

**INSTITUTO FEDERAL**  
Triângulo Mineiro

Campus  
Patrocínio

# LABORATÓRIO DE PROGRAMAÇÃO II

GILBERTO VIANA DE OLIVEIRA

# Introdução

---

- ❖ Ao escrevermos uma página em HTML, buscamos definir a melhor *tag* possível para cada parte do nosso conteúdo.
- ❖ Cada *tag* possui uma configuração padrão diferente. Por exemplo: Um conteúdo que esteja dentro de um `<h1>` é mostrado em fonte maior e negrito, se comparado a um conteúdo dentro de um `<p>`.

# Introdução

---

- ❖ Como discutido anteriormente, eram utilizadas algumas propriedades para alterar a formatação do nosso conteúdo dentro do próprio HTML.

```
<h1><font color="blue">Money, que é good nós num  
have...</font></h1>
```

# Introdução

---

- ❖ Uma analogia muito utilizada para a introdução ao CSS é a seguinte:
- ❖ O HTML é o esqueleto da nossa página.
- ❖ O CSS será a “skin”, aparência externa vista.
- ❖ Outras tecnologias como o javascript serão os sistemas que compõe o funcionamento do corpo, como o sistema nervoso, por exemplo.



# Cascade Style Sheets

---

- ❖ Linguagem separada do HTML, utilizada para a estilização das nossas páginas.
- ❖ Utilizando o CSS, toda a formatação será independente do nosso conteúdo. Cada tipo de documento será responsável por sua parte.



# CSS

- ❖ É possível inserirmos uma configuração CSS dentro de um elemento, mas isso, na prática, não nos ajuda a manter a distribuição entre o conteúdo mostrado e sua formatação.
- ❖ Esse tipo de declaração é denominada declaração ***INLINE***

```
<h1 style="color:red">Money, que é good nós num  
have...</h1>
```

# CSS

- ❖ Outra forma de incluirmos o CSS é adicionarmos a *tag* `<style>` no `<head>` de nossa página e declarar as propriedades para os elementos dentro dessa *tag*.
- ❖ Esse método é mais eficiente que o anterior.
- ❖ Essa declaração é chamada de “***tag style***”

```
<style>  
  h1{  
    color:yellow;  
  }  
</style>
```

# CSS

---

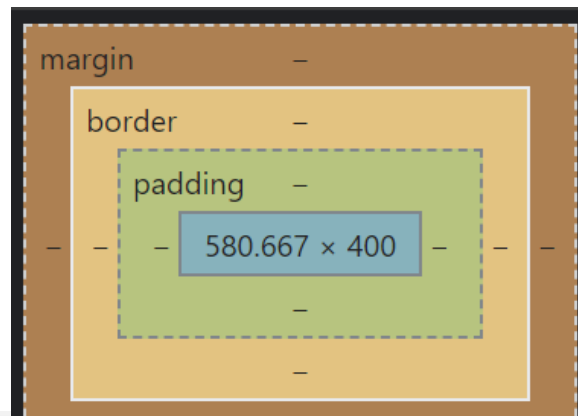
- ❖ Nosso código CSS pode aumentar bastante conforme adicionamos estilos a muitos elementos diferentes.
- ❖ Ao invés de replicarmos o código em cada página, aumentando o tamanho do documento, podemos criar um documento externo e importá-lo.
- ❖ Esse tipo de declaração é chamado de “***link tag***”.

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



# CSS Box Sizing

- ❖ O motor de renderização do navegador representa cada elemento como uma caixa retangular, de acordo com o padrão definido pelo CSS conhecido como box (caixa) model.
- ❖ O conteúdo do elemento é uma das quatro partes que compõem o box, sendo as demais o seu preenchimento, borda e margem.



# CSS – Unidades de medidas

- ❖ Além de medirmos os espaçamentos e tamanhos em pixels, também podemos utilizar:
  - ❖ **cm**: centímetros.
  - ❖ **mm**: milímetros.
  - ❖ **in**: polegadas.
  - ❖ **pt**: pontos (1 ponto 1/72 polegadas).
  - ❖ **%**: Porcentagem.
  - ❖ **em** e **rem**: medida tipográfica. Parecidas com % mas ao invés de serem baseadas no tamanho de um elemento, essas medidas são baseadas em tamanho de fonte. EM usa o tamanho da fonte do elemento pai e REM usa o tamanho da fonte do <html>.
  - ❖ **vh** **vw**: A medida vh é igual a 1/100 da altura da viewport. Então, por exemplo, se a altura do navegador é 900px, 1vh equivale a 9px.

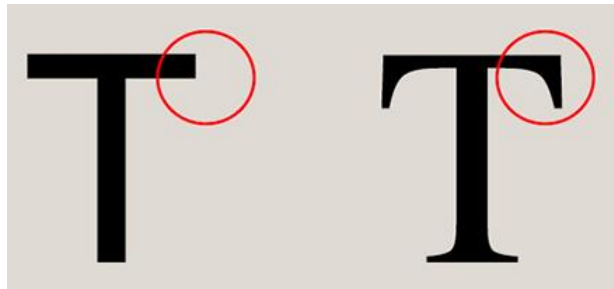
# Exemplo de margin

---

```
/* Aplica para todos os quatro lados */  
margin: 1em;  
  
/* vertical | horizontal */  
margin: 5% auto;  
  
/* topo | horizontal | inferior */  
margin: 1em auto 2em;  
  
/* topo | direita | inferior | esquerda */  
margin: 2px 1em 0 auto;
```

# CSS – Exemplos: Fontes

- ❖ Uma das formas mais básicas de personalizarmos nossas páginas, é através da alteração da fonte.
- ❖ Para alterarmos essa propriedade, utilizamos `font-family`.
- ❖ Os navegadores por padrão apresentam o conteúdo em fontes “serif”, que possuem detalhes em suas terminações.



# CSS - Fontes

---

- ❖ Podemos procurar por fontes instaladas no computador para exibir o texto de nossa página.

```
body {  
    font-family: "Arial", sans-serif;  
}
```

- ❖ Nesse caso, o navegador verifica se a fonte “Arial” está disponível para renderizar o conteúdo de tudo que for um elemento dentro de *body*.

# CSS - Fontes

- ❖ Outra forma de alterarmos a fonte é mantermos um arquivo com a fonte escolhida dentro do nosso projeto.
- ❖ Dessa forma podemos adicionar a fonte ao CSS.

```
@font-face{  
    font-family: 'pecita';  
    src: url("fonts/Pecita.otf");  
}
```

```
body {  
    font-family: 'pecita', sans-serif;  
}
```

# CSS - Fontes

- ❖ Também podemos buscar uma fonte de algum repositório na web. Por exemplo: <https://fonts.google.com/>
- ❖ Ao selecionarmos uma fonte, o site gera o código em CSS que devemos utilizar.



# CSS – Alinhamento do texto

---

- ❖ Podemos utilizar algumas propriedades para alinharmos e configurarmos o espaçamento do texto:

```
p{  
    text-align:center;  
    line-height: 2px; /* tamanho da altura de cada linha */  
    letter-spacing: 2px; /* tamanho do espaço entre cada letra */  
    word-spacing: 3px; /* tamanho do espaço entre cada palavra */  
    text-indent: 20px; /* tamanho da margem da primeira linha do texto */  
}
```



# CSS - Cores

- ❖ Para configurarmos as cores em CSS podemos utilizar diversas formas.
- ❖ Podemos declamar o valor da cor pelo nome, valor em hexadecimal ou valor pela intensidade das cores red, green e blue.
- ❖ [http://www.w3schools.com/cssref/css\\_colors.asp](http://www.w3schools.com/cssref/css_colors.asp)



# CSS - Cores

---

- ❖ Para o uso da função `rgb()`, você deve especificar a quantidade de intensidade de cada cor (entre 0 e 255).
- ❖ Podemos ainda utilizar a função `rgba`, que também possui um parâmetro para definir a opacidade da cor (1 para totalmente opaco).

Também existem as funções `hsl` e `hsla` (hue, saturation e lightness)\*

```
color: rgba(0,0,0,1);
```

# CSS – Imagem de fundo

---

- ❖ Podemos utilizar a propriedade `background-image` para adicionar um arquivo de imagem no fundo de um elemento.

```
background-image: url(img/grunge_wall.png);
```

- ❖ Podemos alterar a propriedade `background-repeat` (que vem ativada por padrão).
- ❖ Também podemos alterar a posição das imagens com a propriedade `background-position` (left top, top, center, etc).

# CSS - Bordas

---

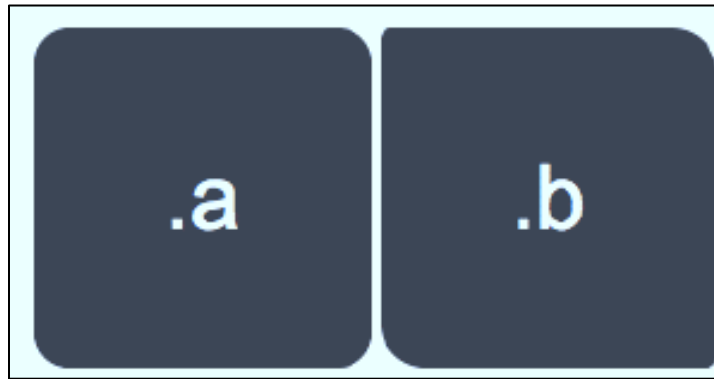
- ❖ A propriedade borda pode ser atribuída a um elemento pelo CSS. Podemos definir a cor, o estilo e a largura de uma borda.

```
body {  
    border-color: blue;  
    border-style: solid;  
    border-width: 1px;  
}
```

- ❖ No CSS3 foi adicionado a possibilidade de bordas arredondadas através da propriedade `border-radius`.

# CSS - Bordas

## ❖ Exemplo:



```
.a{  
    border-radius: 15px;  
}  
.b{  
    border-radius: 5px 20px;  
}
```

## ❖ É possível estilizar cada borda com um *radius* diferente.

# CSS – Usando seletor de ID

---

- ❖ O atributo ID é um dos mais importantes no HTML. Ele possui um “peso” grande dentro da construção do documento.
- ❖ Não podemos ter mais de um elemento com o mesmo ID em um documento HTML.
- ❖ Para usarmos esse atributo, basta fazermos a declaração no HTML.

```
<h1 id="meu-titulo">Título da página</h1>
```

# CSS – Usando seletor de ID

- ❖ Para estilizarmos esse elemento que possui o id único, devemos usar o caractere “#” antes do nome atribuído ao ID. No CSS, fica assim:

```
<style>
#meu-titulo {
  background-color: lightblue;
  color: black;
}
</style>
```

**Título da página**

- ❖ Nesse caso, as propriedades **não serão** aplicadas a outros elementos “h1”, apenas ao de id=“meu-titulo”.

# CSS – Usando seletor de Classe

- ❖ Já o atributo **class**, pode ter múltiplos elementos em um documento HTML.
- ❖ Assim como o ID, elementos que tiverem atribuição de uma determinada classe, podem ter suas propriedades alteradas de forma específica no CSS (usamos o ‘.’ seguido do nome da classe).
- ❖ Exemplo:
  - ❖ Assuma que queremos aplicar a mesma propriedades para diferentes elementos que são do mesmo tipo em uma página. Primeiro, definimos esse tipo (classe) no CSS como sendo, neste caso, “cidade”, então atribuímos essa classe para todos os elementos que compartilham essas características no HTML.



# CSS – Usando seletor de Classe

## ❖ Exemplo - Código

```
.cidade {  
  background-color: tomato;  
  color: white;  
  border: 2px solid black;  
  margin: 10px;  
  padding: 10px;  
}
```

```
<div class="cidade">  
<h2>Londres</h2>  
<p>Londres é a capital da Inglaterra.</p>  
</div>
```

```
<div class="cidade">  
<h2>Paris</h2>  
<p>Paris é a capital da França.</p>  
</div>
```

```
<div class="cidade">  
<h2>Tóquio</h2>  
<p>Tóquio é a capital do Japão</p>  
</div>
```

### **Londres**

Londres é a capital da Inglaterra.

### **Paris**

Paris é a capital da França.

### **Tóquio**

Tóquio é a capital do Japão

# Atividade

---

- ❖ Os alunos devem se reunir em duplas, e pesquisar sobre o seguinte tema:
  - ❖ Regras de cascata no CSS, como funciona a aplicação das regras de formatação no CSS?
  - ❖ O que é o seletor Global? O seletor por tipo, por ID e por classe?
  - ❖ Algum desses elementos é mais importante que outro? Justifique.

# Dúvidas?

---

Para atendimento envie e-mail  
[gilbertooliveira@iftm.edu.br](mailto:gilbertooliveira@iftm.edu.br)

**MAY THE FORCE  
BE WITH YOU**

