Introdução ao CSS3

Cascading Style Sheets

Everton Vargas Guetierres

ALGUMAS DEFINÇÕES DE TERMOS DO HTML

- Parâmetros são instruções e valores posicionados logo após a abertura das Tags. Normalmente, são formados pelo par (parâmetro="valor"). Pode ser usado mais de um parâmetro.

```
<tag parâmetro="valor">
        Conteúdo
</tag>
```

- Div é uma tag HTML feita com o objetivo de formatar uma sessão de uma página web como se fosse um bloco. Podemos deixa-lo colorido, botar uma imagem de fundo, colocar borda, etc...

- Aninhamento acontece quanto uma tag html esta dentro de outra.

PADRONIZAÇÃO DA INTERNET

- O W3C é o órgão que regulariza e padroniza as regras de construção das páginas web (comportamento dos códigos nos navegadores).

- Existem diversos navegadores no mercado. Caso a versão atual do seu navegador não seja compatível com as normas atuais estabelecidas, o comportamento de seu código pode apresentar inconsistências, por isso, sempre use um navegador recente (cof cof..Internet Explorer cof cof..) e atualizado.

O QUE É O CSS?

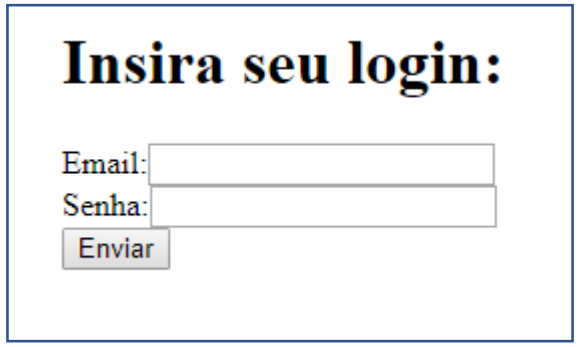
- O CSS é uma linguagem de estilização que opera na camada de apresentação do front-end. Ele serve para dar forma e estilos aos conteúdos criados em HTML- Por exemplo, cor da fonte, cor de um plano de fundo, tamanho de um elemento, etc.

O QUE É O CSS?

- O exemplo abaixo é um simples trecho HTML de um formulário de login e sua representação no navegador.

```
<div class="formulario">
    <h1 class="titulo">Insira seu login:</h1>
    <form method="post" action="#">
        <label for="email">Email:</label><input type="email" name="email" id="email" class="email"><br>
        <label for="senha">Senha:</label><input type="password" name="senha" id="senha" class="senha"><br>
        <input type="submit" name="enviar" class="enviar">
    </form>
</div>
```

- O resultado final será:



O QUE É O CSS?

- Aplicando o CSS pode se tornar algo bem diferente e interessante...

Tudo depende da sua imaginação.

COMO UTILIZAR O CSS?

- Podemos utilizar o CSS de três maneiras:
- Dentro da tag <style>
- Dentro do próprio elemento HTML
- Dentro de um arquivo externo

COMO UTILIZAR O CSS?

- A primeira maneira seria utilizando dentro da tag <head> onde podemos abrir uma tag chamada <style>, onde vamos inserir os códigos css.
- Esse método não é utilizado, devido ao arquivo HTML ficar relativamente pesado para ser carregado no navegador

```
<head>
    <meta charset="utf-8">
    <title>Exemplo html</title>
    <style type="text/css">
        /*insira seus codigos css aqui*/
    </style>
</head>

<body>
```

COMO UTILIZAR O CSS?

- A segunda maneira seria utilizar dentro das tags HTML um parâmetro chamado style, passando como valor os códigos CSS relativos.
- Porém, isso pode acabar deixando o código mais confuso, além de ter os mesmos problemas de executar os códigos dentro da tag <style>.

```
<div class="formulario">
    <h1 class="titulo">Insira seu login:</h1>
    <form method="post" action="#">
        <label for="email">Email:</label><input type="email" name="email" id="email" class="email" style="
            background: red;
            color:blue;
            text-align: center;
            font-family: arial;
        "><br>
        <label for="senha">Senha:</label><input type="password" name="senha" id="senha" class="senha"><br>
        <input type="submit" name="enviar" class="enviar">
    </form>
</div>
```

COMO UTILIZAR O CSS?

- O terceiro método é o mais aconselhável, que é através de um arquivo externo, que devemos importar dentro de uma tag chamada <link> que será declarada dentro da tag <head>. Desse modo o CSS será carregado somente quando for necessário, não deixando a aplicação pesada.

```
<head>
    <meta charset="utf-8">
    <title>Exemplo html</title>
    <link rel="stylesheet" type="text/css" href="styles.css">
</head>

<body>
```

- Rel="stylesheet" e type="text/css" são parâmetros que o navegador recebe para saber que se trata de um arquivo CSS e poder executar o código sem problemas caso encontre algum conflito.
- href="" é o caminho onde se encontra o arquivo com os códigos CSS, podendo ser um arquivo em uma pasta (parecido com src="" da tag img) ou um diretório da internet (CDN).
- O arquivo CSS possui a extensão .css

SINTAXE CSS

- A sintaxe CSS é formada por uma regra com 3 itens fundamentais para definir um estilo. São eles:

- Seletores;

- Propriedades;

- Valores.

SELETORES CSS

- O seletor vincula um elemento do documento HTML a declaração CSS. Pode ser usada de três maneiras .

- Referenciando as tags HTML

- Referenciando uma “class”

- Referenciando um “id”

- Dentro do local dos códigos CSS será citada a referência que se deseja estilizar seguido de uma abertura e fechamento de chaves, onde serão colocados os códigos CSS.

SELETORES CSS: TAG HTML

- Se utilizar como referência as tags HTML, todos os elementos que forem do mesmo tipo vão ser estilizados como você definir.

HTML

```
<body>
    <h1>Tipos de Estilização</h1>
    <div >
        <!-- aqui se encontra uma div que será igual a de baixo -->
    </div>
    <div>
        <!-- aqui se encontra uma div que será igual a de cima -->
    </div>
    <div>
        <!-- essa div será igual as duas de cima -->
    </div>

</body>
```

CSS

```
div{
    /*aqui dentro será colocado
    os códigos de estilização*/
}

h1{
    /*o h1 será diferente do que
    foi utilizado nas divs, pois
    pertence a um tipo diferente
    de elemento html*/
}
```

SELETORES CSS: CLASSES

- Utilizando como referência uma classe, somente os elementos que tiverem o mesmo nome da classe vão ser estilizados.
- Para declarar a classe você deve utilizar um parâmetro chamado *class*; passando como valor o nome da classe. Ao referenciar a classe no documento CSS você deve utilizar um ponto.

HTML

```
<body>
    <h1 class="titulo">Tipos de Estilização</h1>
    <div class="exemplo_1">
        <!-- aqui se encontra uma div
            com a classe exemplo_1, que será estilizada
            de acordo com a definição
            do CSS relativa a essa class-->
    </div>
    <div class="exemplo_2">
        <!-- aqui se encontra uma div
            com a classe exemplo_2, que será estilizada
            de acordo com a definição
            do CSS relativa a essa class
            essa div será diferente da div acima-->
    </div>
    <div class="exemplo_2">
        <!-- essa div terá o mesmo estilo definido na div acima -->
    </div>
```

CSS

```
.exemplo_1{
    /*aqui dentro será colocado
    os códigos de estilização
    referente a classe exemplo_1*/
}

.exemplo_2{
    /*aqui dentro será colocado
    os códigos de estilização
    referente a classe exemplo_2
    que serão diferentes da div acima*/
}
.titulo{
    /*o h1 terá o estilo referente
    ao que for definido aqui*/
}
```

SELETORES CSS: IDS

- Se utilizar como referência os id's, será semelhante as classes, porém no CSS será utilizado uma hashtag na frente do nome. Esse método é mais utilizado para uso no javascript. Evite usar esse método, mas caso utilize, não repita seu uso (quando você estudar javascript você entenderá).

HTML

```
<h1 id="titulo">Tipos de Estilização</h1>
<div id="exemplo_3">
    <!-- aqui se encontra uma div
        com o id exemplo_3, que será estilizada
        de acordo com a definição
        do CSS relativa a esse id-->
</div>
<div id="exemplo_4">
    <!-- aqui se encontra uma div
        com a classe exemplo_4, que será estilizada
        de acordo com a definição
        do CSS relativa a esse id
        essa div será diferente da div acima-->
</div>
<div id="exemplo_4">
    <!-- essa div terá o mesmo estilo definido na div acima
    porém esse é um metódo que é mais utilizado no javascript
    repetir seu uso pode causar alguns erros no código
    então tente sempre usar um id único quando utilizar -->
</div>
```

CSS

```
#exemplo_3{
    /*aqui dentro será colocado
    os códigos de estilização
    referente ao id exemplo_3*/
}

#exemplo_4{
    /*aqui dentro será colocado
    os códigos de estilização
    referente ao id exemplo_4
    que serão diferentes dos
     definidos acima*/
}
#titulo{
    /*o h1 terá o estilo referente
    ao que for definido aqui*/
}
```

SELETORES CSS

- Também pode ser utilizado como referência a citação de mais de um elemento;

```
.classe1, .classe2, .classe3{
    /*deve se separar por
    virgula
    o nome dos elementos
    todos eles terão a mesma
    estilização
    */
}

.classe1, #id1, div {
    /*os elementos podem ser
    de tipos diferentes*/
}
```

SELETORES CSS

- Também pode ser utilizado como referência um elemento que pertence a outro elemento

HTML

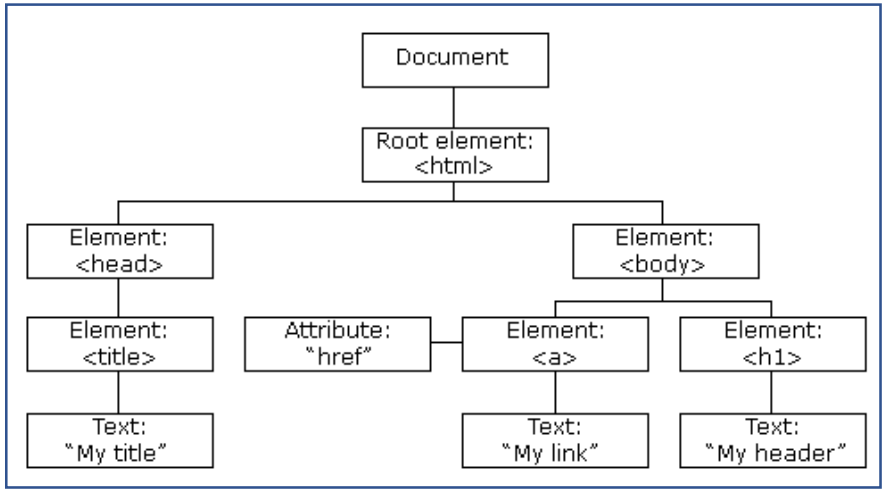
```html
<body class="pai">
    <h1 class="titulo">Seletor CSS</h1>
    <!-- aqui por exemplo temos dois elementos com classes
    chamadas paragrafo, mas que pertencem a elementos
    diferentes. É possivel estilizar apenas uma delas
    dizendo qual dos elementos que possui ela -->
    <div class="exemplo_heranca_1">
        <p class="paragrafo">
            paragrafo 1
        </p>
    </div>
    <div class="exemplo_heranca_2">
        <p class="paragrafo">
            paragrafo 2
        </p>
    </div>
</body>
```

CSS

```css
.pai h1{
    /*aqui por exemplo foi formatado o h1
    que pertence a um elemento com
    a classe pai. Qualquer outro h1
    dentro do código que não pertença
    a classe pai não será formatado*/
}

.pai .exemplo_heranca_1 .paragrafo{
    /*podem ser citado quantos elementos
    quiser*/
}

.exemplo_heranca_2 .paragrafo{
    /*sempre deve ser citado do
    elemento pai para o elemento filho.
    Aqui neste trecho, a classe
    paragrafo terá uma formatação
    diferente da proposta acima
    */
}
```

HERANÇA DE ESTILO

- Quando construímos um documento HTML nós podemos colocar elementos um dentro do outro ou um do lado do outro,
- Quando acontece um aninhamento de elementos chamamos eles de “elementos pai” e “elemento filho”.
- Quando um elemento esta do lado do outro chamamos ele de “elemento irmão”.
- Caso tenha curiosidade em saber mais, procure sobre “DOM” *(Document Object Model).*



HERANÇA DE ESTILO

- Quando acontece um aninhamento, a estilização feita no elemento pai será herdada pelos filhos.

HTML

```html
<body class="pai">
    <!-- body em todos os casos é o pai de todos os elementos de um documento
        tudo que você definir para ele será herdado por todos
        os elementos-->
    <h1 class="titulo">Herança</h1>
    <!-- Esse h1 é filha de body, mas irmão das divs
        perceba que o texto ficou vermelho
        embora a class titulo não tenha sido definida
        tudo que esta dentro de body ganhou a cor vermelha-->

    <div class="exemplo_heranca_1">
        <p class="paragrafo1">
            paragrafo 1

        </p>
    </div>
    <div class="exemplo_heranca_2">
        <p class="paragrafo2">
            paragrafo 2
        </p>
    </div>
</body>
```

CSS

```css
.pai{
    color:red;
    /*aqui indicamos que
    a cor do texto será
    vermelho, o código
    será explicado mais
    adiante*/
}
```

resultado

HERANÇA DE ESTILO

- Quando estilizarmos um dos filhos de <body>, ele deixará de herdar suas características, e os netos de <body> que são filhos do elemento em questão também deixarão de herdar dele, passando a herdar as características de seu pai. E assim sucessivamente.

HTML

```
<body class="pai">
    <h1 class="titulo">Herança</h1>
    <!--perceba agora que o texto esta em azul
        pois agora foi modificado comportamento do titulo
        perceba que esse comportamento não foi transferido
        para os irmãos, pois eles herdam somente do pai-->

    <div class="exemplo_heranca_1">
        <!-- foi definido que tudo que
        esta aqui dentro da div terá a cor
        verde, logo p que é filho de div também
        será verde -->
        <p class="paragrafo1">
            paragrafo 1

        </p>
    </div>
    <div class="exemplo_heranca_2">
        <!-- aqui continua herdando de body -->
        <p class="paragrafo2">
            paragrafo 2
        </p>
    </div>
</body>
```

CSS

```
.pai{
    color:red;
    background-color: orange;
}
.titulo{
    color:blue;
}
.exemplo_heranca_1{
    color:green;
}
```

resultado

HERANÇA DE ESTILO

- Agora você compreende o porque do nome *cascading style sheets* (folhas de estilo em cascata).

PROPRIEDADES CSS

- A propriedade CSS define uma característica visual para o elemento HTML "selecionado" pelo seletor.
- Existem propriedades que são focadas em texto, enquanto outras são focadas em posicionamento, entre outras.
- Se você obtem um conhecimento de inglês básico, verá que os nomes das propriedades são autoexplicativos.
- Toda propriedade CSS será seguida de ":" (dois pontos), onde logo após será dado o valor da propriedade, que pode ser um valor simples ou múltiplo, dependendo da propriedade.
- Após o valor deve ser colocado um ";" (ponto e virgula) para identificar que uma próxima propriedade pode ser colocada no código.

PROPRIEDADES CSS

- E agora a parte boa.....

PROPRIEDADES CSS

- A seguir um exemplo de definição das propriedades para um determinado seletor.

```css
.classeExemplo{
    font-size: 25px; /*exemplo de valor simples*/
    padding: 10px 10px; /*exemplo de valor multiplo*/
    text-shadow: 2px 2px 3px black;
    /*font size por exemplo não aceita valor multiplo
    enquanto padding aceita tanto valor simples como multiplo
    text-shadow aceita de 1 a 4 valores
    cada propriedade é um caso particular
    que deve ser estudado
    */
}
```

PROPRIEDADES CSS: TEXTO

- A seguir alguns exemplos de propriedades para definir textos.

- Cor de texto

```css
.corTexto{
    color: green; /*define a cor do texto. Pode ser definido
    com o nome de uma cor em inglês, usando um codigo rgb
    ou um codigo hexadecimal*/
    color: rgb(0,255,0); /*codigo rgb para verde*/
    color: #00FF00; /*codigo hexadecimal para verde*/
}
```

- Caso queira saber os valores rgb ou hexadecimal de uma cor, você pode utilizar o photoshop ou o site https://color.adobe.com/pt/create/color-wheel/

PROPRIEDADES CSS: TEXTO

- Fonte do texto

```
.fonteTexto{
    font-family: tahoma,arial,times, sans-serif;
/*  aqui você define a fonte do texto
    você pode definir varias.
    caso o usuário não possua a primeira fonte citada
    instalada no computador, o texto será formatado
    com a fonte a seguir, e assim sussessivamente.

    você também pode definir um valor genérico que define
    a caracteristica de uma fonte, no exemplo foi
    usado sans-serif
    serif = usar a primeira fonte serifada encontrada
    no computador do usuario
    sans-serif = usar a primeira fonte não-serifada encontrada
    no computador do usuario
    Existem outros valores opcionais de fontes genericas além do sans-serif e
    serif que definem características de fontes, mas não serão abordados aqui.
    */
}
```

PROPRIEDADES CSS: TEXTO

- Mas o que significa ser serifado e não-serifado?

- Serifa são os detalhes mais trabalhados que algumas fontes possuem

- Exemplo de fontes sem serifa: Verdana, Tahoma, arial, Bahnschrift SemiBold SemiConden(fonte desse slide).
- Exemplo de fontes com serifa: Times new roman, Ionic, Platin.

PROPRIEDADES CSS: TEXTO

- Você pode importar fontes externas através do @font-face

```
@font-face{
    font-family: exemplo;/* aqui você pode definir o nome da sua nova fonte*/
    src: url('exemplo.otf'); /*aqui você define o endereço da sua fonte
    pode ser um link da internet ou o caminho de um arquivo*/
}
/*a seguir você pode usar a sua nova fonte normalmente citando o nome que você deu*/
.fonteTexto{
    font-family: exemplo;
}
```

- Caso deseje procurar por fontes você pode usar o site https://fonts.google.com/ para baixar os arquivos ou instalar no seu computador

- Consulte em https://www.webcis.com.br/utilizando-font-face-tipografia-web.html para mais detalhes

PROPRIEDADES CSS: TEXTO

- Você pode definir o tamanho de uma fonte

```css
.tamanhoFonte{
    font-size: 25px;
    /*geralmente definido em pixel (px) ou unidade (pt) */
}
```

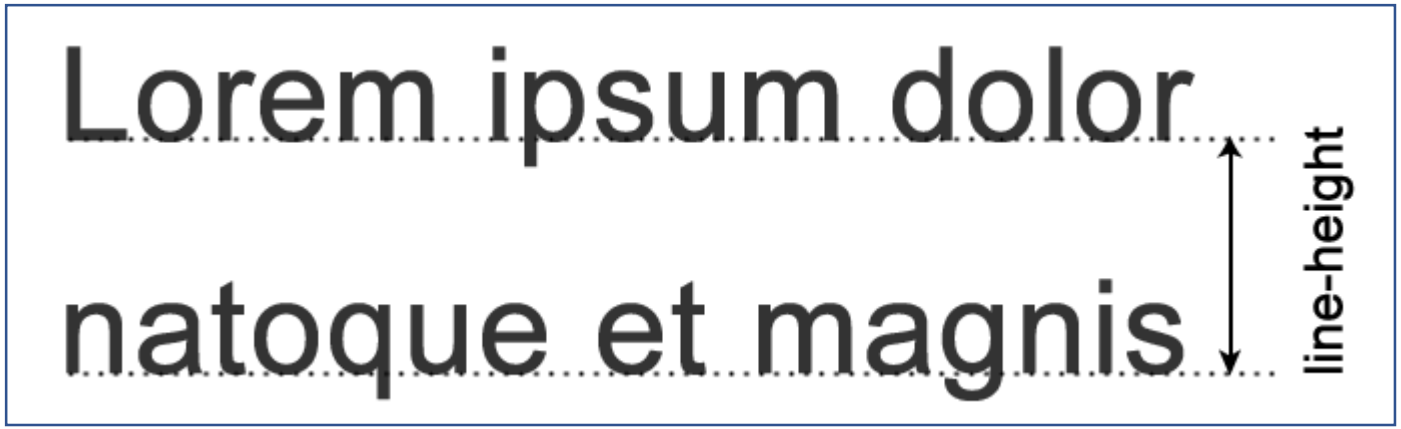
- Também é possível utilizar alguns valores pré-definidos do CSS.
  - xx-small (equivale a 9px)
  - x-small (equivale a 10px)
  - Small ou smaller (equivale a 13px)
  - medium (equivale a 16px)
  - large ou larger(equivale a 18px)
  - x-large (equivale a 24px)(utilizado nesse slide)
  - xx-large (equivale a 32px)

PROPRIEDADES CSS: TEXTO

- Você pode controlar o espaçamento entre linhas

```css
.espacoEntrelinha{
    line-height:10px;
    /*as mesmas unidades usadas no font-size*/
}
```

- Para entender melhor, o line height controla a distância entre a parte inferior do texto acima e a parte inferior do texto abaixo.



PROPRIEDADES CSS: TEXTO

- Você pode definir o alinhamento de um texto (a propriedade a seguir também serve para alguns conteúdos além de texto).

```css
.alinhamentoTexto{
    text-align: left;
    /*valores possiveis: center,left,right,justify;*/
}
```

center

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur

left

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur

right

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur

justify

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur

PROPRIEDADES CSS: TEXTO

- Você pode definir o peso de uma fonte (negrito)

```css
.pesoFonte{
    font-weight: normal;
    /*valores possiveis: bolder e normal;*/
}
```

normal

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur

bold

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur

PROPRIEDADES CSS: TEXTO

- Você pode definir o estilo de uma fonte (itálico ou obliquo)

```css
.estiloFonte{
    font-style: normal;
    /*valores possiveis: normal, italic e oblique;*/
}
```

normal	italic	oblique
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur	*Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur*	*Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur*

Para entender a diferença entre italic e oblique acesse https://caderninhodeideias.wordpress.com/2018/04/23/a-diferenca-entre-texto-italico-e-obliquo/

PROPRIEDADES CSS: TEXTO

- Você pode definir o espaçamento que a primeira linha do texto terá em relação a margem.

```css
.espacamentoMargemFonte{
    text-indent: 30px;
    /*aceita dimensões em px e %;*/
}
```

normal

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur

text-indent de 30px

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur

PROPRIEDADES CSS: TEXTO - ANTES DE TERMINAR...

- Existe uma tag do HTML chamada que serve somente para atribuir alguma classe a um pedaço de texto ou outro código qualquer (em sua maioria é usado para formatar pedaços de texto).

- Exemplo:

```
<body>
    <p>
        Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam finibus magna justo, sed maximus augue
        finibus sed. Pellentesque euismod mattis faucibus. Cras ac pulvinar erat, ac convallis sem. Proin non
        massa euismod, tincidunt odio vel, blandit eros. Quisque eget lacus fermentum, finibus erat vel,
    </p>
    <p>
        Etiam eleifend, neque in ultrices semper, sapien orci pellentesque sapien, vitae tincidunt dui est et
        velit. Sed porttitor erat et justo euismod viverra. <span class="estilo">Curabitur ac ante et arcu
        ultricies porta ut quis lectus. Quisque vitae posuere risus, ac accumsan ipsum.</span> Duis rhoncus
        lacus at consequat gravida. Nullam sodales dui vel metus venenatis, eget elementum lacus porta. Sed ac
        lectus sed ex tristique lacinia.

</body>
```

Isso irá gerar dois parágrafos bem simples

PROPRIEDADES CSS: TEXTO - ANTES DE TERMINAR...

- O resultado será:

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam finibus magna justo, sed maximus augue finibus sed. Pellentesque euismod mattis faucibus. Cras ac pulvinar erat, ac convallis sem. Proin non massa euismod, tincidunt odio vel, blandit eros. Quisque eget lacus fermentum, finibus erat vel,

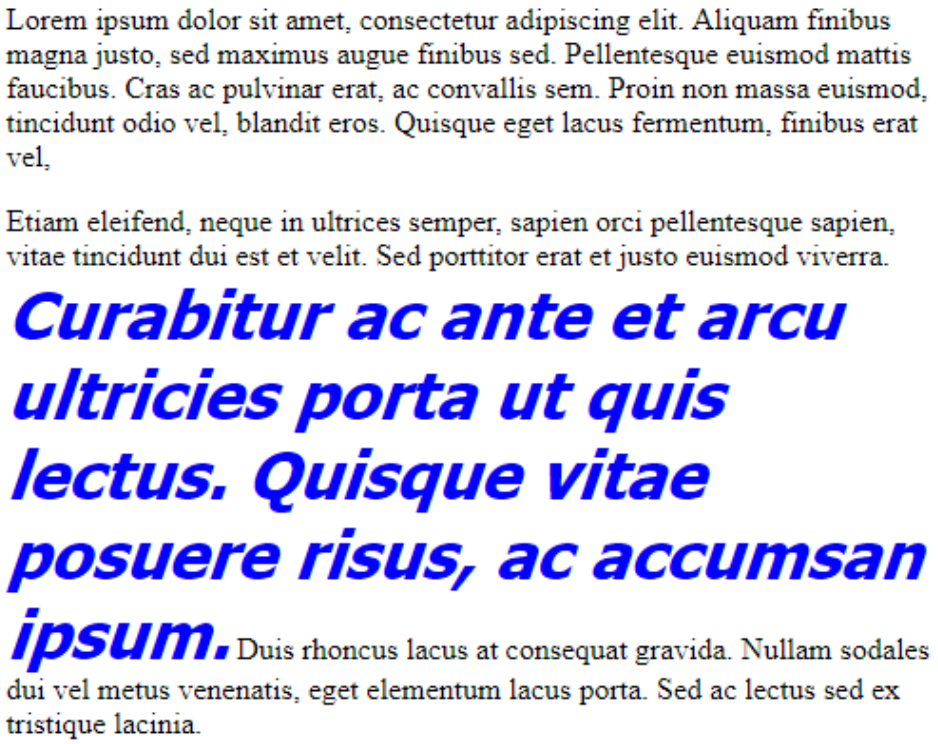
Etiam eleifend, neque in ultrices semper, sapien orci pellentesque sapien, vitae tincidunt dui est et velit. Sed porttitor erat et justo euismod viverra. Curabitur ac ante et arcu ultricies porta ut quis lectus. Quisque vitae posuere risus, ac accumsan ipsum. Duis rhoncus lacus at consequat gravida. Nullam sodales dui vel metus venenatis, eget elementum lacus porta. Sed ac lectus sed ex tristique lacinia.

- A seguir a classe CSS será definida

```css
.estilo{
    color:blue;
    font-family: tahoma,serif;
    text-indent: 46px;
    font-size:35px;
    font-weight: bold;
    font-style: italic;
}
```

PROPRIEDADES CSS: TEXTO - ANTES DE TERMINAR...

- Ao salvar, o resultado será:



- Somente a parte com a tag span com a classe estilo foi estilizada.

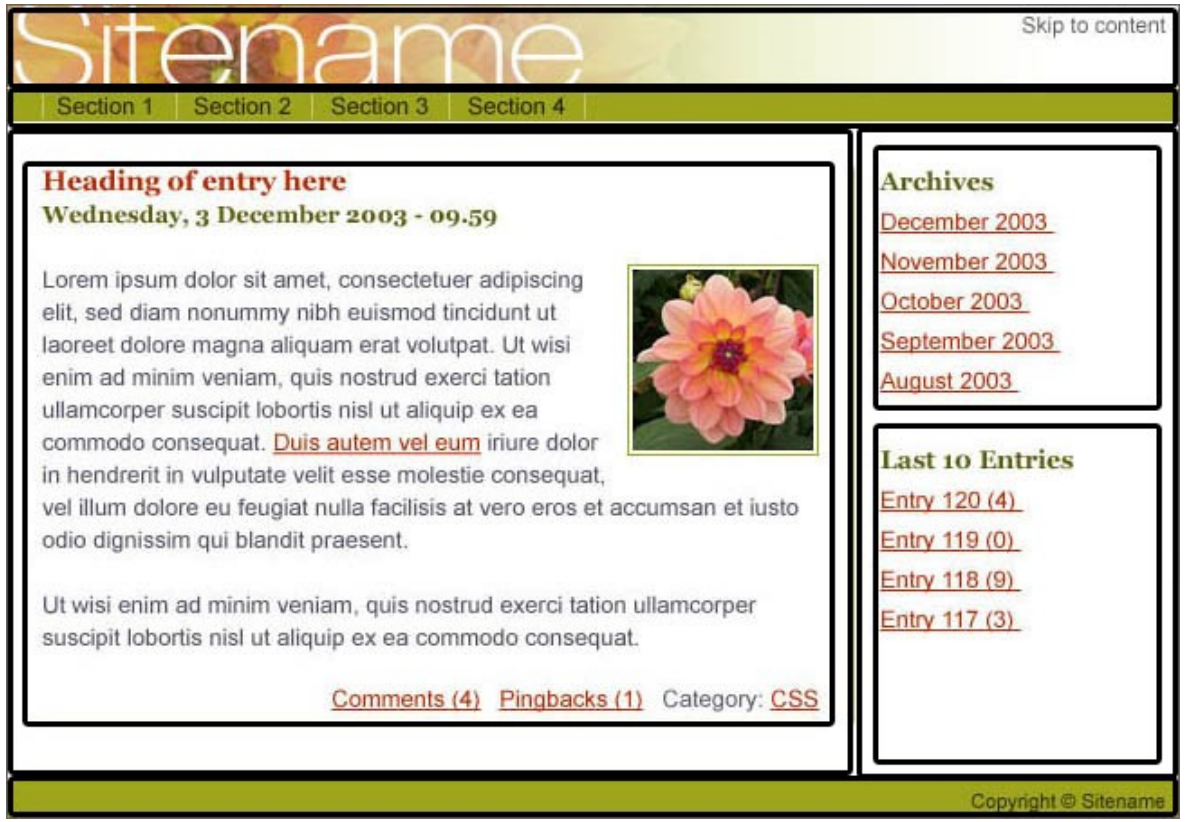
PROPRIEDADES CSS: TEXTO - ANTES DE TERMINAR...

- Bora brincar....

- Abra a pasta exercícios e siga as instruções do arquivo exercicio1.html.
- Existe um arquivo CSS dentro da pasta que você pode usar para estilizar. Ele está linkado a todos os exercícios, então lembre-se de usar classes diferentes em cada exercício, caso contrário, as classes podem acabar formatando os exercícios do arquivo errado sem querer.

PROPRIEDADES CSS: BLOCOS

- A pagina web é dividida em vários partes que definem os limites de cada elemento da página. Para fins didáticos chamaremos de blocos.
- A seguir um exemplo de uma página web com suas divisões representadas.



PROPRIEDADES CSS: BLOCOS

- Agora temos duas div's com nenhuma estilização e seu resultado no HTML.

HTML

```
<body>
    <div class="font diva">
        <!-- é possivel adicionar mais
        de uma classe, separando elas
        com um espaço -->
        <p>texto 1</p>
    </div>
    <div class="font divb">
        <p>texto 2</p>
    </div>
</body>
```

resultado

texto 1

texto 2

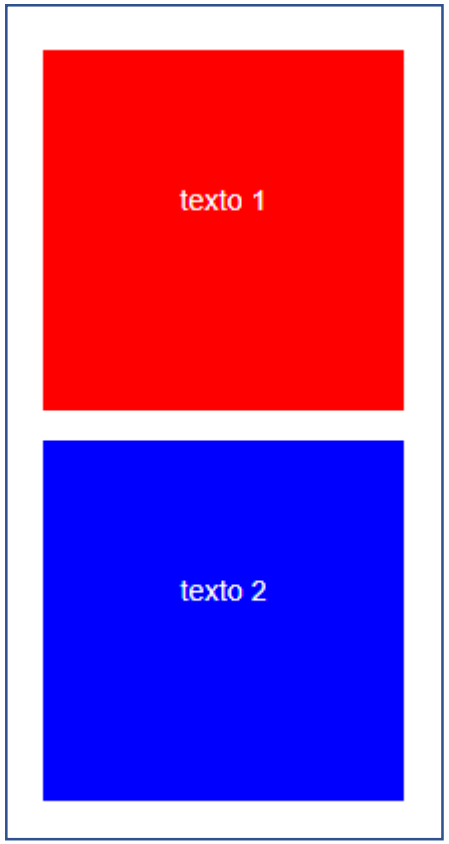
- A seguir vamos definir algumas estilizações para as div's.

PROPRIEDADES CSS: BLOCOS

CSS

```css
.font{
    line-height: 168px;
    text-align: center;
    color:white;
    font-family: sans-serif;
} /*formatação dos textos dentro
das div's*/

.diva{
    width:200px; /*define a largura da div*/
    height:200px; /*define a altura da div*/
    background-color: red;
/*  define a cor de fundo da div
    seguem a mesma regra de definição
    de cor para textos*/
}
.divb{
    width:200px;
    height:200px;
    /*também pode ser definido por porcentagem */
    background-color: blue;
}
```

resultado



PROPRIEDADES CSS: BLOCOS

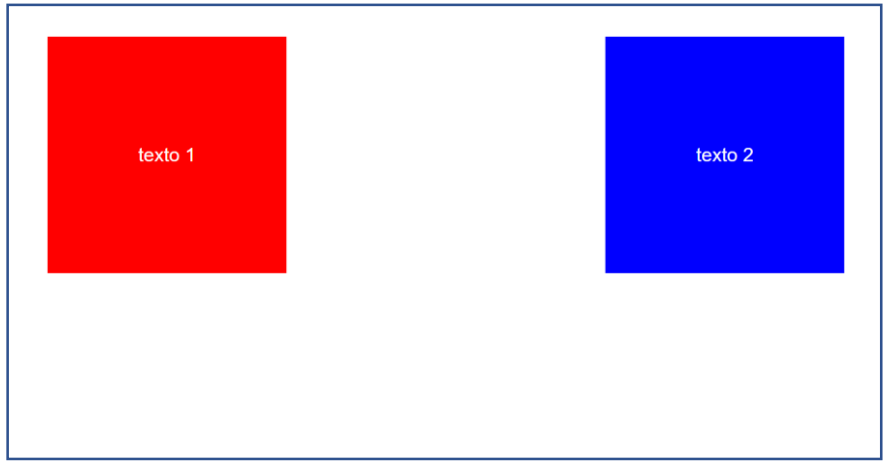
- Como colocar as div's uma do lado da outra?
- Para isso temos que entender como funciona as propriedades float e clear.
- Em resumo, a propriedade float permite que tiremos um elemento do fluxo vertical da página e automaticamente faz com que o conteúdo que venha a seguir flutue ao seu redor.
- O clear tem a função de limpar a flutuação em determinado ponto, fazendo os elementos descerem.

PROPRIEDADES CSS: BLOCOS

CSS

```css
.font{
    line-height: 168px;
    text-align: center;
    color:white;
    font-family: sans-serif;
} /*formatação dos textos dentro
das div's*/

.diva{
    width:200px;
    height:200px;
    background-color: red;
    float:left;
}
.divb{
    width:200px;
    height:200px;
    background-color: blue;
    float:right;
}
```

resultado



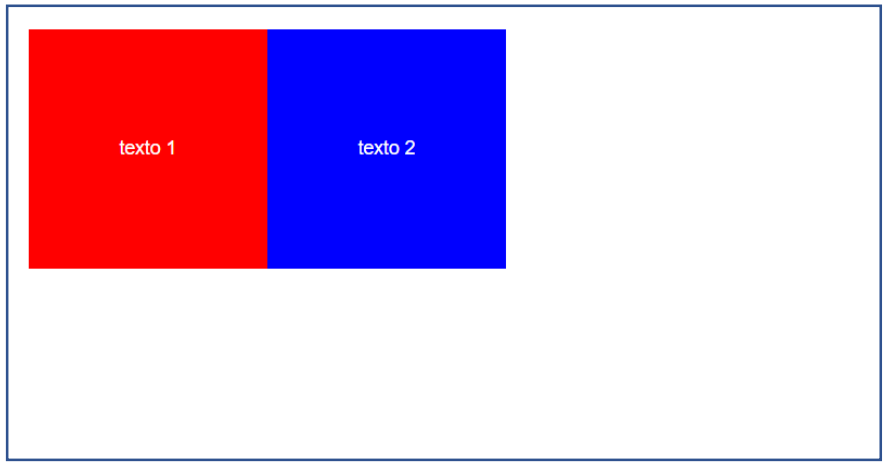
- Float indicará o sentido que o elemento irá flutuar em relação a página

PROPRIEDADES CSS: BLOCOS

CSS

```css
.diva{
    width:200px;
    height:200px;
    background-color: red;
    float:left;
}
.divb{
    width:200px;
    height:200px;
    background-color: blue;
    float:left;
}
```

resultado



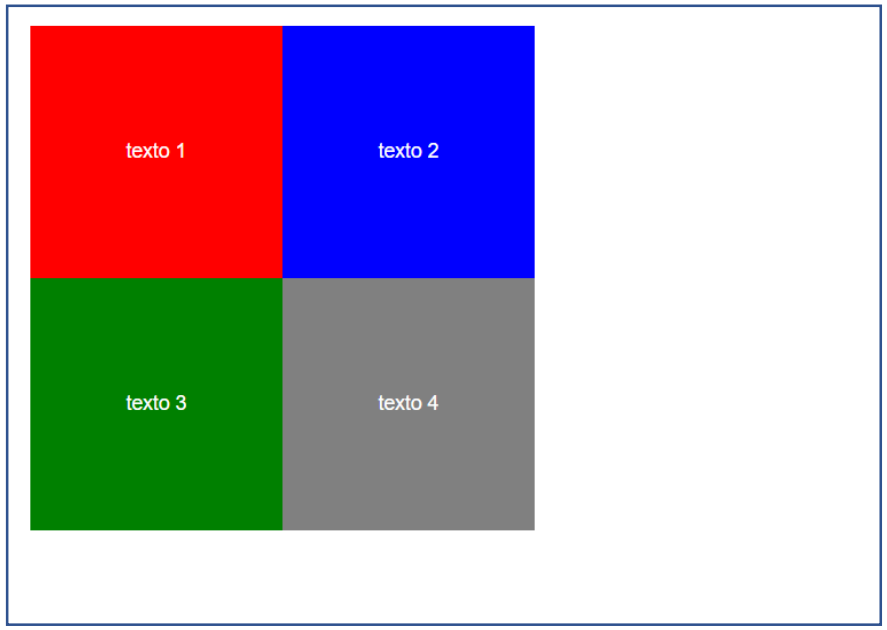
- Para que as div's fiquem juntas, elas devem ter o mesmo float

PROPRIEDADES CSS: BLOCOS

CSS

```css
/*a seguir foram colocadas mais algumas
divs(as anteriores seguem iguais)*/
.divc{
    width:200px;
    height:200px;
    background-color: green;
    float:left;
    clear:left;
    /*clear serve para quebrar a flutuação
    do elemento, fazendo com que ele desça
    aceita os valores left, right e both*/
}
.divd{
    width:200px;
    height:200px;
    background-color: gray;
    float:left;
}
```

resultado



- Aplicando o clear na divc, as div's deixaram de ser enfileiradas todas na mesma "linha", fazendo ela e os elementos a seguir descerem.

PROPRIEDADES CSS: BLOCOS

- Bora brincar de novo....
- Entre na pasta exercicios e abra o arquivo exercicio2.html.
- Tente fazer o seguinte layout, sem se preocupar com as dimensões e cores, porém a largura deve ocupar 100% da pagina.

Acesse o link abaixo para

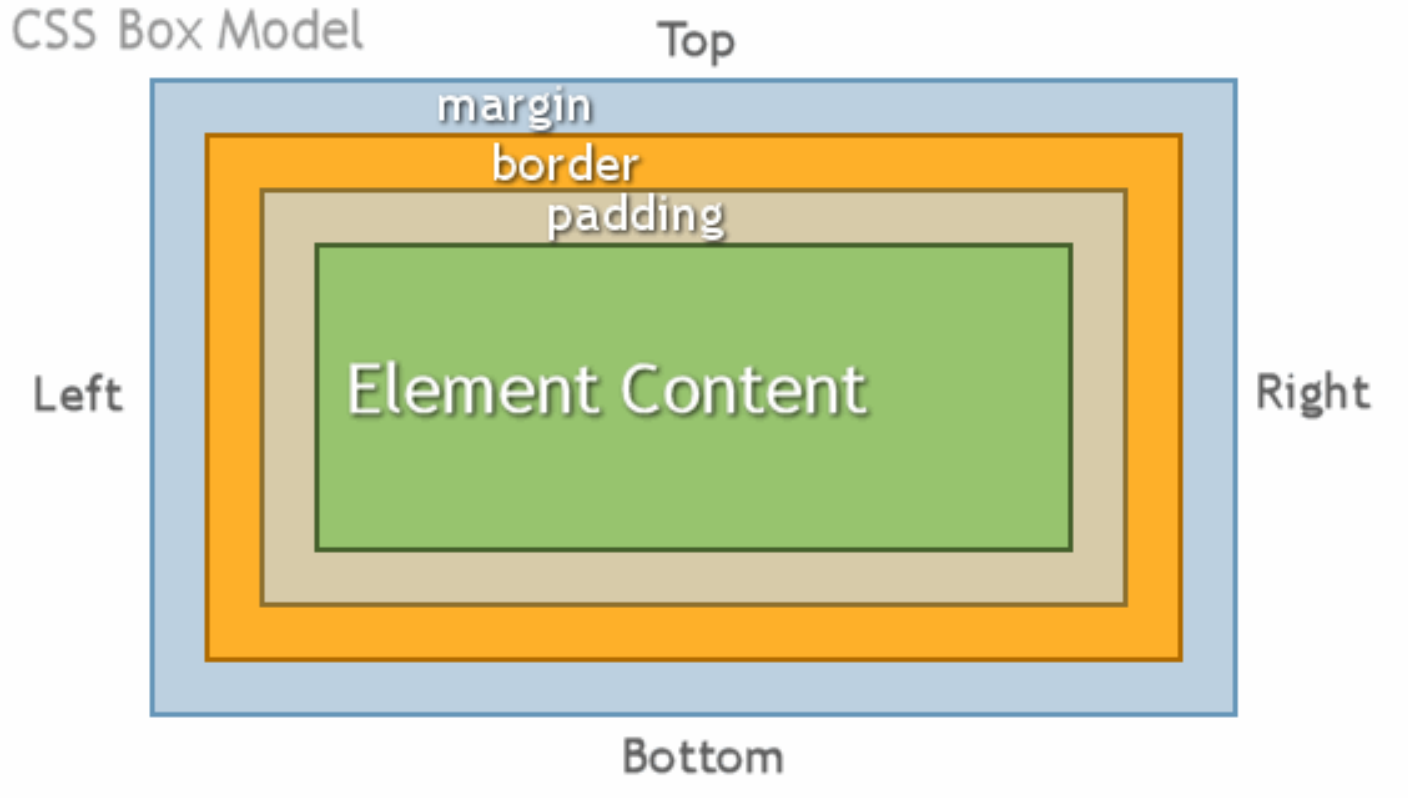
Entender sobre HTML semântico

Caso tenha curiosidade.

https://www.devmedia.com.br/html-semantico-conheca-os-elementos-semanticos-da-html5/38065

PROPRIEDADES CSS: BOX MODEL

- Cada elemento é representado como um *box* retangular, representado abaixo.



PROPRIEDADES CSS: BOX MODEL

- Padding é o espaçamento entre o conteúdo e a borda do elemento. Pode ser definido por :

```
.padding{
    padding:10px; /*espaçamento de 10px nas quatro direções*/
    padding: 20px 10px; /*espaçamento de 20px no eixo y (cima e embaixo)
    e 10px no eixo x (direita e esquerda) */
    padding: 10px 20px 10px;/* defininição em cima, dos lados e embaixo*/
    padding:20px 10px 20px 10px; /*primeiro valor define o espaçamento de cima
    o restante segue o sentido do relogio*/
    padding-top: 10px; /*auto explicativos*/
    padding-right: 10px;
    padding-bottom: 10px;
    padding-left: 10px;
    /*aceita unidades em porcentagem e pixels*/
}
```

PROPRIEDADES CSS: BOX MODEL

- Border nada mais é do que a borda que limita um elemento. Pode ser definido por:

```
.border{
    border-width:1px;/* define a espessura da borda*/
    border-style:solid; /*define o estilo da borda
    pode ser definida por none, hidden, dotted, dashed,
    solid, double, groove, ridge, inset e outset*/
    border-color:black; /*define a cor da borda
    segue a mesma regra para definição de cor de textos*/
    border-radius:10px; /*define o arredondamento dos
    cantos da borda,*/
    border-top-width:2px; /*também pode ser usado
    para somente uma das bordas, semelhante ao padding*/
    border:1px solid black; /*maneira abreviada de definir
    espessura, estilo e cor*/
}
```

PROPRIEDADES CSS: BOX MODEL

- Margin é o espaço que existe entre a borda de um elemento e a borda de outros elementos, pode ser definido por:

```
.margin{
    margin:1px; /*espaçamento de 1px nas 4 direções*/
    margin: 20px 10px; /*espaçamento de 20px no eixo y (cima e embaixo)
    e 10px no eixo x (direita e esquerda) */
    margin: 10px 20px 10px;/* defininição em cima, dos lados e embaixo*/
    margin: :20px 10px 20px 10px; /*primeiro valor define o espaçamento de cima
    o restante segue o sentido do relogio*/
    margin-top: 10px; /*auto explicativos*/
    margin-right: 10px;
    margin-bottom: 10px;
    margin-left: 10px;
    /*aceita unidades em porcentagem e pixels*/
    /*em resumo, margin funciona com as mesmas declarações de padding*/
}
```

PROPRIEDADES CSS: BOX MODEL: BOX-SIZING

- Utilizada para alterar a propriedade padrão da box model, usada para calcular larguras (widths) e alturas (heights) dos elementos.

- Quando definimos um padding ou uma borda ao elemento, por padrão esses valores se somam ao valor da altura e da largura, ou seja, o elemento fica maior do que realmente foi especificado.

- No exemplo a seguir perceba que as divs foram definidas com a mesma largura e altura, porem uma esta claramente maior que a outra devido a ela possuir borda e padding.

PROPRIEDADES CSS: BOX MODEL: BOX-SIZING

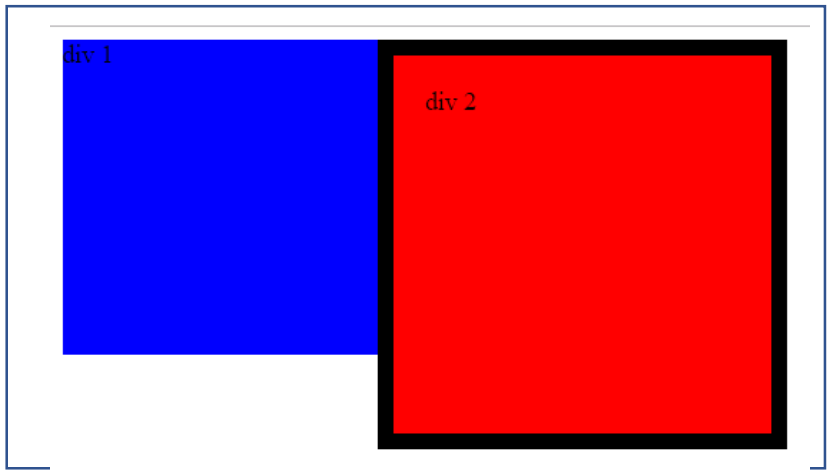
HTML

```
<div class="real">

</di>
<div class="somada">

</div>
```

CSS

```
.real{
    width:200px;
    height: 200px;
    background-color: blue;
    float: left;
}
.somada{
    width:200px;
    height: 200px;
    background-color: red;
    border: 10px solid black;
    padding:20px;
    float: left;
}
```

resultado



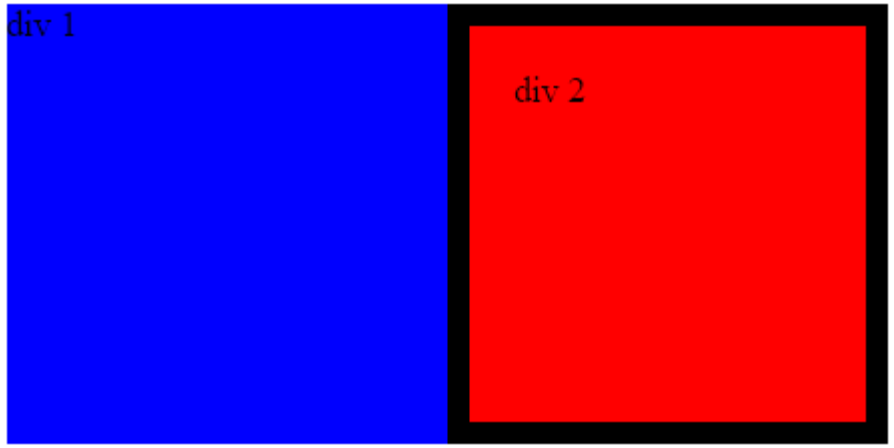
PROPRIEDADES CSS: BOX MODEL: BOX-SIZING

- Para evitar esse problema utilizamos a propriedade box-sizing com o valor border-box, fazendo com que o elemento fique realmente com o valor que foi definido sem perder a borda e o padding definido.

CSS

```css
.real{
    width:200px;
    height: 200px;
    background-color: blue;
    float: left;
}
.somada{
    width:200px;
    height: 200px;
    background-color: red;
    border: 10px solid black;
    padding:20px;
    float: left;
    box-sizing: border-box;
}
```

resultado



PROPRIEDADES CSS: UMA DICA...

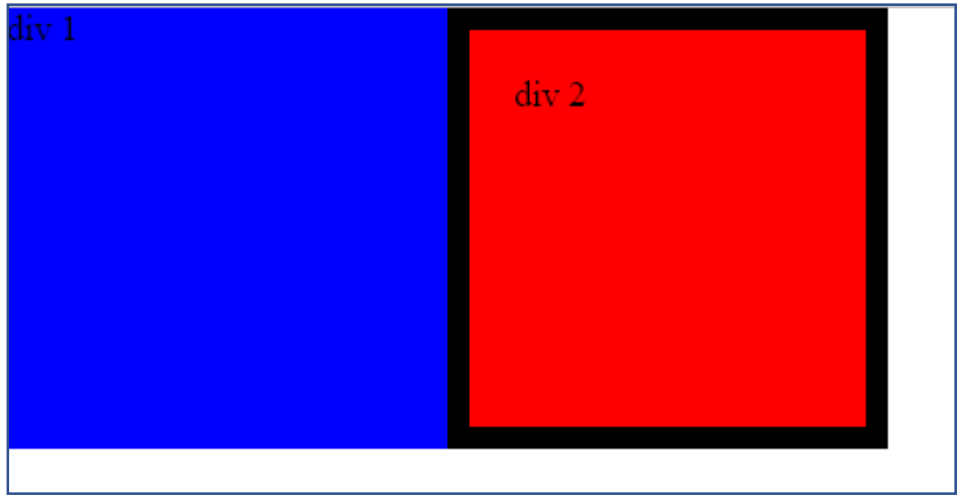
- Perceba que seus elementos possuem uma margem em relação ao navegador, isso acontece devido ao valor de margin padrão do body não ser nulo (valor 0).

- Para evitar esse problema, além do box-sizing e de alguns elemento que possuem padding padrão, é aconselhavel "resetar" o estilo, ou seja, comece seu arquivo CSS sempre com o código abaixo:

CSS

```css
*{/*estrela significa que isso será
    aplicado a todos os elementos da página*/
    box-sizing: border-box;
    margin:0;
    padding: 0;
}
```

Resultado do exemplo anterior



PROPRIEDADES CSS: IMAGENS

- É possível atribuir imagens de fundo aos elementos

HTML

```html
<div class="imagem">
    <div class="foto"></div>
</div>
```

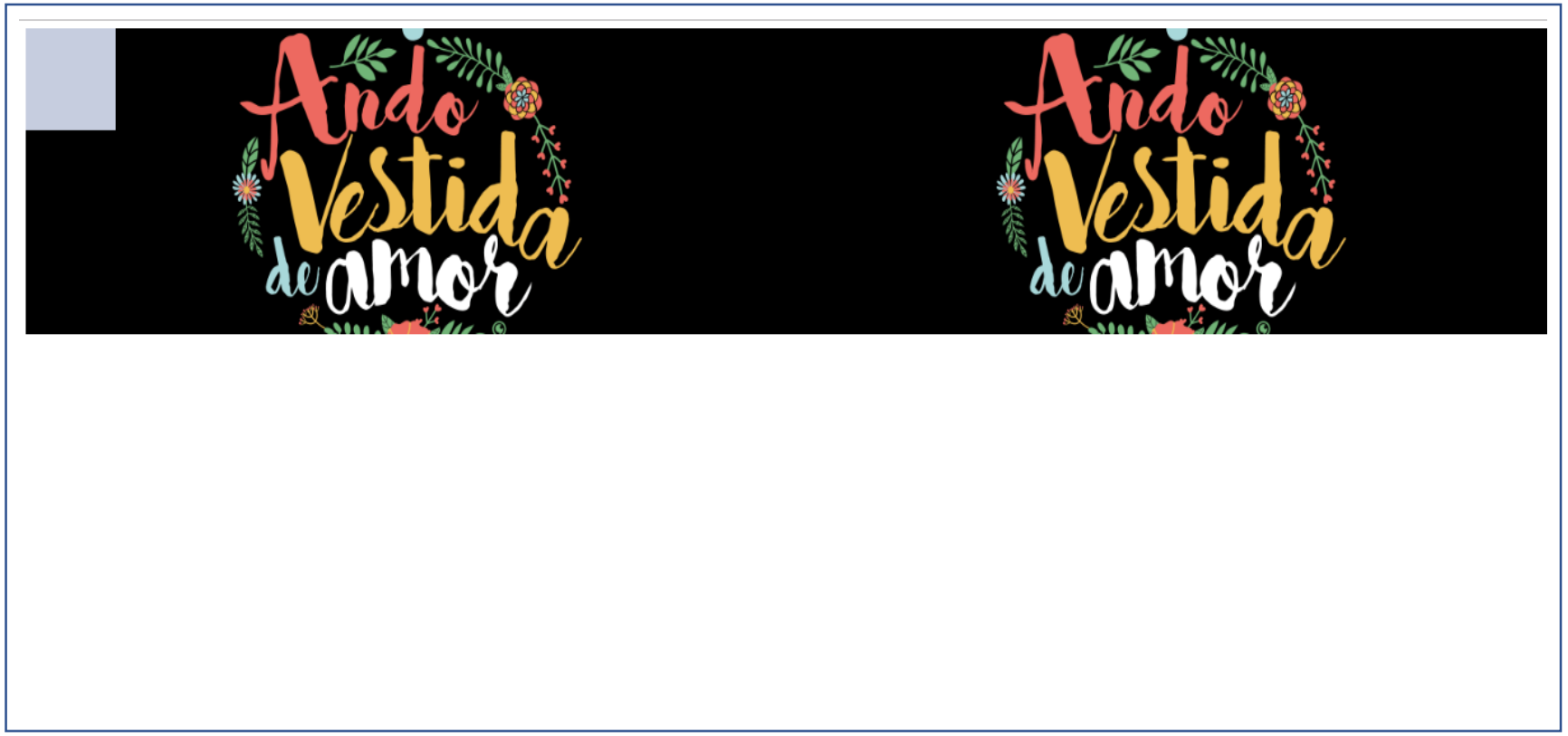
CSS

```css
.imagem{
    width:100%;
    height:300px;
    background-image: url("novela.png");
    /*para as fotos aparecerem, dimensoes
    de largura e altura devem ser especificadas*/
}
.foto{
    width:100px;
    height:100px;
    background-image: url("fotoperfil.jpg");
}
```

- Essas são as fotos:

PROPRIEDADES CSS: IMAGENS

- Teremos como resultado:



PROPRIEDADES CSS: IMAGENS

- A primeira imagem possui um problema, ela esta se repetindo. Para resolver isso podemos usar a propriedade abaixo:

```css
.imagem{
    width:100%;
    height:300px;
    background-color: red; /*para ver o resto do background*/
    background-image: url("novela.png");
    background-repeat: no-repeat;
    /*aceita os valores
    repeat|repeat-x|repeat-y|no-repeat|initial|inherit;*/
}
```

- O resultado será:

PROPRIEDADES CSS: IMAGENS

- O próximo problema a ser resolvido é tentar fazer com que a imagem cubra todo o campo da div. Podemos fazer isso através da propriedade abaixo:

```css
.imagem{
    width:100%;
    height:300px;
    background-color: red; /*para ver o resto do background*/
    background-image: url("novela.png");
    background-repeat: no-repeat;
    background-size:100% 100%;
    /* isso define que na direção x e na direção y será usada
    toda a extensão da imagem em toda a div*/
    /*aceita outros valores: contain e cover */
}
```

- O resultado será:

PROPRIEDADES CSS: IMAGENS

- A segunda imagem possui uma dimensão muito maior do que o espaço destinado a ela, por isso ela tem apenas um espacinho definido, isso pode ser ajustado com uma propriedade já vista:

```css
.foto{
    width:100px;
    height:100px;
    background-image: url("fotoperfil.jpg");
    background-size:100% 100%;
}
```

- O resultado será:

PROPRIEDADES CSS: POSISÕES

- Podemos definir a posição de um elemento em relação a outro através da propriedade position:

```css
.imagem{
    width:100%;
    height:300px;
    background-image: url("novela.png");
    background-repeat: no-repeat;
    background-size:100% 100%;
    position: relative;
    /*relative significa que ele será o elemento de referencia*/
}

.foto{
    width:100px;
    height:100px;
    background-image: url("fotoperfil.jpg");
    background-size:100% 100%;
    position: absolute;
    /*absolute significa que ele será absoluto em relação a referencia*/
    bottom:10px;
    left:10px;
/*  a seguir posso dizer qual a posição que o absoluto terá
    em relação ao relative
    uma página pode ter diversos position: relative;
    qual que meu elemento irá usar de referencia?
    simples, o mais próximo dele na hierarquia*/
}
```

PROPRIEDADES CSS: POSISÕES

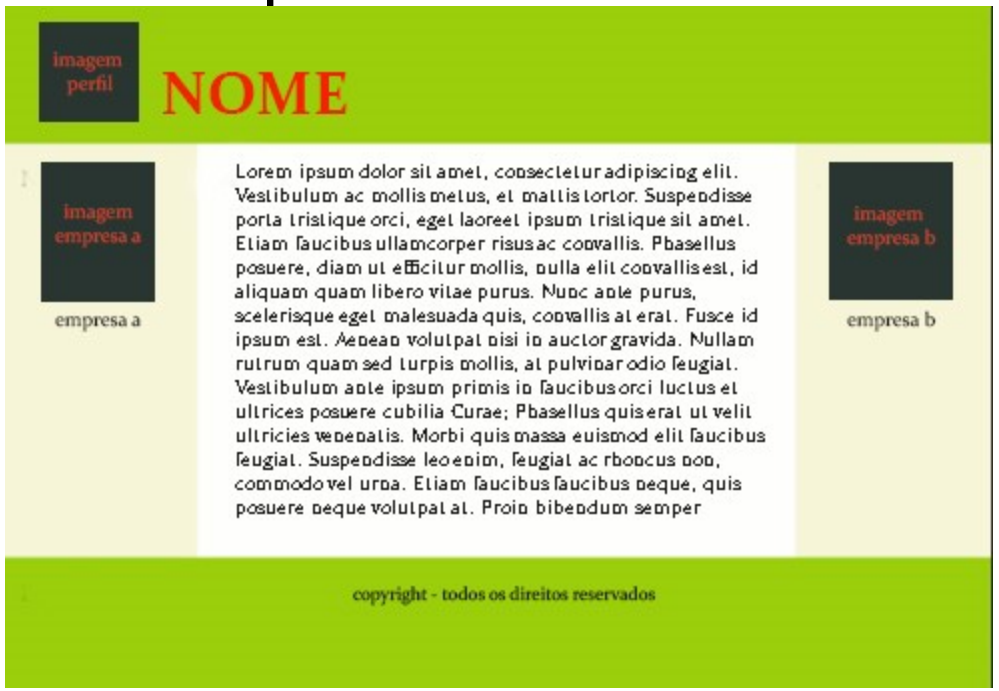
- Temos como resultado:

PROPRIEDADES CSS: POSIÇÕES

- Bora brincar de novo....
- Utilize o mesmo arquivo do exercício anterior (de preferência faça uma nova cópia da pasta e dê o nome de exercicio3.html).
- Tente inserir imagens e textos de acordo com o que é mostrado.

Gerador de lorem ipsun (texto aleatório para testes)

https://www.lipsum.com/



PROPRIEDADES CSS: DISPLAY

- Especifica o tipo de caixa de renderização usada por um elemento (como um elemento será exibido).
- É considerada por muitos a propriedade CSS mais importante de todas.

```
.ExemploDisplay{
    display: block;
    /*os valores mais usados: block, none, inline, inline-block, flex, inherit.*/
}
```

- A seguir será mostrado alguns comportamentos.

PROPRIEDADES CSS: DISPLAY

- block : significa que o elemento esta em nível de bloco.

```
<div>
    div é o exemplo de bloco mais comum. O elemento de bloco sempre começa em uma nova linha e se expande pra esquerda e direita o tanto quanto for possível. Outros elementos de bloco comuns são p e form, e agora no HTML5 temos: header, footer, section e outros.
</div>
```

- inline : significa que o elemento esta em nível de linha.

span é o elemento de linha padrão. Um elemento de linha pode preencher algum texto dentro de um parágrafo como esse sem quebrar o fluxo daquele parágrafo. O elemento a é o elemento de linha mais comum, já que ele é utilizado para links.

PROPRIEDADES CSS: DISPLAY

- none : significa que o elemento não esta visível e nem sequer existindo. Esse comportamento geralmente é utilizado junto de javascript para esconder elementos como por exemplo um modal ou um menu.

- Inline-block : significa que o elemento esta em nível de bloco, porém sem usar toda a largura do navegador, apenas o espaço que ele realmente ocupa .

- Perceba que os 3 elementos acima estão com display: inline-block, isso fez com que o elementos não fossem postos um abaixo do outro, algo que precisaria de um float para conseguir.

MENUS

- Antes de tudo, parabéns por ter chegado até aqui...

CONSTRUÇÃO DE UM MENU HORIZONTAL

- Vamos fazer agora a estrutura de um menu horizontal, iniciamos pelo HTML simples

HTML

```html
<nav> <!-- nav é uma div semanticamente dedicada a menus -->
    <ul class="menu">
        <li><a href="#">home</a></li>
        <li><a href="#">sobre</a></li>
        <li><a href="#">o que fazemos</a></li>
        <li><a href="#">contato</a></li>
    </ul>
</nav>
```

Resultado

- home
- sobre
- o que fazemos
- contato

- Em seguidas vamos fazer as configurações iniciais do CSS

CSS

```css
*{
    box-sizing: border-box;
    margin:0;
    padding: 0;
}
body{
    font-family: arial, helvetica, sans-serif;
    font-size: 12px;
}
```

Resultado

home
sobre
o que fazemos
contato

*perceba que os pontos desapareceram
Isso acontece porque os pontos não são
o ponto de começo do ul
Em tese os pontos estão fora da tela

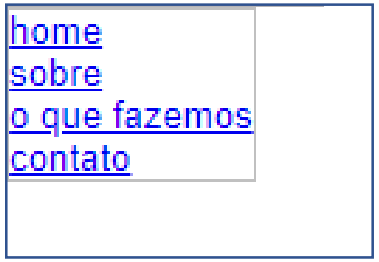
CONSTRUÇÃO DE UM MENU HORIZONTAL

- Para ter certeza que não teremos problemas com as marcações da lista, vamos removelas, além de dar uma borda para o ul e faze-lo flutuar a esquerda

CSS

```css
.menu{
    list-style:none;
     /*list-style serve para mudar ou retirar os pontos
    da lista*/
    /*aceita os valores: circle, square, disc(padrão), none
    entre outros*/
    border:1px solid #c0c0c0;
    float:left;
}
```

Resultado

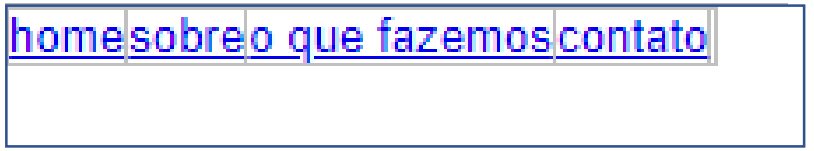


- Em seguida vamos fazer os li's flutuarem a esquerda, fazendo com que se torne um menu horizontal, além de dar uma borda para cada li

CSS

```css
.menu li{
    float:left;
    border-right:1px solid #c0c0c0;
}
```

Resultado



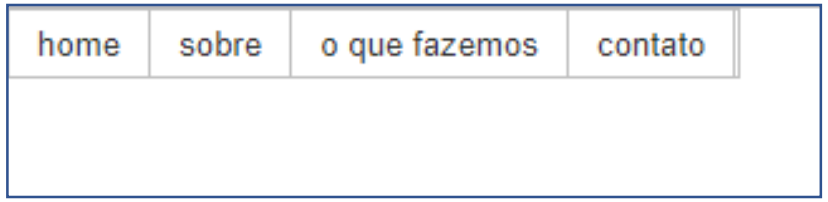
CONSTRUÇÃO DE UM MENU HORIZONTAL

- Em seguida vamos dar uma cor ao link, retirar sua linha inferior e dar um espaçamento interno.

CSS

```css
.menu li a{
    color:#333;
    text-decoration:none;
    /*sublinhados na palavra selecionada
    aceita os valores: underline.overline e none;*/
    padding:5px 10px;
    display:block;
}
```

Resultado

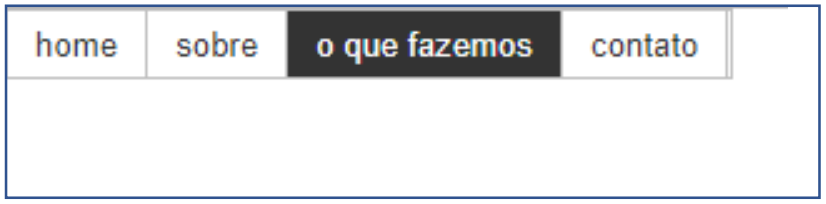


- Agora vamos dar uma característica diferente aos menus quando o mouse passar por cima deles.

CSS

```css
.menu li a:hover{
/*  hover serve para mudar a caracteristica do elemento
    quando o mouse passar por cima
    muito utilizado na estilização de links*/
    background:#333;
    color:#fff;
}
```

Resultado



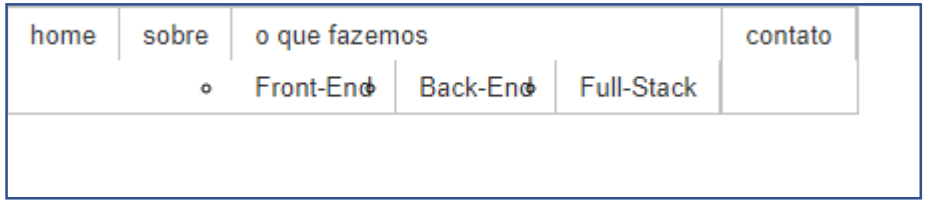
CONSTRUÇÃO DE UM SUBMENU(DROPDOWN)

- O menu está pronto, porém podemos fazer com que um submenu apareça abaixo de algum dos elementos assim que passarmos o mouse por cima. Para isso precisamos fazer algumas mudanças no código anterior temos de inserir uma nova lista dentro de um dos li's e devemos dizer que os li's terão uma posição relativa, para servir de referencia para posicionar o submenu.

HTML

```
<nav> <!-- nav é uma div semanticamente dedicada a menus -->
    <ul class="menu">
        <li><a href="#">home</a></li>
        <li><a href="#">sobre</a></li>
        <li><a href="#">o que fazemos</a>
            <ul> <!-- devemos inserir um novo ul dentro de um
                dos li's -->
                <li><a href="#">Front-End</a></li>
                <li><a href="#">Back-End</a></li>
                <li><a href="#">Full-Stack</a></li>
            </ul>
        </li>
        <li><a href="#">contato</a></li>
    </ul>
</nav>
```

Resultado



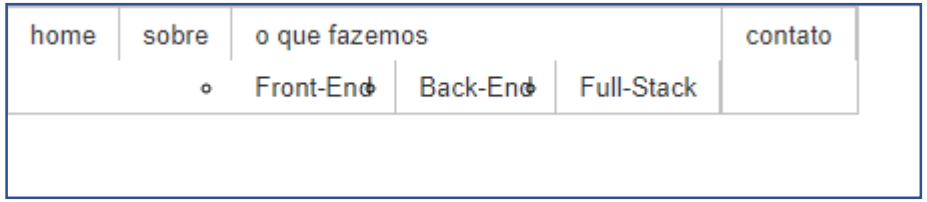
*Fica feio assim mesmo, não se preocupe

CONSTRUÇÃO DE UM SUBMENU(DROPDOWN)

CSS

```css
.menu li{
    float:left;
    border-right:1px solid #c0c0c0;
    position:relative;
}
```

Resultado



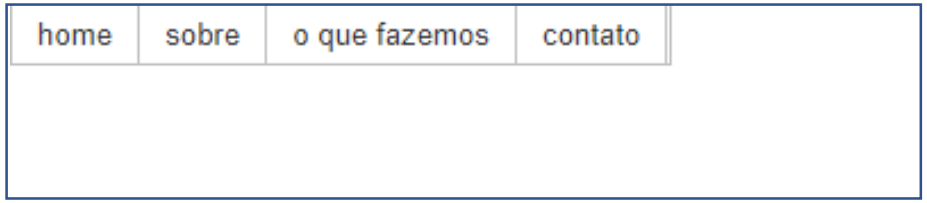
*Sem resultado imediato

- A seguir devemos dar uma posição para o menu interno e faze-lo desaparecer com um display:none.

CSS

```css
.menu li  ul{
    position:absolute;
    top:25px;
    left:0;
    background-color:#fff;
    display:none;
}
```

Resultado



CONSTRUÇÃO DE UM SUBMENU(DROPDOWN)

- Agora temos temos que dar um jeito de fazer o submenu aparecer um embaixo do outro. Primeiro damos um display:block no ul quando o mouse estiver em cima do li relativo, além de faze-lo perder a flutuação lateral e retirar o estilo de lista.

CSS

```css
.menu li:hover ul{
    display:block;
    float: none;
    list-style: none;
}
```

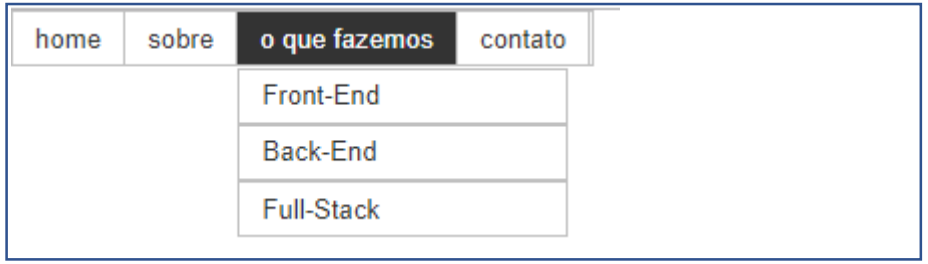
Resultado

- Por último, damos alguns acabamentos de borda e tamanho dos itens do submenu.

CSS

```css
.menu li ul li{
    border:1px solid #c0c0c0;
    width:150px;
}
```

Resultado



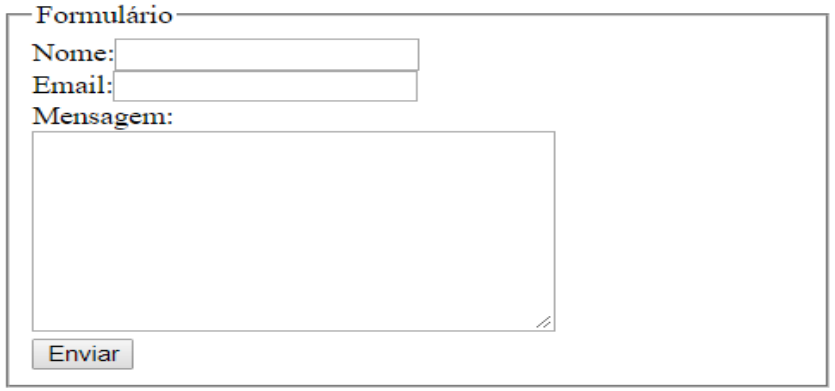
CONSTRUÇÃO DE UM FORMULÁRIO

- Primeiro precisamos de um formulário HTML bem simples.

HTML

```
<div id="area">
    <form id="formulario" autocomplete="off">
      <fieldset>
        <legend>Formulário</legend>
        <label for="nome">Nome:</label><input class="campo_nome" type="text" id="nome"><br>
        <label for="email">Email:</label><input class="campo_email" type="email" id="email"><br>
        <label for="mensagem">Mensagem:</label><br><textarea class="msg" cols="35" rows="8" id="
        mensagem"></textarea><br>
        <input class="btn_submit" type="submit" value="Enviar">
      </fieldset>
    </form>
</div>
```

resultado



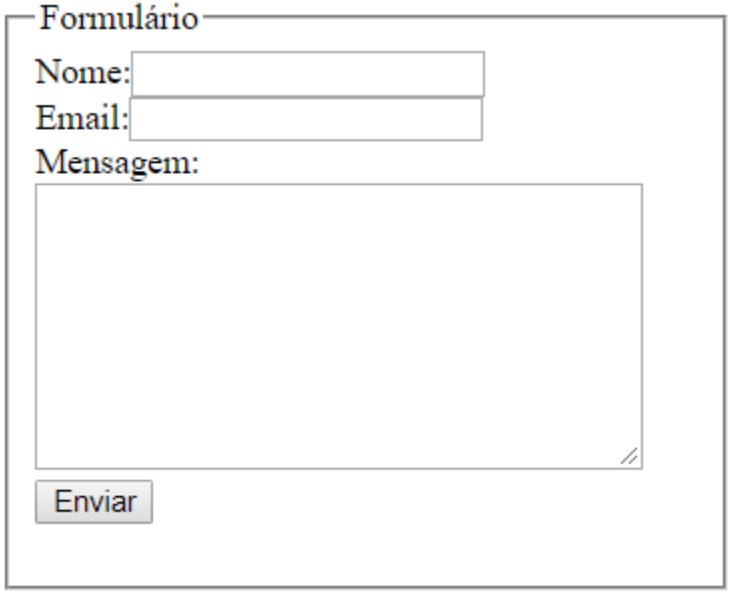
CONSTRUÇÃO DE UM FORMULÁRIO

- Vamos fazer algumas configurações para centralizar o formulário e diminuir o fieldset.

CSS

```css
area{
  position:relative;
  left:37%;
  top:29%;
  width:300px;
  height:270px;
}
#area #formulario{
  position:absolute;
  display:block;
}
/*as classes acima são opcionais
servem apenas para centralizar*/
fieldset{
  width:300px;
  height:250px;
}
```

resultado



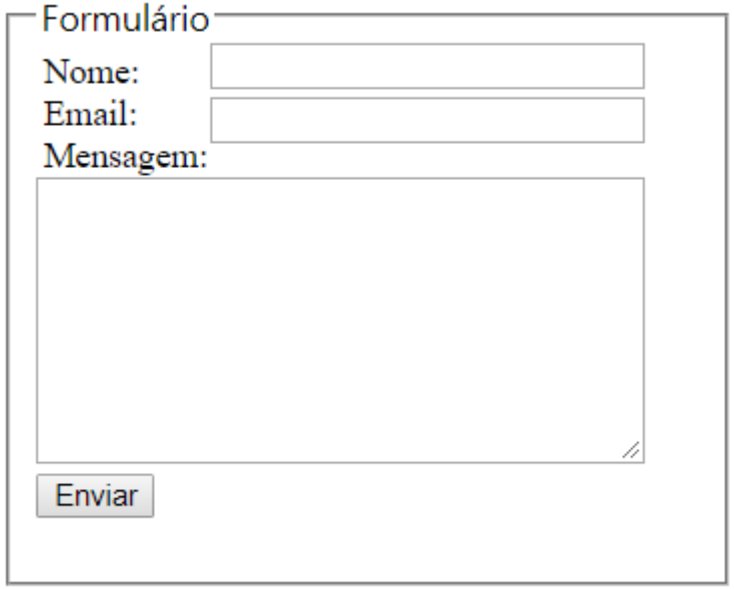
CONSTRUÇÃO DE UM FORMULÁRIO

- Vamos agora estilizar o texto do legend e padronizar os labels e os campos input afim de deixa-los alinhados.

CSS

```css
legend{
  font-style:bold;
  font-family: "Segoe UI","Arial","Times New Roman";
}
#formulario label{
  position:absolute;
  left:19px;
  margin-right:5px;
}
#formulario input.campo_nome{
  position:absolute;
  left:95px;
  top:23px;
  width:193px;
}
#formulario input.campo_email{
  position:absolute;
  left:95px;
  margin-top:2px;
  width:193px;
}
```

resultado



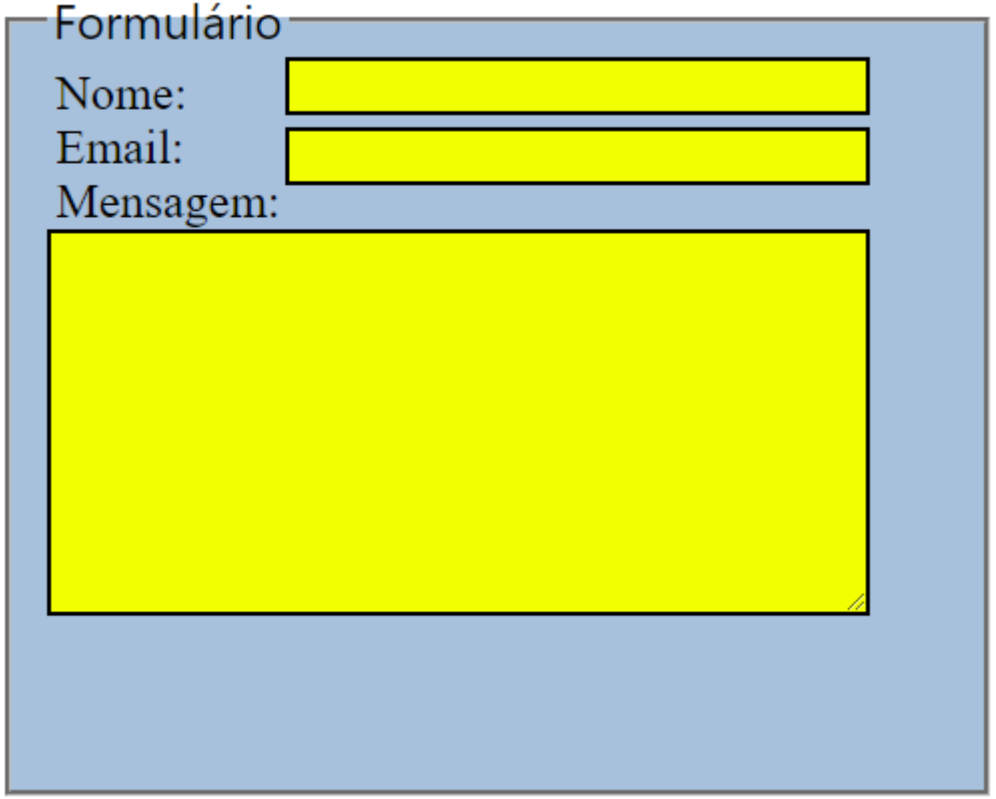
CONSTRUÇÃO DE UM FORMULÁRIO

- Agora que aprendemos a alinhar basta estilizar os campos com algumas cores e uma borda.

CSS

```css
#formulario input.campo_nome{
  position:absolute;
  left:95px;
  top:23px;
  width:193px;
  background-color:#f1ff00;
  border: 1px solid black;
}
#formulario input.campo_email{
  position:absolute;
  left:95px;
  margin-top:2px;
  width:193px;
  background-color:#f1ff00;
  border: 1px solid black;
}
#formulario textarea.msg{
  position:absolute;
  background-color:#f1ff00;
  border: 1px solid black;
}
```

resultado



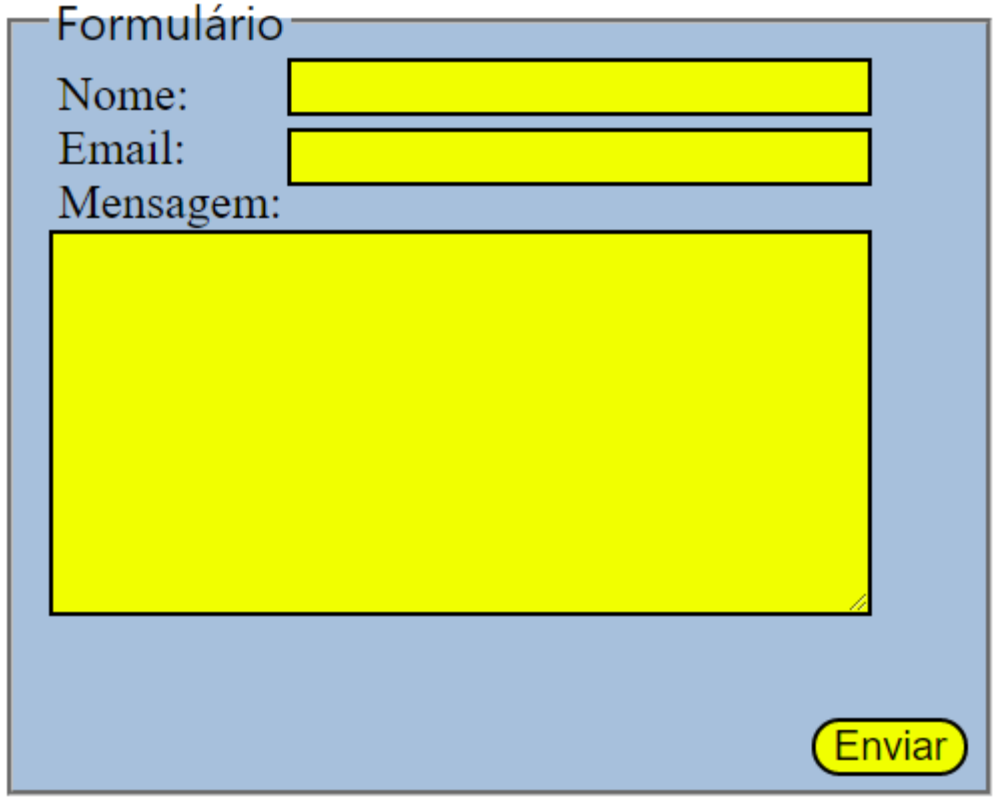
CONSTRUÇÃO DE UM FORMULÁRIO

- Por último posicionamos o botão de enviar e estilizamos ele.

CSS

```css
#formulario input.btn_submit{
  border: 1px solid black;
  position:absolute;
  bottom: 10px;
  right:10px;
  background-color:#f1ff00;
  border-radius: 10px;
}
```

resultado



REFERÊNCIAS

- W3Schools – https://www.w3schools.com/
- Developer Mozilla – https://developer.mozilla.org/
- Canal CBFCursos – https://www.youtube.com/channel/UCqHIWCQSq0yeE-1nbcRnt2w
- Curso em vídeo – https://www.cursoemvideo.com.br
- Devmedia – https://www.devmedia.com.br