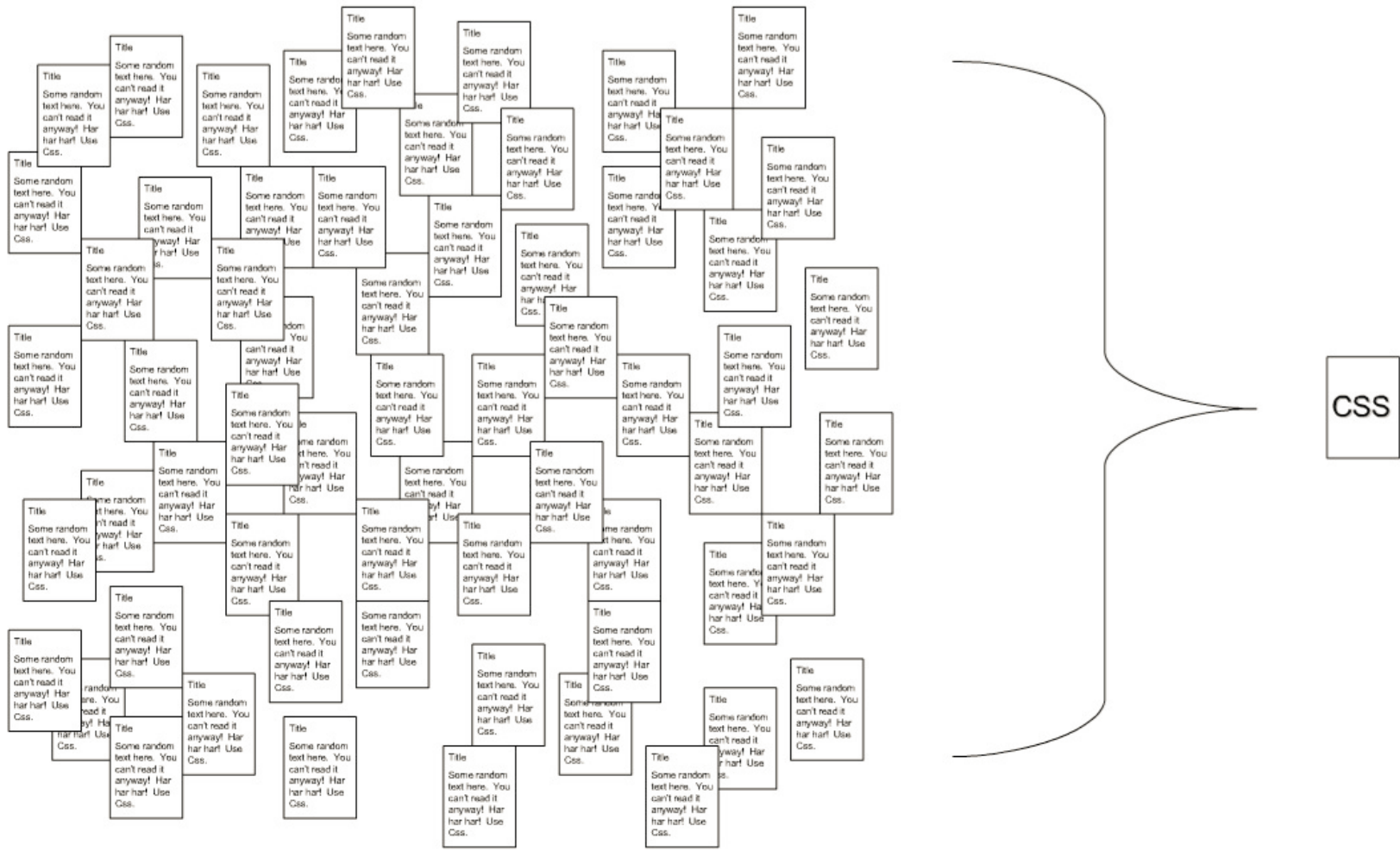


HTML  
Styles...  
JavaScript

# CSS: A solução

- Atualmente, organização em “camadas”

# CSS: A solução



# CSS: O que é?

- Acrônimo de **C**ascading **S**tyle **S**heets;
- Define como os elementos HTML serão exibidos;
- Podem ser definidos de três formas:
  - ***Inline:***
    - Para modificações totalmente localizadas.
  - **Incorporada:**
    - Para modificações específicas de uma página.
  - **Externa:**
    - Uso geral.



# Estilos *Inline*

- Estilos *Inline* são definidos dentro das *tags*.
- A maioria dos benefícios são perdidos
  - Mistura conteúdo e apresentação
  - Sempre que a página for carregada, CSS é relido
  - Parâmetro **style**

...

```
<body>
```

```
<p style="color: #000000; margin: 5px;">
```

Parágrafo na cor preta com margem de 5 pixels.

```
</p>
```

```
</body>
```

...

# Estilos Incorporados

- Estilos Incorporados são definidos no documento.
  - Boa parte dos benefícios são perdidos;
  - Mistura conteúdo e apresentação (menos que *inline*).
- Sempre que a página for carregada o CSS é relido.
- Sintaxe:
  - Tag **<style>**
  - Dentro da seção **<head>...</head>**



# Estilos Incorporados

```
...  
<head>  
...  
<style type="text/css">  
  body {  
    background: #000000;  
    url(imagens/minhaimagem.gif);  
  }  
  h3 {  
    color: #0000FF;  
  }  
  p {  
    margin-left: 20px;  
  }  
</style>  
...  
</head>  
...
```

**Tudo que estiver aqui  
dentro será considerado  
definição de estilo**

# Estilos Externos

- Arquivo .CSS com os estilos:
  - Separa conteúdo e apresentação;
  - Uso de um mesmo CSS para várias páginas;
  - Carregamento de novas páginas reaproveita CSS já carregado.
- Sintaxe:
  - Uso da tag `<link />`
  - Dentro da seção `<head>...</head>`

# Estilos Externos

...

<head>

...

```
<link rel="stylesheet" type="text/css" href="arquivo.css" />
```

...

</head>

...

# A regra CSS e sua sintaxe

- Uma regra CSS é uma declaração que segue uma sintaxe própria e que define como será aplicado estilo a um ou mais elementos HTML;
- Um conjunto de regras CSS formam uma Folha de Estilos;
- Uma regra CSS compõe-se de três partes: um **seletor**, uma **propriedade** e um **valor** e tem a seguinte sintaxe:

**seletor** { **propriedade**: **valor**; }

# Definindo os Elementos

- **Seletores:** é o elemento HTML para qual a regra será válida:
  - Pode ser uma tag, uma classe, ou por uma ID, etc;
  - Por exemplo: <p>, <h1>, <form>, .minhaclasse, etc...).
- **Propriedade:** é o atributo do elemento HTML ao qual será aplicada a regra (por exemplo: font, color, background, etc...);
- **Valor:** é a característica específica a ser assumida pela propriedade.

**A atribuição dos valores às propriedades é feito através do operador ':' e não do operador '='**

# Exemplos

- Elementos 'p' e 'body' customizados:

```
p {  
    font-size: 12px;  
}
```

```
body {  
    color: #000000;  
    background: #FFFFFF;  
    font-weight: bold;  
}
```

- Alteração da fonte para o conteúdo do elemento 'h3':

```
h3 {  
    font-family : "Comic Sans MS";  
}
```

# Comentários

- É possível introduzir comentários ao longo das regras CSS;
- Para isso, coloca-se entre os elementos `/* */`

```
seletor {  
    propriedade: valor;  
    /* O que estiver aqui não será considerado */  
}
```

# Agrupamento de Seletores

- Muitas vezes precisamos definir a mesma regra para vários seletores;
- Deste modo, é necessário agrupar todos os elementos em uma única regra.
- Exemplo de agrupamento de seletores:

```
h1 , h2 , h3 , h4 , h5 , h6 {  
    color: #00FF00;  
}
```



# O seletor class

- Até agora vimos a aplicação de estilos as tags;
- Mas quais são as limitações disso?
- Não é possível definir diferentes estilos ao mesmo tempo.
- Nestes casos fazem-se uso dos seletores **class** e **id**.

# O seletor class

- O seletor class permite que você defina seus próprios identificadores que são utilizados nas regras de estilo.

```
elemento.minhaclasse {  
    propriedade: valor;  
}  
p.cor-um {  
    color:#000;  
}  
p.cor-dois {  
    color: #0ff;  
}
```

# Usando o seletor class

CSS:

```
p.cor-um {  
    color:#000;  
}  
p.cor-dois {  
    color: #0ff;  
}
```

HTML:

```
<p class="cor-um">este parágrafo será na cor preta.</p>
```

```
<p class="cor-dois">este parágrafo será na cor azul.</p>
```

# Estendendo o seletor class

- Ao criar uma classe você talvez queira que ela seja aplicável a qualquer elemento HTML.
- Neste caso basta que se omita o nome do elemento antes da classe.

```
.cor-tres {  
    color: #00f;  
}
```

```
<p class="cor-tres"> conteúdo </p>
```

```
<span class="cor-tres"> conteúdo </p>
```

# O seletor id

- O seletor ID difere do seletor class, por ser ÚNICO.
- Um seletor ID de determinado nome só pode ser aplicado a UM e somente UM elemento HTML dentro do documento.

```
#meuID {  
    propriedade: valor;  
}
```

```
p#cor {  
    color: #00f;  
}
```

# Usando o seletor id

```
p#cor {  
    color: #00f;  
}
```

<p id = "cor"> parágrafo na cor vermelha </p>

# id x class: quando usar

- O seletor id deve ser usado quando **apenas 1 elemento** deverá ser alterado.
- O elemento class será utilizado quando **diversos** elementos serão customizados.

# Conflito entre estilos

- O que acontece quando dois estilos são aplicados ao mesmo elemento em escopos diferentes?
  - Há uma situação de conflito!
  - Neste caso é utilizado o efeito Cascata (Cascade)

**É utilizado o valor mais interno!**



# Conflito entre estilos, exemplo:

- Se você, ao escrever sua folha de estilo declarar na segunda linha do arquivo uma regra assim:

Linha 2    p {font-size:14px;}

- a seguir continuar escrevendo outras regras e lá embaixo na trigésima linha escrever:

Linha 30   p {font-size:12px;}

- cria um conflito entre duas regras para o tamanho da fonte nos parágrafos.

Qual será o tamanho da fonte nos parágrafos do documento?

**Resposta: 12px.**

# Conflito entre estilos !important

- Se você, ao escrever sua folha de estilo declarar na segunda linha do arquivo uma regra assim:

Linha 2    p {font-size:14px !important;}

- a seguir continuar escrevendo outras regras e lá embaixo na trigésima linha escrever:

Linha 30   p {font-size:12px;}

- cria um conflito entre duas regras para o tamanho da fonte nos parágrafos.

Qual será o tamanho da fonte nos parágrafos do documento?

**Resposta: 14px.**

# Conflito entre estilos

- Uma das diretrizes do efeito cascata estabelece que a regra de estilo **escrita depois** (ou posteriormente) no fluxo da folha de estilos, **prevalece** sobre aquelas escritas anteriormente.
- Outro exemplo:
- Um estilo **inline** sempre vai prevalecer os **incorporados e externos** caso não haja nenhuma exceção com **!import**.

# Customizando a fonte

- As propriedades para as fontes, definem as características das letras que constituem os textos dentro dos elementos HTML.
  - color: cor da fonte
  - font-family: tipo da fonte
  - font-size: tamanho da fonte
  - font-style: estilo da fonte
  - font-variant: fontes maiúsculas de menor altura
  - font-weight: nível de negrito

# Color

- Existem três formas de representar cores:
  - código hexadecimal: #FFFFFF;
  - código rgb: rgb(255,235,0);
  - nome da cor: red, blue, green, etc.



Saiba mais  
sobre cores

...

```
<body>
```

```
<p style="color:#FF0000">
```

Cor definida por um valor hexadecimal

```
</p>
```

Cor definida por um valor hexadecimal

```
<p style="color:rgb(0,255,0)">
```

Cor definida por um valor rgb

```
</p>
```

Cor definida por um valor rgb

```
<p style="color:blue">
```

Cor definida pelo nome

```
</p>
```

Cor definida pelo nome

```
</body>
```

...

# Font-family

- Define o nome das fontes a serem utilizadas

```
<style type="text/css">  
  h2 { font-family: arial, helvetica, serif; }  
  p { font-family: blabla, calibri; }  
</style>
```

## Header 2 com fonte definida

Parágrafo com fonte definida.

# Font-size

- Especifica o tamanho da fonte:
  - xx-small
  - x-small
  - small
  - medium
  - large
  - x-large
  - xx-large
  - length: uma medida reconhecida pelas CSS (px, pt, cm,%,etc.)

# Font-size

<p style="font-size:16px;"> Parágrafo com fonte definida. </p>

Parágrafo com fonte definida.

<p style="font-size:25px;"> Parágrafo com fonte definida. </p>

Parágrafo com fonte definida.

<p style="font-size:small;"> Parágrafo com fonte definida. </p>

Parágrafo com fonte definida.

<p style="font-size:large;"> Parágrafo com fonte definida. </p>

Parágrafo com fonte definida.



# Font-style

- Define estilos de fontes como itálico, obliquo:

```
<style type="text/css">
```

```
h1 { font-style: italic; }
```

```
h2 { font-style: normal; }
```

```
p { font-style: oblique; }
```

```
</style>
```

***Texto no Header 1***

**Texto no Header 2**

*Texto no parágrafo*

# Font-weight

- Define a intensidade da cor (nível de negrito):
  - font-weight: normal
  - font-weight: bold

<h2 style='font-weight:normal'>Texto font-weight: normal</h2>

<h2 style='font-weight:bold'>Texto font-weight: bold</h2>

Texto font-weight: normal

**Texto font-weight: bold**

# Formatação de texto

- Além das fontes, é possível realizar diversas customizações no texto propriamente dito:
  - **letter-spacing:** espaçamento entre letras;
  - **word-spacing:** espaçamento entre palavras;
  - **text-align:** alinhamento;
  - **text-decoration:** decoração de texto (underline, overline, blink etc);
  - **text-transform:** transformação de texto (maiúsculo para minúsculo e vice-versa).

# Letter-spacing

- Espaçamento entre letras:
  - normal: é o espaçamento default;
  - São válidos valores negativos!

```
<style type="text/css">
  h2 { letter-spacing: 0.5cm; }
  p.espaco { letter-spacing: 2px; }
</style>
```

```
<h2>Header 2</h2>
```

```
<p class='espaco'>Parágrafo com
  espaçamento entre letras</p>
```

```
<p>Parágrafo sem espaçamento</p>
```

**H e a d e r 2**

Parágrafo com espaçamento entre letras

Parágrafo sem espaçamento

# Word-spacing

- Espaçamento entre palavras:
  - Mesma estrutura do espaçamento entre letras.

```
<style type="text/css">
  h2 { word-spacing: 0.5cm; }
  p.espaco { word-spacing : 2px; }
</style>
```

```
<h2>Header 2</h2>
<p class='espaco'>Parágrafo com
  espaçamento entre palavras</p>
<p>Parágrafo sem espaçamento</p>
```

**Header 2**

Parágrafo com espaçamento entre palavras

Parágrafo sem espaçamento

# Text-align

- Define o estilo de alinhamento:

- **left**

```
<style type="text/css">
```

```
  p { text-align:left; }
```

```
</style>
```

Com css você pode definir o alinhamento do texto utilizando a propriedade text-align e definindo um valor entre left, right, center e justify. Com css você pode definir o alinhamento do texto utilizando a propriedade text-align e definindo um valor entre left, right, center e justify.

- right
  - center
  - justify

# Text-align

- Define o estilo de alinhamento:

- left

- **right**

```
<style type="text/css">
```

```
  p { text-align:right; }
```

```
</style>
```

Com css você pode definir o alinhamento do texto utilizando a propriedade text-align e definindo um valor entre left, right, center e justify. Com css você pode definir o alinhamento do texto utilizando a propriedade text-align e definindo um valor entre left, right, center e justify.

- center

- justify

# Text-align

- Define o estilo de alinhamento:

- left
- right
- **center**

```
<style type="text/css">
```

```
p { text-align:center; }
```

```
</style>
```

Com css você pode definir o alinhamento do texto utilizando a propriedade text-align e definindo um valor entre left, right, center e justify. Com css você pode definir o alinhamento do texto utilizando a propriedade text-align e definindo um valor entre left, right, center e justify.

- justify



# Text-align

- Define o estilo de alinhamento:
  - left
  - right
  - center
  - **justify**

```
<style type="text/css">
```

```
p { text-align:justify; }
```

```
</style>
```

Com css você pode definir o alinhamento do texto utilizando a propriedade text-align e definindo um valor entre left, right, center e justify. Com css você pode definir o alinhamento do texto utilizando a propriedade text-align e definindo um valor entre left, right, center e justify.

# Atividades



- Na página:
  - Exercício 03 – Parte 1.