



# Desenvolvimento de Aplicações para Internet

## CSS

**Celso Olivete Júnior**

**olivete@fct.unesp.br**

### ❑ Formulário



The image shows a screenshot of a web form titled "Formulário de contato". The form has a dark background with light-colored text. It contains four input fields: "Nome", "e-mail", "Assunto", and "Conteúdo". The "Conteúdo" field is a larger text area with a vertical scrollbar. At the bottom left, there is a button labeled "Enviar". At the bottom right, there is a note: "\* Campos de preenchimento obrigatório".

Responsável por fazer a interação entre os dados fornecidos pelo usuário (através do navegador) e o servidor.

Principais itens: menu, campos de textos, botões,...

### ❑ Formulário

➤ Tag <FORM> ... </FORM>

➤ Atributos

✓ **ACTION**: Especifica o URL do script ao qual serão enviados os dados do formulário.

✓ **METHOD**: Seleciona um método para acessar o URL de ação

✓ **GET**

✓ os dados enviados fazem parte do URL (endereço) associado à consulta enviada para o servidor;

✓ suporta até 128 caracteres. Ex: `www.xxx.com.br?nome=teste`

✓ **POST**

✓ os dados entrados fazem parte do corpo da mensagem enviada para o servidor;

✓ transfere grande quantidade de dados.

EX: <FORM METHOD="**POST**" action = "pagina.html">

### ❑ Formulário

- Pode conter qualquer formatação - parágrafos, listas, tabelas, imagens - exceto outros formulários
- Geralmente utiliza-se `<INPUT>`, `<SELECT>` e `<TEXTAREA>`.
- **OBS:** todos os campos de entrada devem ter um atributo `NAME` → que será utilizado posteriormente pelo *script*

### ❑ Formulário <INPUT>

- <INPUT>: tem um atributo TYPE, ao qual atribuímos seis valores diferentes para gerar seis tipos diferentes de entrada de dados.

- TIPO TEXTO (1)

Nome: <INPUT TYPE=TEXT NAME="Nome">

Nome:

- TIPO PASSWORD (2)

Password: <INPUT TYPE=PASSWORD NAME="senha">

Password:

### ❑ **Formulário <INPUT>**

#### ➤ **ATRIBUTOS ADICIONAIS** para campos do *tipo texto e password*

- **VALUE** → usado para dar um valor inicial

`<INPUT TYPE=TEXT NAME="nome" VALUE="Seu nome">`

- **SIZE** → especifica o tamanho do espaço no display para o campo

Endereço: `<INPUT TYPE=TEXT SIZE=35>`

- **MAXLENGTH** → número de caracteres aceitos em um campo

Dia do mês: `<INPUT TYPE=TEXT NAME="ex" MAXLENGTH=2>`

### ❑ Formulário <INPUT>

#### ➤ Múltipla escolha

➤ *TIPO CHECKBOX (3)* → insere um botão para escolha de opções. A cada alternativa corresponde um valor a ser manipulado pelo *script* que processa os dados. **NAME** deve ser diferente para cada ítem

➤ `<INPUT TYPE=CHECKBOX NAME="futebol" VALUE="futebol">Futebol`

➤ `<INPUT TYPE=CHECKBOX NAME="volei" VALUE="volei" CHECKED>Vôlei`



Futebol  
 Vôlei

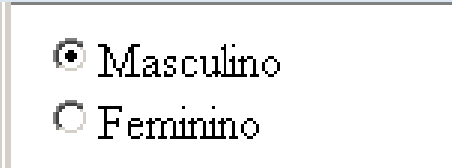
### ❑ Formulário <INPUT>

#### ➤ Escolha única

- *TIPO RADIO (4)* → insere um botão de escolha de valores para uma opção. **NAME** deve ser o mesmo para todos os ítems

```
<INPUT TYPE=RADIO NAME="sexo" VALUE="masc" CHECKED>Masculino <br>
```

```
<INPUT TYPE=RADIO NAME="sexo" VALUE="fem">Feminino
```



```
 Masculino  
 Feminino
```

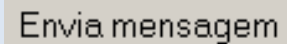


### ❑ Formulário <INPUT>

#### ➤ BOTÕES DE AÇÃO

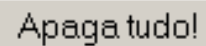
- **SUBMIT** (5) → realiza o envio dos dados para o servidor

```
<INPUT TYPE=SUBMIT NAME=b_enviar VALUE="Envia mensagem">
```



- **RESET** (6) → restaura os valores iniciais das entradas de dados

```
<INPUT TYPE=RESET NAME=b_limpar VALUE="Apaga tudo!">
```



### ❑ Formulário **<SELECT>**

➤ **<SELECT>** apresenta uma lista de valores (menu suspenso) através de campos **OPTION**.

➤ **<SELECT NAME="sabor">**

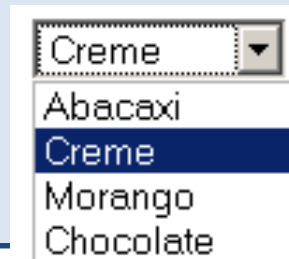
**<OPTION VALUE = "abacaxi">**Abacaxi **</OPTION>**

**<OPTION VALUE="creme" SELECTED >**Creme **</OPTION>**

**<OPTION VALUE = "morango">**Morango **</OPTION>**

**<OPTION VALUE = "chocolate">**Chocolate **</OPTION>**

**</SELECT>**



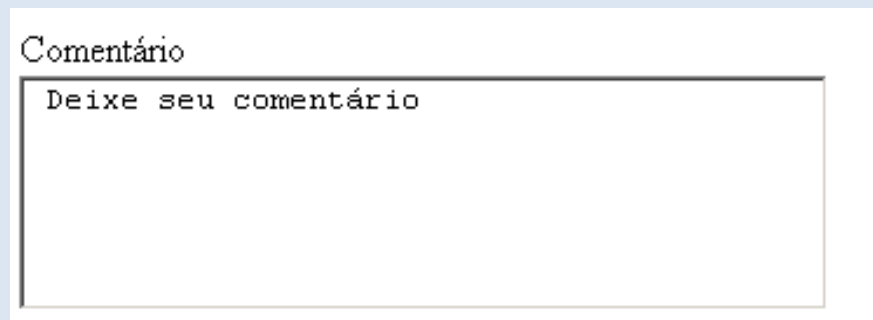
OPCIONAL

VALOR DE RETORNO

### ❑ Formulário **<TEXTAREA>**

- **<TEXTAREA>** abre uma área para entrada de texto de acordo com atributos para número de colunas/linhas e, se for o caso, um valor inicial.

```
<TEXTAREA COLS=40 ROWS=5 NAME="comentario"> Deixe seu  
comentário  
</TEXTAREA>
```



Comentário

Deixe seu comentário

### ❑ Formulário completo

**Cadastro de Conselho**

**Programa**

Unidade

Programa

**Conselho de Programa de Pós-Graduação**

Código

Nome

Tipo

Fundamento



# HTML

## Formatando o documento...

### **Folhas de estilos – CSS**

HTML é responsável pelo trabalho pesado (a estrutura), enquanto CSS dá o toque de elegância (layout).

## □ **CSS.** Vantagens:

- Economizar o seu tempo.
- Diminuir o tamanho do código de sua página.
- Sua página irá carregar mais rapidamente.
- Mais facilidade de manter e fazer alterações na página.
- Maior controle no *layout* da página.



# CSS

## Criando estilos

- ❑ Cada estilo é definido como uma regra CSS.

**elemento {atributo1: valor; atributo2: valor ...}**

**ELEMENTO** → descreve o elemento de design ao qual o estilo será aplicado.

**ATRIBUTO1** → o aspecto específico do elemento que você quer usar como estilo

**VALOR** → a configuração aplicada ao atributo.

- ❑ Estilos CSS são formados por regras que, geralmente, são colocadas em um arquivo separado (.css)
  - **Facilitar a manutenção**
  - Existe também a possibilidade de declarar as regras CSS dentro do próprio arquivo HTML – não indicado!!

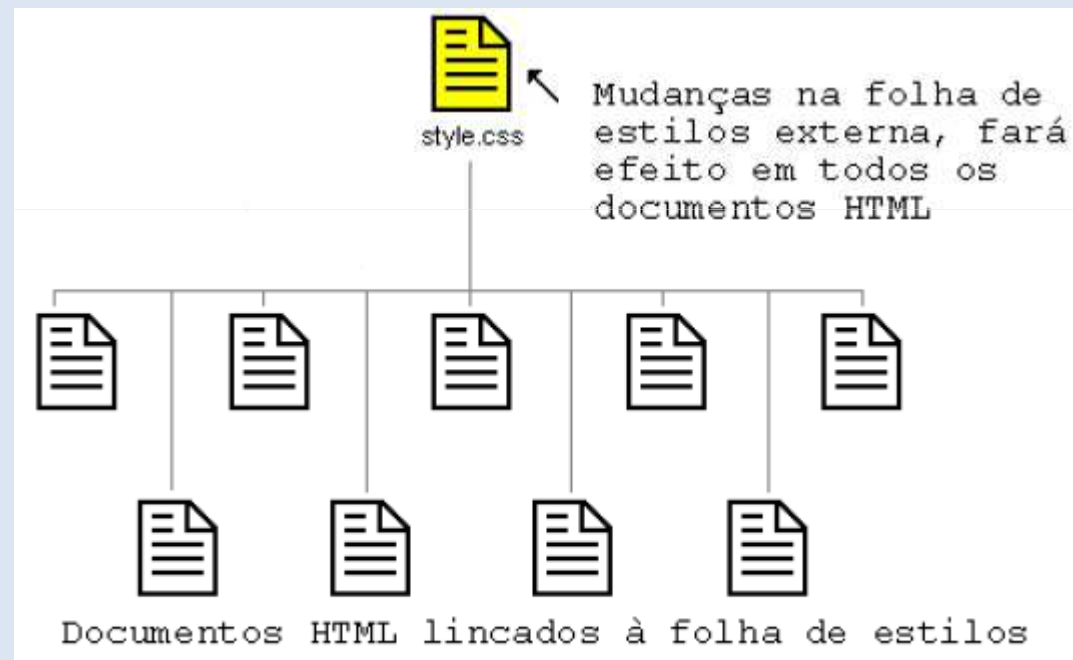




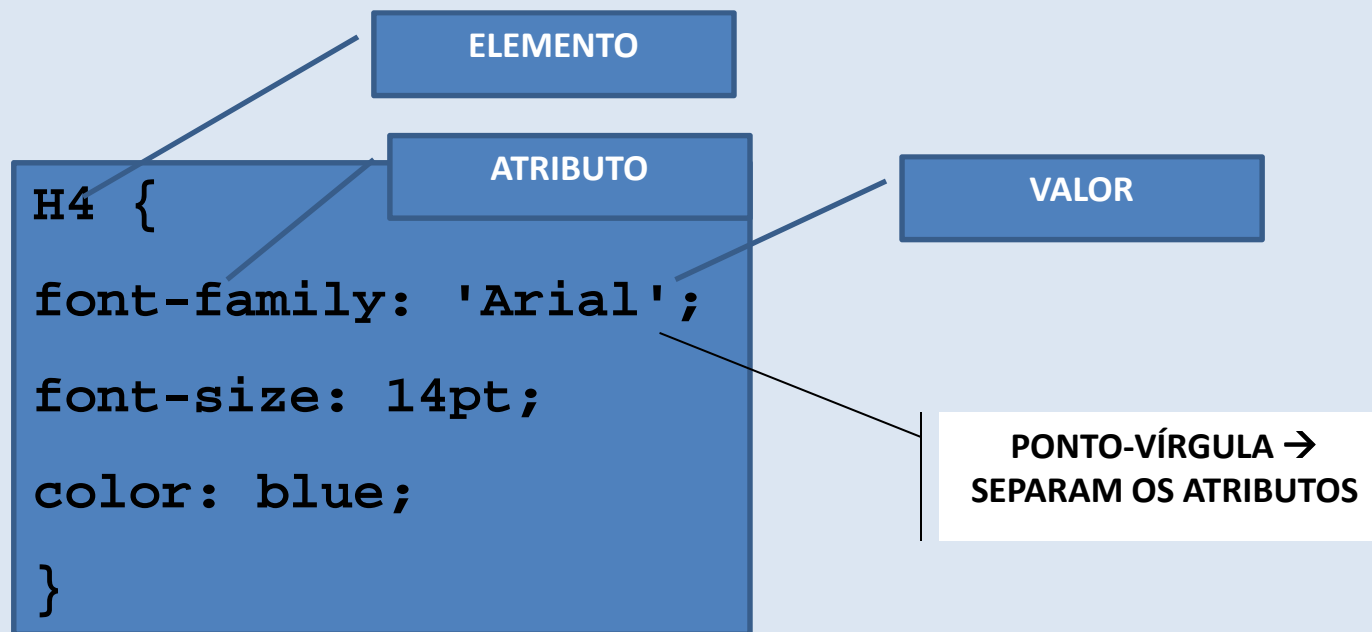
unesp

# CSS

## ❑ CSS em arquivo externo



## ❑ Exemplo de uma regra CSS (config.css)





unesp

## CSS

- ❑ Para facilitar a definição das regras, pode-se utilizar um editor CSS



## ❑ Utilizando a CSS criada

### ✓ Utilizando a CSS criada

- Basta incluir a seguinte TAG dentro de `<HEAD> ... </HEAD>` e informar a localização correta do arquivo

```
<LINK REL="STYLE SHEET" HREF="CSS/Config.css"  
TYPE="text/css">
```

...

```
<h4> Bem vindo ao mundo HTML </h4> ...
```

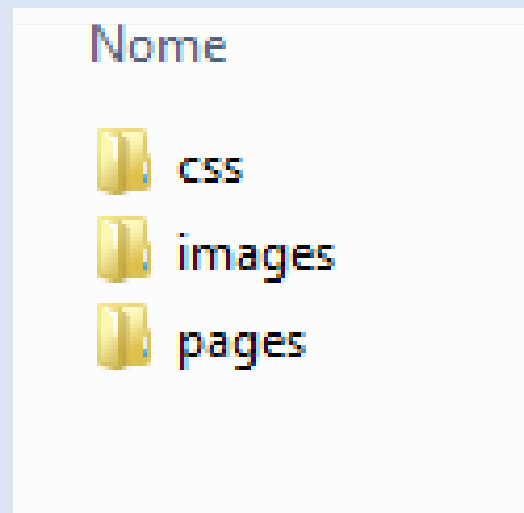
**Bem vindo ao mundo do HTML**



unesp

## CSS

- ❑ Lembre-se de salvar os arquivos CSS no local apropriado (diretório CSS)



## ☐ Trabalhando com cores e fundo

- Color;
- Background-color;
- Background-image;
- Background-repeat;
- Background-attachment;
- Background-position;
- Background.

## ❑ Color

- ❑ A propriedade color define a cor do primeiro plano de um elemento.
- ❑ Considere, por exemplo, que desejamos que todos os cabeçalhos de primeiro nível no documento sejam na cor vermelha **<h1>**

```
h1 {  
  color: #ff0000;  
}
```



## ❑ Background-color

- ❑ Define a cor do fundo de um elemento. O elemento **<body>** contém todo o conteúdo de um documento HTML. Assim, para mudar a cor de fundo da página, devemos aplicar a propriedade **background-color** ao elemento **<body>**.
- ❑ Pode-se aplicar cores de fundo para outros elementos, inclusive para cabeçalhos e textos.



## □ Background-color

```
body {  
  background-color: FFCC66;  
}  
h1 {  
  color: #990000;  
  background-color: #FC9804;  
}
```

- ❑ Background-image: propriedade usada para definir imagens de fundo

```
body {  
background-color: #FFCC66;  
background-image:  
url( "../images/imagem.jpg" );  
}
```



## ❑ Background-repeat

### ➤ imagens de fundo repetidas

Valor	Descrição
Background-repeat: repeat-x	A imagem se repete na horizontal
background-repeat: repeat-y	A imagem se repete na vertical
background-repeat: repeat	A imagem se repete na tanto na horizontal como na vertical
background-repeat: no-repeat	A imagem não se repete

```
body {  
  background-image:  
  url("../images/imagem.jpg");  
  background-repeat: no-repeat;}
```

## ❑ Background-attachment

- ✓ A propriedade background-attachment define se a imagem será fixa ou se irá rolar juntamente com o elemento que a contém.
- ✓ Uma imagem de fundo fixa permanece no mesmo lugar e não rola com a tela ao contrário da imagem que não é fixa e rola acompanhando o conteúdo da tela.



## ❑ Background-attachment

Valor	Descrição
Background-attachment: scroll	A imagem rola com a página
Background-attachment: fixed	A imagem é fixa

## ❑ Background-position

- Por padrão uma imagem de fundo é posicionada no canto superior esquerdo da tela.
- A propriedade background-position permite alterar este posicionamento padrão e colocar a imagem em qualquer lugar na tela.

## ❑ Background-position

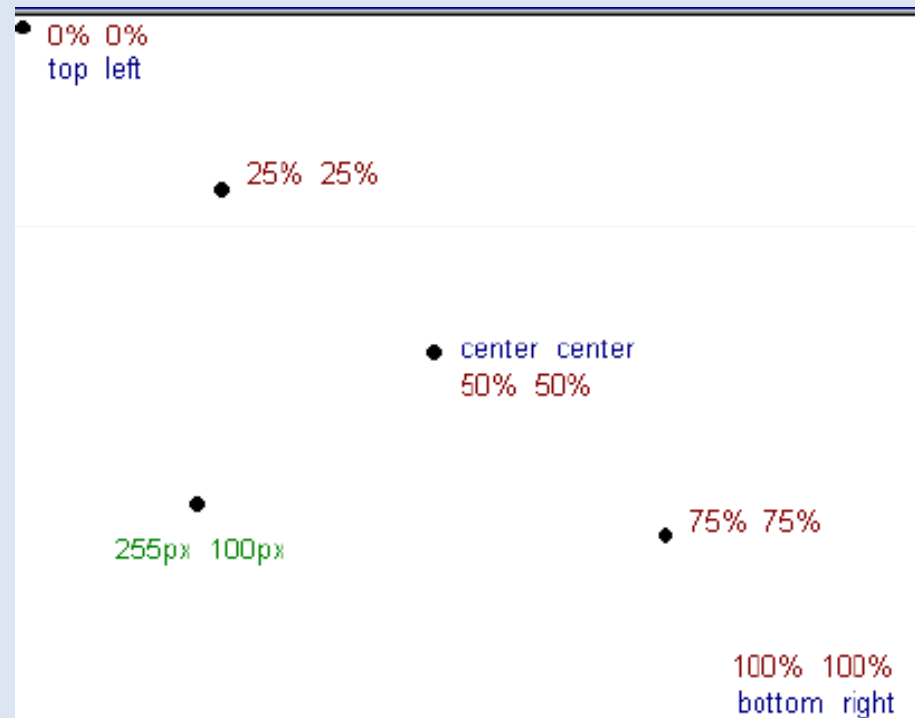
- Define-se o posicionamento através do sistema de coordenadas.
- Por exemplo, os valores '100px 200px' posicionam a imagem a 100px do topo e a 200px do lado esquerdo da janela do navegador.
- As coordenadas podem ser expressas em percentagem da largura da janela, em unidades fixas (pixels, centímetros, etc.) ou pode-se usar as palavras top, bottom, center, left e right.



unesp

# CSS

## ❑ Background-position







## ❑ Background

- ❑ todas estas propriedades podem ser expressas de forma abreviada

```
background-color: #white;  
background-image: url("borboleta.jpg");  
background-repeat: no-repeat;  
background-attachment: fixed;  
background-position: right bottom;
```

- ❑ ficando dessa forma...

```
background: white url("borboleta.jpg") no-repeat fixed right  
bottom;
```

- ❑ seguindo a sintaxe

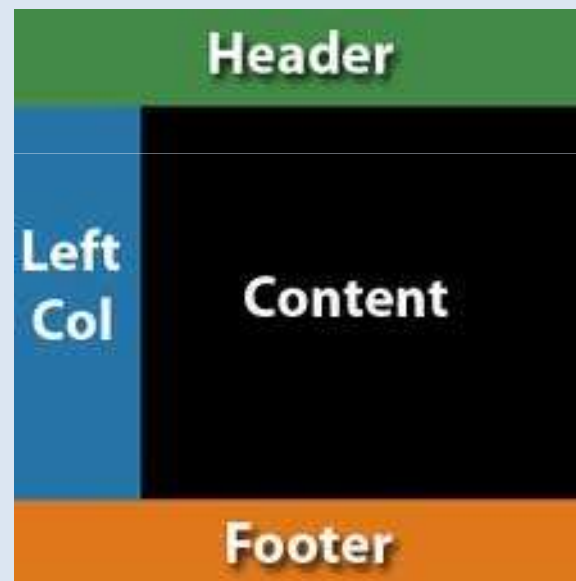
```
[background-color] | [background-image] | [background-repeat]  
| [background-attachment] | [background-position]
```



## Elementos *Div* e *Span*

- ❑ Os elementos **div** e **span** foram criados com o HTML 4 com a finalidade de fornecer um mecanismo genérico para **agrupar** e **prover estrutura** aos documentos. O elemento **div** é um **container nível de bloco** e **span** é um **elemento inline**.
- ❑ Uma **class** é em geral usada para **fornecer** uma **identidade** a uma **div** ou **span** com o propósito de fornecer uma referência para estilização por CSS ou captura por um script.

- ❑ Exemplo de *layout* utilizando **div**



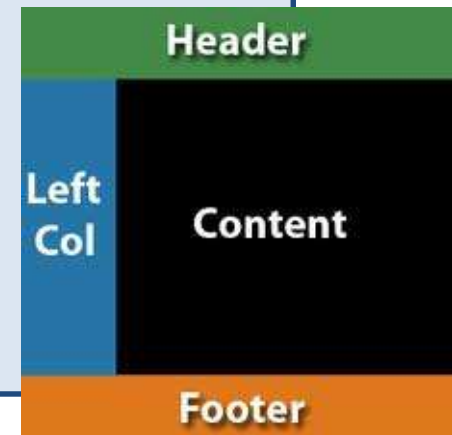
- ❑ As *divs* são definidas com regras CSS



## Definindo a tag *body* CSS

- ❑ Primeiramente é necessário definirmos a área principal do documento → tag **body**

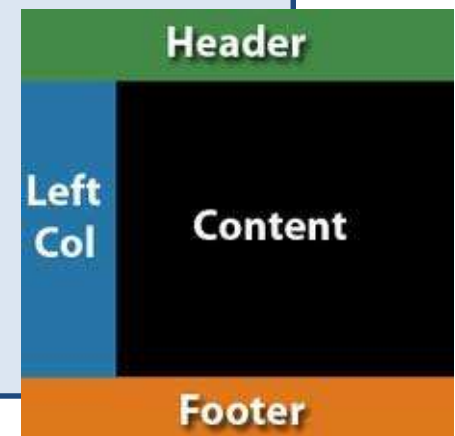
```
body{  
  width:80%;  
  background-color: white;  
  font-size: 25px;  
  color:white;  
}
```





## Definindo a div *Header* CSS

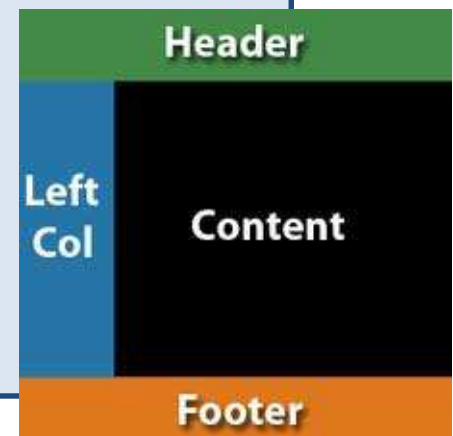
```
.header{  
  width: 80%;  
  background-color:Green;  
  float:right;  
}
```





## Definindo a div *Content* CSS

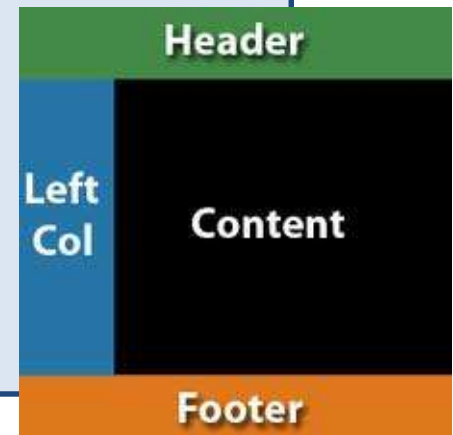
```
.content {  
  width: 70%;  
  background-color: black;  
  float: right;  
}
```





## Definindo a div *Left* *CSS*

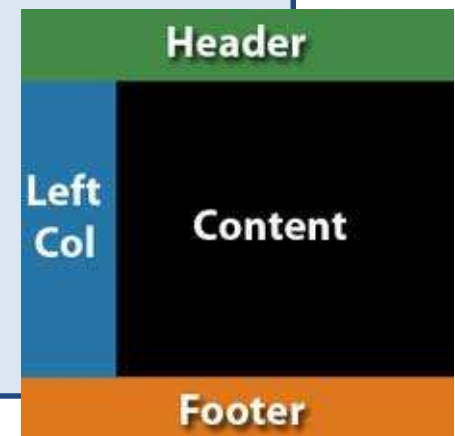
```
.left{  
  width: 10%;  
  background-color: blue;  
  float:right;  
}
```





## Definindo a div *Footer* CSS

```
.footer{  
  width: 80%;  
  background-color:#FFA500;  
  float:right;  
}
```

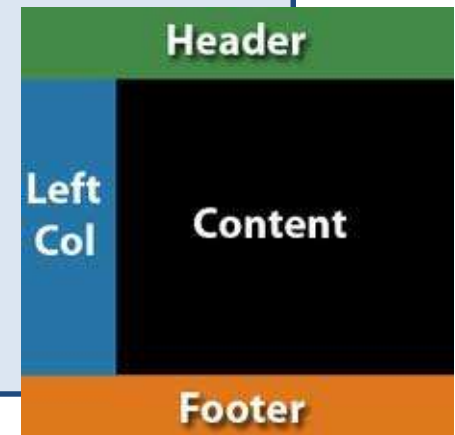






# Definindo o documento *HTML*

```
<BODY>
  <div class="header">
    Logotipo
  </div>
  <div class="content">
    Conteudo
  </div>
  <div class="left">
    Item A <br>
    Item B <br>
  </div>
  <div class="footer">
    Rodape
  </div>
</BODY>
```





## Elementos *Div* e *Span*

- ❑ Os elementos **div** e **span** foram criados com o HTML 4 com a finalidade de fornecer um mecanismo genérico para **agrupar** e **prover estrutura** aos documentos. O elemento **div** é um **container nível de bloco** e **span** é um **elemento inline**.
- ❑ Uma **class** é em geral usada para **fornecer** uma **identidade** a uma **div** ou **span** com o propósito de fornecer uma referência para estilização por CSS ou captura por um script.

## Elemento *Span*

### *Span* - Exemplo de utilização

- Definição de blocos internos ao formulário
- Este mesmo formulário também poderia ter sido formatado utilizando tabelas – **técnica menos usual**



Formulario de Contato :

Nome Completo

Email

Assunto

Opções

Sexo  Masculino  Feminino

Mensagem

Enviar



## Elemento *Span*

- ❑ Definindo duas colunas dentro da *div content*

```
.coluna1{  
  width: 50%;  
  background-color:#9370DB;  
  float:left; /* flutuar para esquerda*/  
}  
.coluna2{  
  width: 50%;  
  background-color:Aqua;  
  float:right; /* flutuar para direita*/  
}
```



## Elemento *Span*

- ❑ Coluna 1 e Coluna 2 definidas com o elemento *span* dentro da *div content*

Logotipo

Item A

Coluna 1

Coluna 2

Item B

Item C

Item D

Rodape

```
<div class="content">
  <span class="coluna1">
    Coluna 1
  </span>
  <span class="coluna2">
    Coluna 2
  </span>
</div>
```



## Elemento *Span*

### ☐ Observações:

1. Dentro do elemento ***span*** podem ser criados, por exemplo, elementos do tipo `input`, `textarea` e `select`
2. Regras CSS podem ser aplicadas aos elementos do tipo `input`, `textarea` e `select`

## Exercício

- Estudar como agrupar elementos com o uso de **classes, id e div**
- Visitem o site <http://www.maujor.com>
- Crie um site utilizando todos os conceitos vistos na aula de hoje (principalmente a parte de formulários) **COM CSS. Pode ser feito em dupla**

## Exercício

- ❑ A interface deve ser **dividida** utilizando DIV e deverá ter um formato seguindo este padrão

