**Declaração de cores com uso da função CSS***rgb()*

rgb é abreviatura para:  
r = red (vermelha)  
g = green (verde)  
b = blue (azul)

O valor CSS expresso pela função *rgb(red, green, blue)* indica uma cor obtida com a mistura de uma quantidade *red* da cor vermelha com *green* da cor verde e *blue* da cor azul.

Duas são as maneiras de se definir a quantidade de cada uma das três cores (os parâmetros da função:

Uma faixa de números de 0 (zero) até 255  
Uma faixa de porcentagens de 0% até 100%

**Não é válido** usar em uma definição de cor uma mistura de números e porcentagens. Observe a seguir as duas sintaxes válidas e uma inválida.

**CSS**

/\* Sintaxes válidas \*/

rgb(145, 230, 50)

rgb(20%, 0%, 70%)

/\* Sintaxe inválida \*/

rgb(255, 20%, 120)

Formas de declarar as cores abaixo:

#primeiro-titulo {

    color:magenta;

}

.paragrafo {

    color:#df0ed4;

}

#segundo-titulo {

    color: rgb(123, 65, 22);

}

**Colocando imagens em background**

## O que é o Container

O container é um componente que garante que o seu layout vai ficar alinhado corretamente na página. Ele pode definir as margens laterais da página, ou deixar sem margens e, também, posiciona o conteúdo no centro do browser.

O container é invisível. Mas, você pode saber que ele existe pelos limites que ele coloca na página.

Para criar layouts com o Bootstrap, você precisa entender como usar o container, que é um dos componentes base do framework.

<div class="box" id="container">

        <p class="paragrafo-box">conteúdo do box</p>

        <p class="paragrafo-box">conteúdo do box></p>

        <p class="paragrafo-box">conteúdo do box></p>

        <p class="paragrafo-box">conteúdo do box></p>

    </div>

RESULTADO NA TELA:



**Colocando imagens em background**

Arquivo.html

<body>

    <div id="background">

       <h2>Banner Principal</h2>

    </div>

Arquivo css

#background {

    background-image: url("../img/flores.jpg");

    background-size: cover;

    height: 1300px;

}

#background h2 {

    color:blue;

    text-align: center;

}

Background-image: Serve para puxar a imagem da pasta do projeto para a tela.

Você deve invoca-la assim:

background-image: url("../img/flores.jpg");

Url: significa o caminho de origem

.. Significa que você voltou o diretório para puxar o arquivo

/ serve para referenciar a pasta do arquivo e clicando outra vez, você chama o arquivo se lá ele estiver.

height: 1300px;

Serve para ajustar a altura da imagem.

background-size: cover;

Serve para aproveitar ao máximo a imagem no fundo da tela. Para ela ficar por igual entendeu? Este comando é usado bastante na manipulação de imagens.

**Conceito de Box Model**

**Meu texto está aqui!**

Este conceito nunca deve sair da sua cabeça ao mexer com CSS e alinhar e manipular os elementos ok?!

Vamos a explicação:

**Padding:** Representa a parte branca, que significa que o espaçamento entre o texto e as extremidades da parte branca. Então, a cada valor inserido para distanciamento, você deve imaginar a tela deste modo!

EX: padding: 20px;

**Border:** Representa a tarja laranja que significa toda a borda da parte laranja!

EX: border: 2px solid;

**Margin:** Representa a tarja azul que significa que o espaçamento entre o azul e o laranja conforme a imagem acima. Então, a cada valor inserido para distanciamento, você deve imaginar a tela deste modo entendeu!

EX: margin: 50px;

**Bordas com CSS**

Normalmente a bordar mais utilizada que vemos por ai é a Solid, mas há outras que não são muitos usadas.

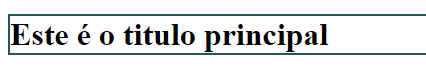
Vamos ver os tipos:

#primeiro-titulo {

    border: 2px solid #245858;

}

Resultado abaixo:



#segundo-titulo {

    border: dotted red;

}

Resultado na tela:



#segundo-titulo {

    border: dashed red;

}

Resultado na tela:



#segundo-titulo {

    border: double red;

}

Resultado na tela:

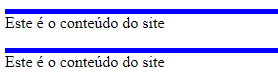


.paragrafo {

    border-top: 5px solid blue;

}

A borda será aplicada somente no topo e será de 5px solida azul.

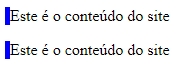


.paragrafo {

    border-left: 5px solid blue;

}

Resultado na tela:



.paragrafo {

    border-right: 10px solid blue;

}

Resultado na tela:

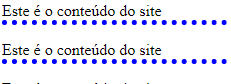


.paragrafo {

    border-bottom: 5px dotted blue;

}

Resultado na tela:



**Margens**

O elemento margin, manipula as margens da sua tela da web.

#primeiro-titulo {

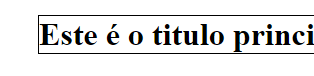
    border: 1px solid #000;

    margin: 30px;

}

Resultado na tela do espaçamento, mas será apenas para as laterais:





TIPOS DE MARGENS:

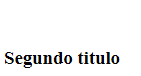
#segundo-titulo {

    margin-top: 50px;

}

Resultado na tela do espaçamento somente do topo:





#segundo-titulo {

    margin-left: 20px;

}

Resultado na tela do espaçamento somente da esquerda:



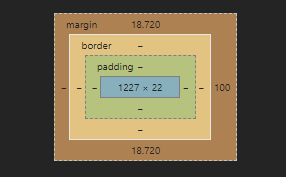


#segundo-titulo {

    margin-right: 100px;

}

Resultado na tela do espaçamento somente da direita:

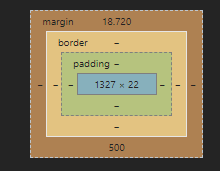


#segundo-titulo {

    margin-bottom: 500px;

}

Espaçamento da parte inferior :



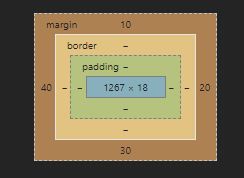
Da para você determinar os espaçamentos de uma vez só assim:

.paragrafo {

    margin: 10px 20px 30px 40px;

}

**Quer dizer:** 10px (topo), 20px (direita), 30px(para baixo), 40px(Esquerda)

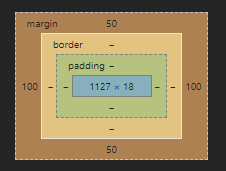


Esse você padroniza Topo e a margem inferior, além da esquerda e direita assim:

.box {

    margin: 50px 100px;

}



**Explorando os paddings**

#primeiro-titulo {

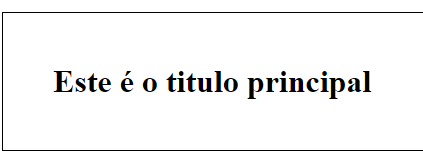
    border: 1px solid #000;

    padding: 50px;

}

É o espaçamento do preenchimento externo, para não deixar o texto colado. Veja abaixo:





#segundo-titulo {

    padding-top: 20px;

    padding-right: 10px;

    padding-bottom: 20px;

    padding-left: 50px;

    border: 2px solid red;

}

.paragrafo {

    padding: 5px 10px 15px 20px;

}

O Funcionamento é igual ao margin, só muda as posições!

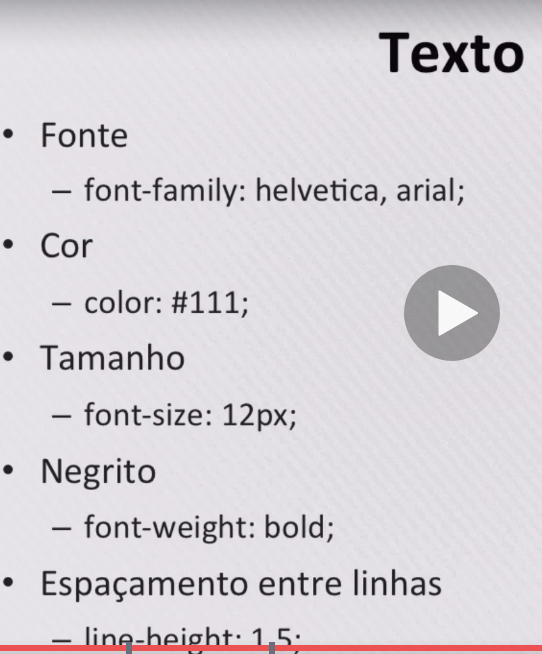
**Largura e altura**

Definindo altura e largura de um elemento.

Este tipo de propriedade é mais aconselhável ser utilizado em container, ou seja outros objetos que encapsulam o Html.

**Alterando fontes**

Fontes são aonde alteramos as letras do nosso projeto de web site. Exemplo: estilo de letra e tamanho de letra.



Alterando os estilos:

Para alterar os estilos, usamos o comando font-family.

#primeiro-titulo {

    font-family: arial;

}

Resultado abaixo:

# Este é o título principal

#background h2 {

    font-family: georgia;

}

Alterando o tamanho da letra:

Usamos o font-size.

  font-size: 34px;

Para deixar todas as letras com o mesmo tamanho, usamos o comando courier:

#segundo-titulo {

    font-family: courier;

}

**ALINHANDO OS TEXTOS**

Essa propriedade é muito importante no mundo dos front end.

Para alinhar os textos e manipula-los, usamos os seguintes comandos e códigos:

Alinhamento no centro **text-align: center;**

#primeiro-titulo {

    font-family: arial;

    text-align: center;

}

# Este é o título principal

Alinhamento à direita text-align: right;

#background h2 {

    font-family: georgia;

    font-size: 34px;

    text-align: right;

}

## Banner Principal

Existe o comando text-decoration, para decorações também.

Veja os tipos:

// text-decoration: underline

#background h2 {

    font-family: georgia;

    font-size: 34px;

    text-align: right;

    text-decoration: underline;

}

## Banner Principal

// text-decoration: line-through

#primeiro-titulo {

    font-family: arial;

    text-align: center;

    text-decoration: line-through;

}

# ~~Este é o título principal~~

PARA DEIXAR AS LETRAS ESPAÇADAS UMAS DAS OUTRAS:

#segundo-titulo {

    font-family: courier;

    letter-spacing: 5px;

}

### **Segundo titulo**

### **Segundo titulo**

**A propriedade display**

Outra propriedade muito usada em CSS é o display, vamos ver como ela funciona de fato.

Display: inline-block; Serve para deixar os elementos um do lado do outro. Veja:

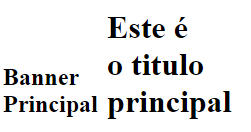
#background, #primeiro-titulo {

    display: inline-block;

    width: 100px;

}

O Texto abaixo ficou em linha e em bloco!



Importante: O comando <span> tem como característica deixar o seu texto em linha já por natureza! Ele não apresenta a forma de bloco.

Mas da para deixar <span> em bloco, fazendo assim:

Arquivo html

<span>Testando</span>

<span>Bloco</span>

**Testando Bloco**

Arquivo CSS

span {

    display: block;

}

**Testando**

**Bloco**

Existe uma propriedade do display que apaga ou oculta algo que queira, mas o seu html permanece no código.

Veja:

#background {

    display: none;

}

**Posições no CSS**

No CSS existem as propriedades de posições de textos e imagens, chamado de

**Position: absolute;** significa que o texto vai se referenciar no alinhamento, com base no Body do html. Toda regra que você impor a ele, o mesmo vai pegar como base o body entendeu.

#primeiro-titulo {

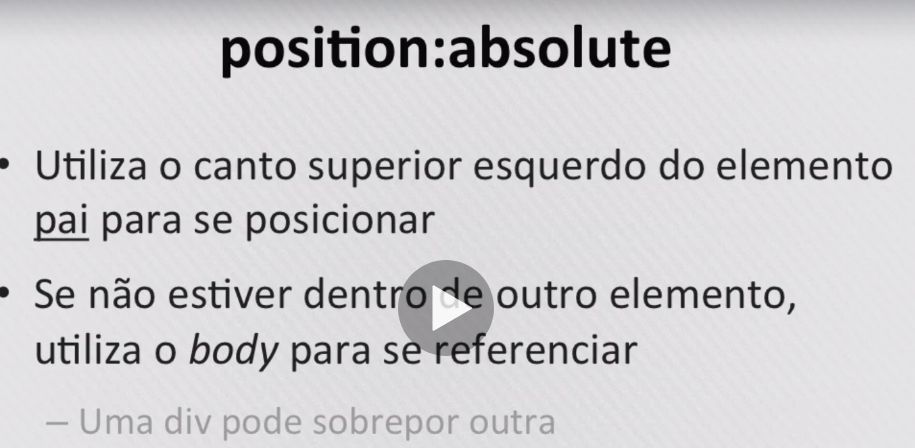
    position: absolute;

    top: 0;

    left: 300px;

}

Position: relative; significa que o texto vai se referenciar por outra base criada por você, ou seja, outra div e não mais o body como era antes entendeu?



É a agora vamos posicionar os elementos na tela da forma que queremos com o position: relative.

#pai {

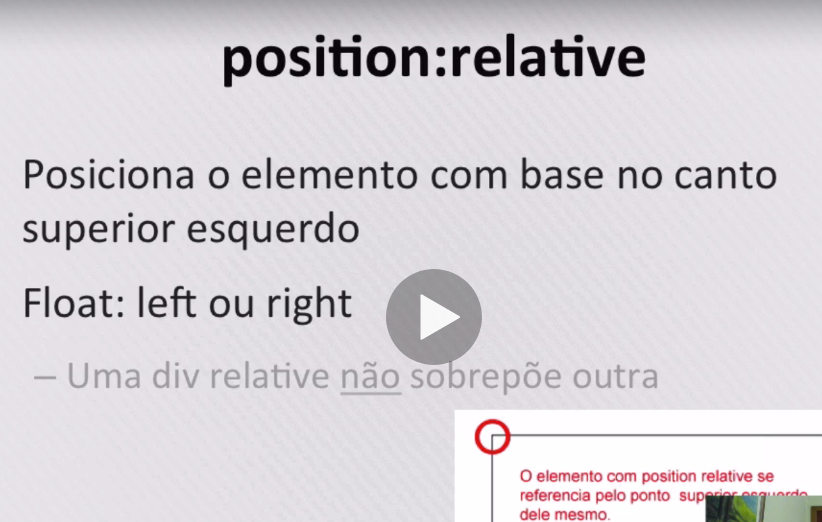
    position: relative;

    width: 200px;

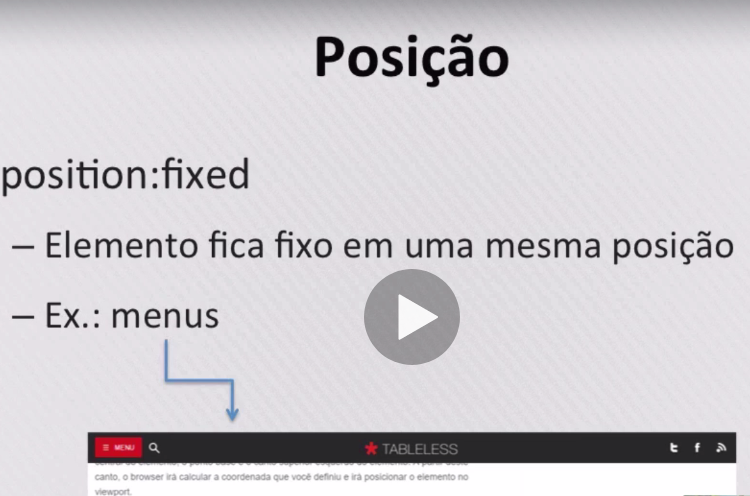
    height: 200px;

    top: 30px;

}



Position: fixed; É quando queremos um elemento fixo em nossa tela. Você movimenta a barra de rolagem, mas o elemento fica estático!



**Arquivo html**

<p id="fixed">fixo</p>

**Arquivo CSS**

#fixed {

    position:fixed;

    top: 10px;

    right: 10px;

    background-color: red;

    color: #fff;

}



DICAS:

Limitador de largura **max-width: 1200px;**

Isso é muito utilizado para as Tvs inteligentes de hoje em dia e na indústria de software.

Centralizar o elemento container na nossa tela por igual

**margin: 0 auto;**

Tirar o sublinhado da tag <a> : Home - Home

**Text-decoration: none;**

**Deixar negrito**

**Font-weight: bold;**

**Tempo que demorar para dar animação ao botão.**

**Transition: .5s;**

**Animação quando passar o mouse em cima do botão**

**.btn:hover {**

**}**

**Há três Formas de declarar o CSS:**

**Inline ( style=””)**

**Ex: <div style=”height:100px”><div>**

**Ex: <p style=”color:red”>Meu texto<div>**

**Tag <style> declaradas no próprio html**

<head>

    <title>Documento</title>

    <style>

     p {

         color: red;

       }

    </style>

</head>

**Importação do link**

**O jeito mais recomendado e trabalhado hoje em dia com CSS, pois você trabalha as tags do CSS em um arquivo separado do html.**

**Ex:**

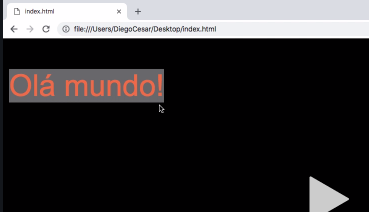
<head>

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

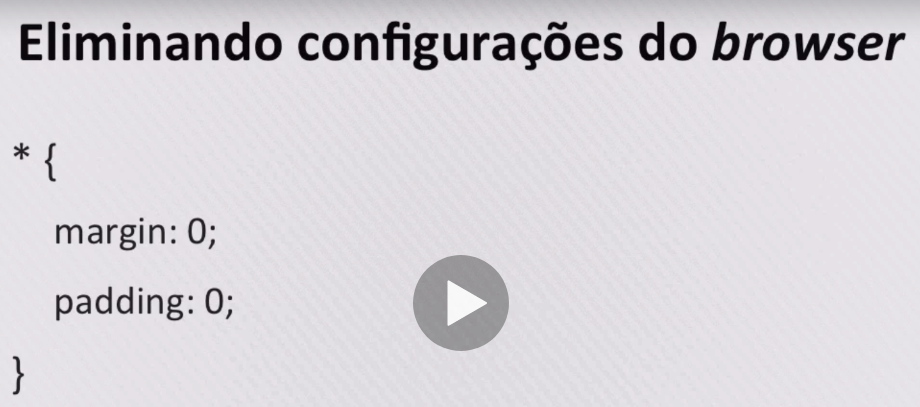
</head>

**Eliminando as configurações do browser no inicio do projeto, para colocar as nossas dimensões ao projeto. O que acontece é que por padrão as páginas web já vem com espaçamentos pré estipulados.**

**Veja o espaçamento do topo e margin padrão da página web abaixo:**

****

**Quando eliminamos essas configurações, damos uma espécie de RESET. Isso é uma boa prática! Para fazer isso, fazemos desta forma:**

****

Explicações: \* significa toda a página. Margin 0 é que definimos a margem colada, assim com o padding.

COLOCAR DIV LADO A LADO

Fazemos isso com a propriedade Float.

**#div esquerda {**

Background-color: red;

height: 200px;

width: 50px;

float:right;

**}**

**CONTAINER**

**É uma div mãe que pega todo o conteúdo da página e altera tudo que está dentro dela.**

****

**As div container deve ser definidas como class, pois pode aparecer em varias páginas ao mesmo tempo.**

**.container {**

**width: 100px;**

**margin: 0 auto;**

**}**

**Explicação:** quando você coloca 0 auto, quer dizer que você quer que o posicionamento fique centralizado e alinhado referente a margem.

**FLOAT**

A propriedade **float** do CSS determina que um elemento deve ser retirado do seu fluxo normal e colocado ao longo do lado direito ou esquerdo do seu container, onde textos e elementos em linha irão se posicionar ao seu redor.

**left**

É uma palavra-chave que indica que o elemento deve flutuar à esquerda do bloco.

**right**

É uma palavra-chave que indica que o elemento deve flutuar à direita do bloco.

**none**

É uma palavra-chave que indica que o elemtno não deve flutuar.

Observação: Ao usar os elementos float: left e right, devemos usar a propriedade clear: both; para assegurar o seu alinhamento no bloco. Para isso vamos até o html e vemos qual bloco de códigos com div está abaixo e ai inserimos o clear: both;