

**Resumo**

Tag	Descrição
<html>	Define um documento HTML
<head>	Define informações de cabeçalho. Não é mostrada na janela do navegador.
<body>	Define o corpo do documento
<h1> a <h6>	Define título 1 a título 6
<p>	Define um parágrafo
 	Insere uma única quebra de linha
<hr>	Define um filete horizontal
<!-->	Define um comentário
	Define texto em negrito
<big>	Define texto grande
	Define texto enfatizado;
<i>	Define texto em itálico
<small>	Define texto pequeno
	Define texto forte
<sub>	Define texto subscrito
<sup>	Define texto superescrito
<ins>	Define texto inserido
	Define texto cancelado
<s>	Desaprovada. Use em vez
<strike>	Desaprovada. Use em vez
<u>	Desaprovada. Use estilos em vez
<a>	Define uma âncora
	Define uma lista ordenada
	Define uma lista não ordenada
	Define um item de lista
<dl>	Define uma lista de definições
<dt>	Define um termo de definição
<dd>	Define uma descrição de definição
	Define uma lista ordenada
<tb>	Define uma tabela
<tr>	Define uma linha da tabela
<td>	Define uma coluna da tabela

**Entidades de Caractere Mais Comuns:**

Resultado	Descrição	Nome da Entidade	Número da Entidade
	espaço não separável	 	
<	menor do que	<	<
>	maior do que	>	>
&	e comercial	&	&
"	aspas	"	"
'	apóstrofe		'

Algumas Outras Entidades de Caractere Comumente Usadas:

Resultado	Descrição	Nome da Entidade	Número da Entidade
¢	cent	¢	¢
£	libra	£	£
¥	yen	¥	¥
§	parágrafo (em inglês section)	§	§
©	copyright	©	©
®	marca registrada	®	®
×	multiplicação	×	×
÷	divisão	&division;	÷

**Tags e seus atributos**

` `

Alterna tamanho, cor e tipo de fonte exibida

- *size* – O tamanho da fonte varia de 1 a 7
- *color* – A cor da fonte #RRGGBB
- *face* – O tipo da fonte

``

Insera uma imagem in-line no documento e inclui atributos comuns

- *usemap* – Um mapa de imagens do lado cliente
- *src* – O URL da imagem
- *alt* – Uma string de texto que será exibida em navegadores que não possam suportar imagens
- *align* – Determina o alinhamento de uma determinada imagem: top, middle, bottom, left e right
- *height* – É a altura sugerida em pixels
- *width* – É a extensão sugerida em pixels
- *vspace* – O espaço entre a imagem e o texto acima e abaixo dela
- *hspace* – O espaço entre a imagem e o texto à esquerda e à direita dela
- *border* – Largura da borda

`<table> </table>`

Cria uma tabela

- *background* – Imagem de plano de fundo
- *bgcolor* – Cor de plano de fundo
- *border* – Largura da borda em pixels
- *cols* – Número de colunas
- *cellpadding* – Espaçamento nas células
- *cellspacing* – Espaçamento entre as células
- *width* – Largura da tabela
- *align* – Alinhamento da tabela: left, center, right
- *bordercolor* – Cor na borda da tabela

`<caption> </caption>`

A legenda para a tabela



`<tr> </tr>`

Uma linha na tabela

- *align* – O alinhamento horizontal do conteúdo das células dentro dessa linha com os valores possíveis left, right, center, justify e char
- *bgcolor* – Cor de fundo
- *valign* – o alinhamento vertical do conteúdo das células dentro dessa linha com os valores possíveis top, middle, bottom e baseline
- *background* – Figura como plano de fundo

`<th> </th>`

Um cabeçalho de célula da tabela

- *align* – Alinhamento horizontal
- *valign* – Alinhamento vertical
- *bgcolor* – Cor de plano de fundo
- *rowspan* – O número de linhas pelo qual essa célula se expandirá
- *colspan* – O número de colunas pelo qual essa célula se expandirá
- *nowrap* – Desliga o enquadramento de texto em uma célula

`<td> </td>`

Define uma célula de dados da tabela

- *align* – Alinhamento horizontal
- *valign* – Alinhamento vertical
- *bgcolor* – Cor de plano de fundo
- *rowspan* – O número de linhas pelo qual essa célula se expandirá
- *colspan* – O número de colunas pelo qual essa célula se expandirá
- *nowrap* – Desliga o enquadramento de texto em uma célula
- *width* – Largura da célula
- *height* – Altura da célula



Aula 05 e 06

Tag “div”

O elemento de divisão [HTML](#) `<div>` é um container genérico para conteúdo de fluxo, que de certa forma não representa nada. Ele pode ser utilizado para agrupar elementos para fins de estilos (usando **class** ou **id**), ou porque eles compartilham valores de atributos, como **lang**. Ele deve ser utilizado somente quando não tiver outro elemento de semântica.

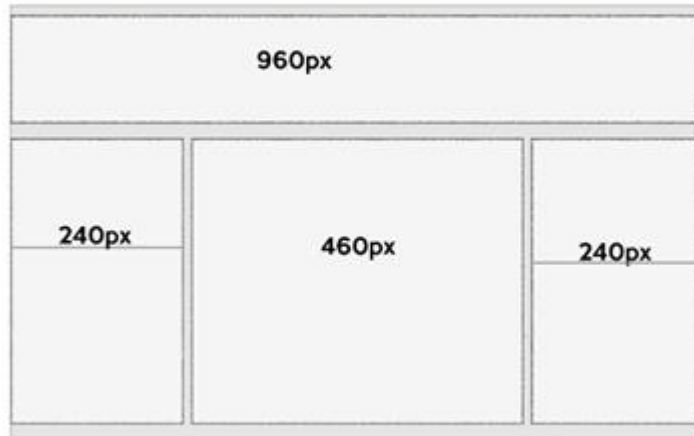
O Elemento HTML Div costuma ter entre seus principais atributos:

- **id** – Define uma ID;
- **class** – Define uma classe;
- **title** – Define um título;
- **height** – Define uma altura;
- **width** – Define uma largura.
- **Align** – Define o alinhamento*
- **Bgcolor** – Define a cor do fundo*

Porém, com exceção do ID, da class e do title, todos os demais atributos já são considerados depreciados. Muitos destes atributos estão contidos no atributo “style”.

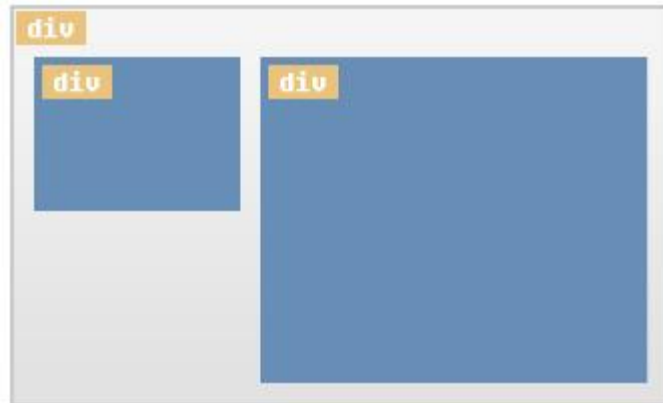


Layout fixo



Layout fixo







Exemplo 1

Tag “div” e “span”

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="utf-8">
  <title>Aula 05 e 06 - HTML - Exemplo 1</title>
</head>
<body>

  <h1>Tags "div" e "span" </h1>

  <div>div numero 1</div>
  <div>div numero 2</div>
  <div>div numero 3</div>

  <span>span numero 1</span>
  <span>span numero 2</span>
  <span>span numero 3</span>

</body>
</html>

<!--
  tanto a div quando span são tag geralmente é usada para agrupar elementos
  eles não fazem "nada".

  mas quando agrupamos com div. por padrão a div
  pula de linha ao final

  diferente do span que não pula de linha ao final.
-->
```

<https://logbook.itstep.org/#/home> X Nova guia

← → ↻ ⓘ Arquivo | E:/STEP/Modulo3/

Tags "div" e "span"

```
div numero 1
div numero 2
div numero 3
span numero 1 span numero 2 span numero 3
```




Exemplo 2

Tag “div”

```
<!DOCTYPE html>
<html lang="pt-br">

<head>
  <meta charset="utf-8">

  <title>Aula 05 e 06 - HTML - Exemplo 2</title>
</head>

<body>

  <div>conteúdo 1</div>
  <div bgcolor="blue" width="800px" height="100px" border="1">conteúdo 2</div>
  <div bgcolor="blue" width="800px" height="100px">conteúdo 3</div>
  <div bgcolor="blue" width="800px" height="100px">conteúdo 4</div>

  <div
    style="width:98%;height:140px;border:1px solid #000;float:left;padding:4px;ba
ckground:#000;color:#fff;">
    conteúdo 5</div>
  <div style="width:98%;height:auto;border:1px solid #000;float:left;padding:4px;">
</div>

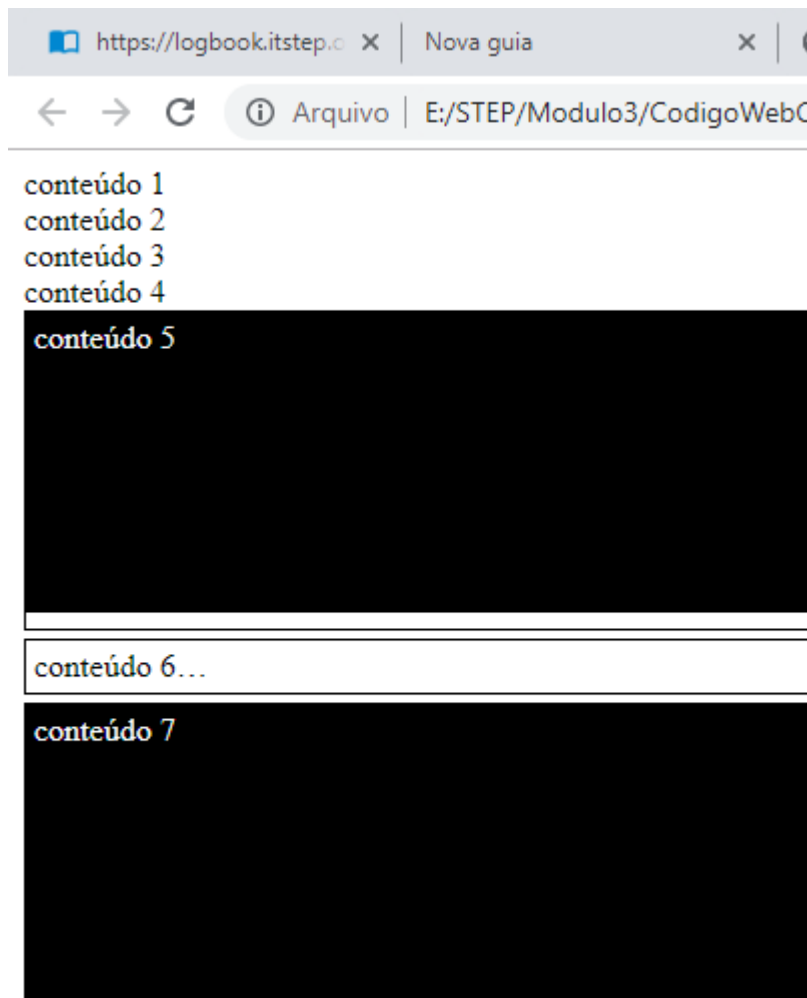
  <div style="width:98%;height:auto;border:1px solid #000;float:left;padding:4px;ma
rgin-top:4px;margin-bottom:4px;">
    conteúdo 6...
  </div>

  <div
    style="width:98%;height:140px;border:1px solid #000;float:left;padding:4px;ba
ckground:#000;color:#fff;">
    conteúdo 7 <br>

  </div>

</body>

</html>
```



Obs: o resultado final não foi o esperado



Exemplo 3

Tag “div”

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="utf-8">

  <title>Aula 05 e 06 - HTML - Exemplo 3</title>
</head>
<body>
  <h1>Criando div com style</h1>

  <div style="position:absolute; margin-left:10px; margin-
top:10px; width:300px; height:140px; z-index:1; background-color:#00BFFF; text-
align:center;"> </div>
  <div style="position:absolute; margin-left:50px; margin-
top:20px; width:300px; height:140px; z-index:1; background-
color:rgb(0, 255, 128); text-align:center;"> </div>
  <div style="position:absolute; margin-left:60px; margin-
top:150px; width:300px; height:140px; z-index:1; background-
color:rgb(251, 255, 0); text-align:center;"> </div>
  <div style="position:absolute;
    margin-left:150px;
    margin-top:100px;
    width:300px;
    height:140px;
    z-index:1;
    background-color:rgb(255, 0, 200);
    text-align:center;">

  </div>

</body>
</html>

<!--
```

A tag div não é mais do que um container para outras tags. Aqui são alguns dos atributos div:

```
id
class
```

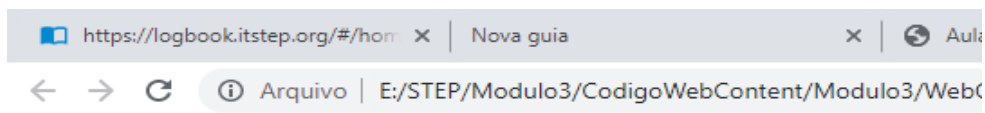


```
title
style
height
width
```

atributos do style de uma div

```
/* Definimos a largura da página */
width:1000px;
/* Definirmos a altura da página*/
height:700px;
/* Aqui definiremos a posição que a página ocupará na tela, nesse caso estamos defini
ndo que fique centralizado na tela*/
/*top : posição de cima */
margin-top:0;
/* left: posição de esquerda*/
margin-left:auto;
/* bottom : posição de baixo */
margin-bottom:0;
/* right : posição de direita */
margin-right:auto;
/* Aqui definiremos a cor de fundo da página */
background-color:#666;

-->
```



Criando div com style





Exemplo 4

Tag “div”

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="utf-8">

  <title>Aula 05 e 06 - HTML - Exemplo 4</title>
</head>
<body style="background-color: black" >

  <div style="position:absolute; margin-
left:0px; width:1000px; height:150px; border:0; background-color:#00BFFF; text-
align:center"> Titulo </div>
  <div style="position:absolute; margin-left:0px; margin-
top:160px; width:130px; height:600px; border:0; background-
color:rgb(0, 255, 128); text-align:center"> Menu </div>
  <div style="position:absolute; margin-left:140px; margin-
top:160px; width:860px; height:600px; border:0;background-
color:rgb(251, 255, 0); text-align:center"> conteudo </div>
  <div style="position:absolute;
margin-left:0px;
margin-top:770px;
width:1000px;
height:50px;
background-color:rgb(255, 0, 200);
text-align:center"> rodapé</div>

</body>
</html>

<!--
```

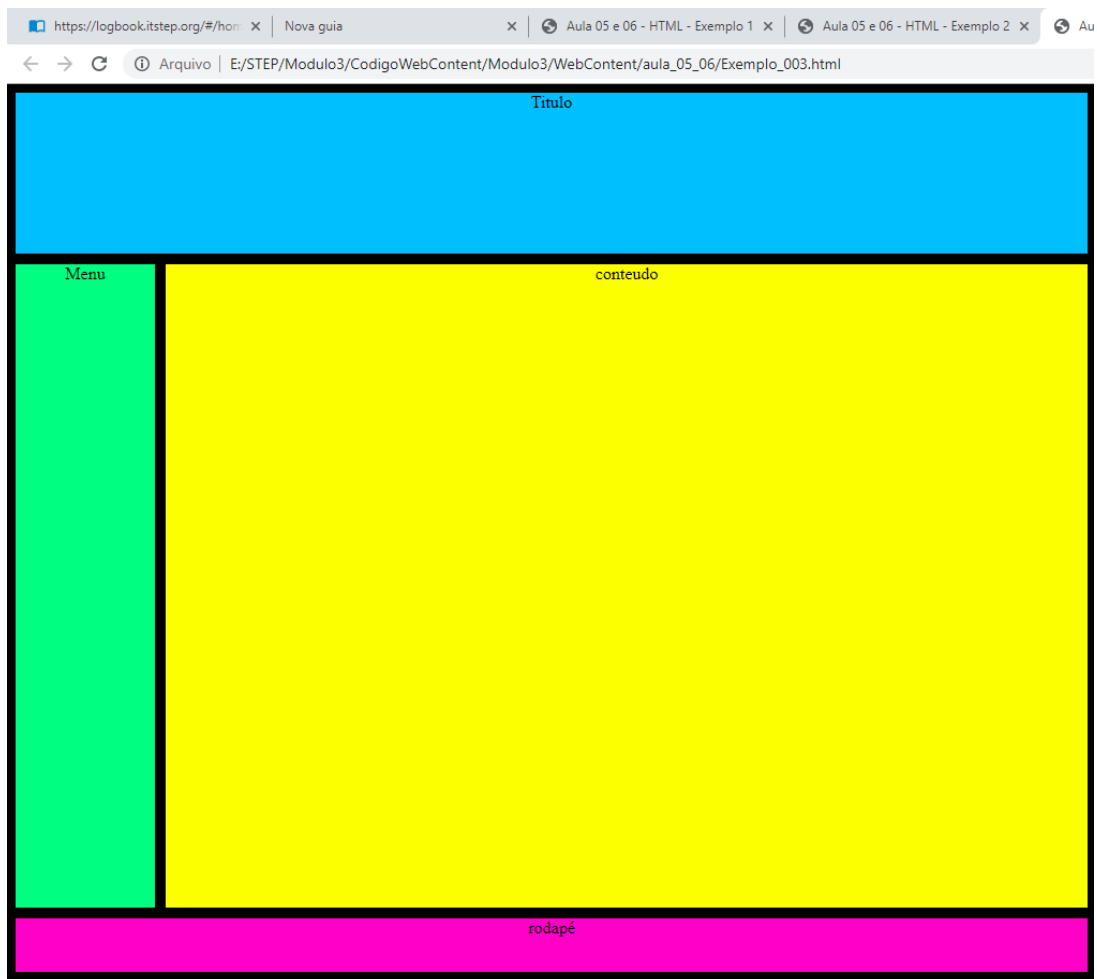
A tag div não é mais do que um container para outras tags. Aqui são alguns dos atributos div:

```
atributos do style de uma div
/* Definimos a largura da página */
width:1000px;
/* Definirmos a altura da página*/
height:700px;
```



```
/* Aqui definiremos a posição que a página ocupará na tela, nesse caso estamos defini-  
ndo que fique centralizado na tela*/  
/*top : posição de cima */  
margin-top:0;  
/* left: posição de esquerda*/  
margin-left:130px;  
  
/* right : posição de direita */  
margin-right:auto;  
/* Aqui definiremos a cor de fundo da página */  
background-color:#666;
```

A largura de um elemento no HTML/CSS é calculado somando algumas propriedades:
 $\text{padding} + \text{margin} + \text{border} = \text{width}$
por enquanto use px em vez de % para definir a posição das DIV
-->





Exemplo 5

O elemento

HTML <style> contém informações de estilo para um documento ou uma parte do document o. As informações de estilo específico estão contidas dentro deste elemento, geralmente no [CSS](#).

Uma folha de estilo é dita inline, quando as regras CSS são declaradas dentro da tag de abertura do elemento HTML com uso do atributo `style`.

```
<p style="color:#000; margin: 5px;">
```

Sintaxe:< tag style="propriedade:valor;">

Sintaxe:< tag style="propriedade:valor; outra_propriedade:valor;">



Exemplo 6

Tag “div”

```
<!DOCTYPE html>
<html lang="pt-br">

<head>
  <meta charset="utf-8">

  <title>Aula 05 e 06 - HTML - Exemplo 6</title>
</head>

<body style="background-color: black; margin-top: 0px;margin-left: 0px;margin-
right: 0px;margin-bottom: 0px">

  <div
    style="position:absolute; margin-
left:0px; width:100%; height:150px; border:0; background-color:#00BFFF; text-
align:center">
    Titulo </div>
  <div
    style="position:absolute; margin-left:0px; margin-
top:160px; width:130px; height:600px; border:0; background-
color:rgb(0, 255, 128); text-align:center">
    Menu </div>
  <div
    style="position:absolute; margin-left:140px; margin-
top:160px; width:1120px; height:600px; border:0;background-
color:rgb(251, 255, 0); text-align:center">
    conteudo </div>
  <div
    style="position:absolute; margin-left:0px; margin-
top:770px; width:100%; height:50px; background-color:rgb(255, 0, 200); text-
align:center">
    rodapé</div>

</body>

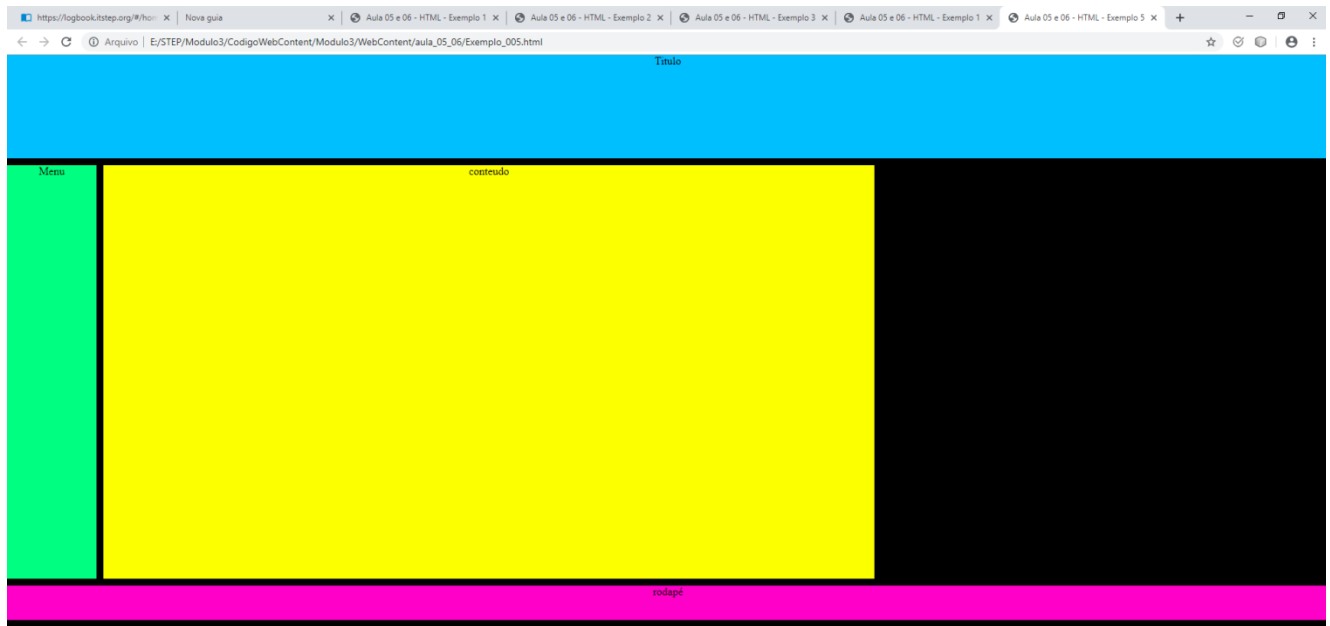
</html>
```




Professor: Luciano

Módulo HTML/CSS

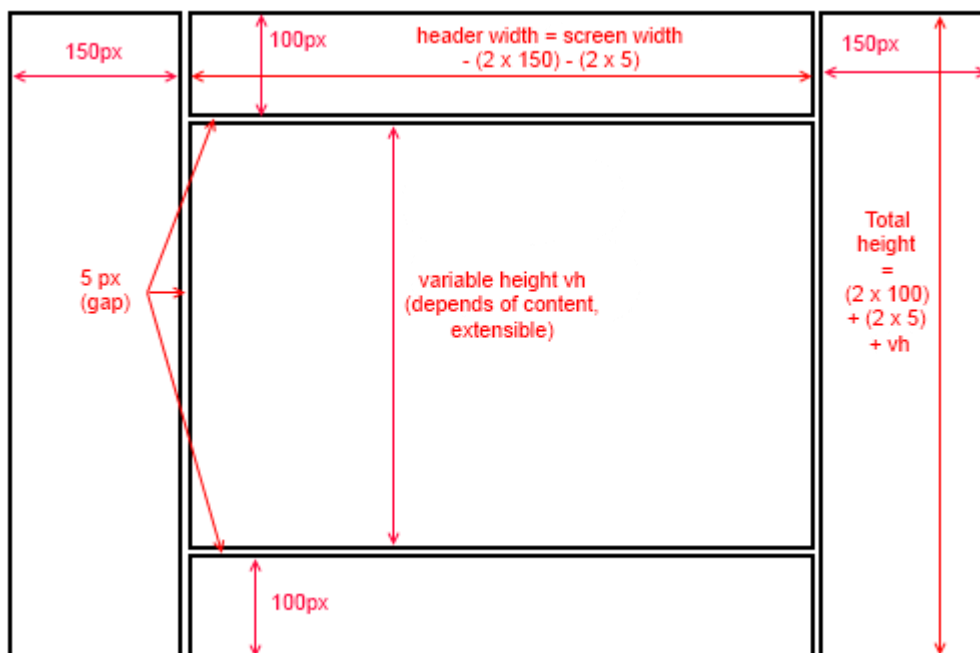
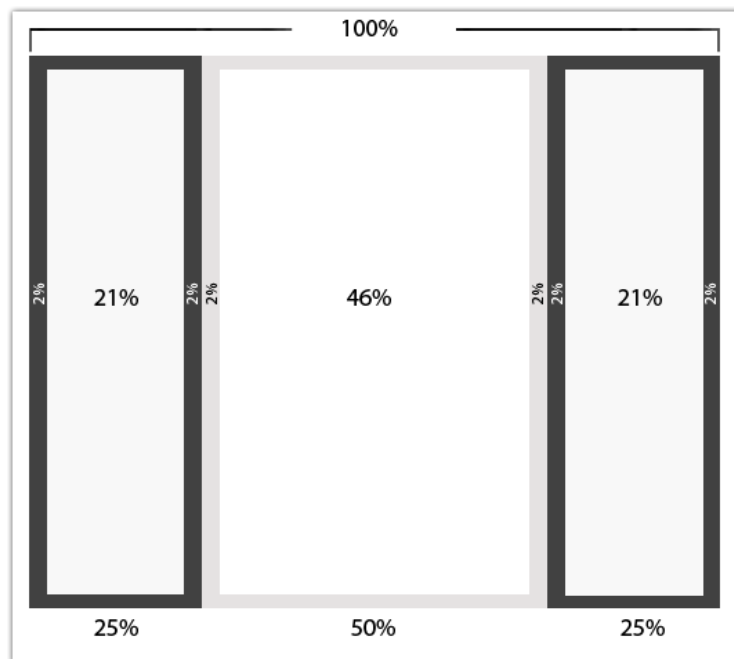
Aula 05 e 06





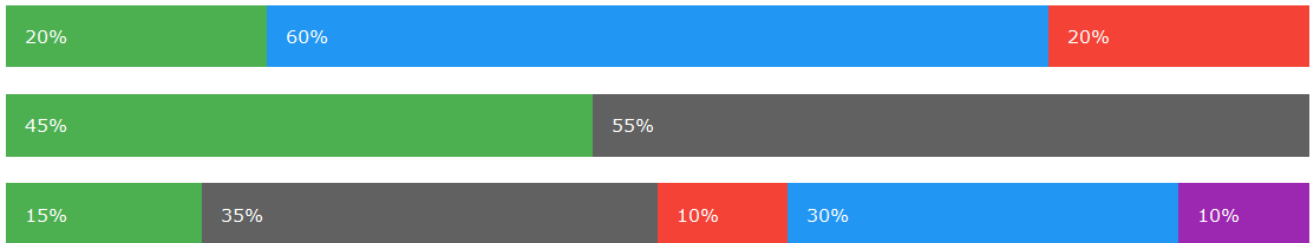
Exemplo 7

Layout



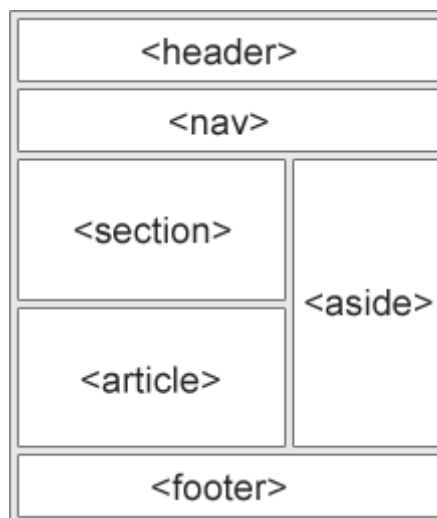


Usando “div” e percentuais no tamanho



Exemplo 8

Layout em HTML5





Entre as várias mudanças que o HTML 5 traz para o desenvolvimento web, existem algumas *novastags* para o código-fonte. Não são apenas novas regras para *tags* antigas, mas *tags* completamente novas que não existiam antes no HTML 4.01.

Entre estas novidades estão as *tags* chamadas de *tags* de layout. Elas servem para classificar seções de sua página Web como áreas de significado sem que você precise de uma `<div>` para defini-las. O cabeçalho da página pode ser encapsulado dentro da *tag* `<header>` (não confundir com `<head>...`) e sua navegação pode ser encapsulada pela *tag* `<nav>`, por exemplo.

Você provavelmente usará estas *tags* da mesma forma que já está usando uma `<div>` para estas áreas, com a principal diferença de que agora elas estão definidas para você e seu navegador, de forma que você não precisa mais utilizar um `<id>` para identificá-las e sistemas automatizados serão capazes de entender o seu contexto.

Mas existem outras diferenças e vamos dar uma olhada em cada uma delas agora.

ARTICLE

<article> é um indicador de uma área de conteúdo do seu site, o núcleo de uma determinada página, a porção mais central e relevante da página.

Antes da sua existência, mecanismos de busca, como o Google não tinham como identificar a parte mais relevante de seu conteúdo. Tanto a navegação, quanto o topo da página ou o seu miolo tinham o mesmo peso, sendo analisados de baixo para cima e sua importância determinada pela ordem de aparição no código-fonte. Com esta *tag*, navegadores, buscadores e outros sistemas automatizados podem identificar facilmente o conteúdo principal.

De acordo com a especificação semântica, o elemento contido dentro de **<article>** está pronto para ser distribuído ou reutilizado de forma independente, sem precisar dos demais elementos da página para ser compreendido. Pode abranger uma postagem, um artigo, um comentário de usuário, um *gadget* etc.

ASIDE

<aside> é um indicador de uma área de conteúdo paralelo, complementar ao conteúdo principal em sua página.

Existe uma interpretação equivocada das especificações que indica seu uso para “sidebars”. Não é o caso, o conteúdo contido nesta *tag* é “lateral” no sentido figurado, não no sentido estrito da palavra, uma vez que, assim como as outras *tags* de layout, ela não guarda em si nenhuma instrução de posicionamento para o navegador.

O elemento contido dentro de **<aside>** funciona como uma nota, um adendo, um esclarecimento ao conteúdo que a cerca. Sua formatação fica ao critério do *designer* responsável e do CSS implementado.



FIGURE

<figure> é um indicador de uma área de conteúdo gráfico dentro da página, seja uma foto, uma ilustração, um gráfico, um diagrama. A *tag* também pode ser utilizada para especificar fragmentos de código-fonte.

Na prática, funciona como um *container* para `` e `<code>`. A *tag* introduz um candidato a substituto para o antigo `<legend>`: o **<figcaption>**.

A função do **<figcaption>** é marcar um cabeçalho ou legenda para um ou mais elementos inseridos dentro de **<figure>**.

FOOTER

<footer> é um indicador de uma área de conteúdo que vem ao final de um conteúdo, popularmente chamado de “rodapé”.

Semanticamente, de acordo com as especificações seu uso não se restringe ao rodapé da página, mas pode ser aplicado ao final de uma seção ou área. Dentro deste elemento costumam vir informações como autor, contato, *copyright*, documentos relacionados etc.

HEADER

<header> é um indicador de uma área de conteúdo que vem no começo de um conteúdo, popularmente chamado de “cabeçalho”.

Semanticamente, de acordo com as especificações, seu uso não se restringe ao topo do documento, mas pode ser aplicado ao início de uma seção ou área. Dentro deste elemento costumam vir informações introdutórias e navegação.

Embora a interpretação das *tags* **<header>** e **<footer>** pelos navegadores não atribua qualquer posicionamento visual no layout da página, a regra de bom uso determina que o primeiro apareça antes do conteúdo principal e o segundo venha após o conteúdo principal.

NAV

<nav> é um indicador de uma área na página que contém majoritariamente links de navegação.

Nem todos os links de um documento precisam estar contidos dentro deste elemento, apenas aquelas áreas popularmente conhecidas como “menu” ou listas de links.

Semanticamente, sistemas automatizados como navegadores, mecanismos de busca ou leitores de tela não tinham como diferenciar links de navegação de conteúdo propriamente dito. Com a implementação da *tag* **<nav>**, por exemplo, leitores de tela podem pular inicialmente este tipo de área no site para usuários com deficiências.



SECTION

<section> é um indicador de seção, capítulo, divisão de um determinado capítulo, cujo conteúdo está agrupado pelo mesmo tema.

A princípio, esta é a *tag* que causa mais confusão entre os usuários. Mas não deveria. Semanticamente, ela define uma seção genérica do conteúdo, como o próprio nome sugere. Se **<article>** é o grande agrupador de conteúdo, **<section>** são os lugares onde antes você aplicaria um **<h2>**, **<h3>**, **<h4>** etc.

Somando-se a introdução e as definições de cada *tag* de layout do HTML 5, este artigo que você está lendo, por exemplo, poderia ser dividido facilmente em 8 **<section>** diferentes.

Entretanto, não se deve confundir **<section>** com **<div>**. O primeiro agrupa conteúdo que compartilha o mesmo tema. O segundo representa uma ruptura genérica no fluxo da página. Em bom português, toda seção é uma divisão do conteúdo, mas nem toda divisão chega a formar uma seção.

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="utf-8">
  <title>Aula 05 e 06 - HTML - Exemplo 8</title>
</head>
<body>

</body>
</html>
```

<!--

As tags abaixo são utilizadas nos documentos em HTML5, e têm função estrutural no seu código. Portanto, essas tags têm grande importância na questão semântica da sua página, saber utilizá-

las pode adequar o seu código para uma melhor visualização por parte do navegador e do usuário, além de proporcionar uma otimização para os processos de SEO.

<header></header> -

Essas tags definem um cabeçalho. Portanto, todo conteúdo que estiver dentro dela faz



parte de um cabeçalho, podendo ser utilizado dentro de outras sessões. Não confundir com as tags <head>;

<main></main> -

Essas tags representam o conteúdo principal do seu corpo, ou seja, o conteúdo relacionado diretamente com o tópico central da página ou com a funcionalidade central da aplicação;

<footer></footer> -

Essas tags definem um rodapé para a página, geralmente utilizadas no final da página;

<section></section> - Essas tags definem uma sessão para sua página;

<article></article> -

Essas tags definem um artigo da sua página. Nesse sentido, são utilizadas para separar o conteúdo da sua página. Muito utilizado principalmente por blogs ou páginas de conteúdo;

<aside></aside> -

Essas tags representam uma seção de uma página cujo conteúdo é tangencialmente relacionado ao conteúdo do seu entorno, que poderia ser considerado separado do conteúdo;

<nav></nav> -

Essa tag define um conteúdo de navegação. Portanto, é muito utilizado em conjunto com listas e na criação de menus;

<div></div> -

Define uma divisão da página. Desta forma, funciona como um container para conteúdo de fluxo. Uma vez que não possui um valor semântico, é muito utilizado para organizar melhor o conteúdo. Anteriormente ao HTML5, era utilizado no lugar das categorias acima.-->



Exemplo 9

```
<!DOCTYPE html>
<html lang="pt-br">

<head>
  <meta charset="utf-8">

  <title>Aula 05 e 06 - HTML - Exemplo 9</title>
</head>

<body style="font-family: Arial, Helvetica, sans-serif;">
  <h2>CSS3 e Layout HTML5</h2>
  <p>Exemplo de layout.</p>

  <header style="background-color: #666; padding: 30px; text-align: center; font-size: 20px; color: white;">
    <h6>Aqui coloco uma logo ou algo que define o cabeçalho da página</h6>
  </header>

  <section style="display: -webkit-flex; display: flex;">
    <nav style="width: 200px; background: rgb(77, 103, 104);">
      <p style="text-align: left;">MENU para outras páginas</p>
      <ul>
        <li><a href="#">Página 1</a></li>
        <li><a href="#">página 2</a></li>
        <li><a href="#">página 3</a></li>
        <li><a href="#">página 4</a></li>
        <li><a href="#">página 5</a></li>
      </ul>
    </nav>

    <article style="background: purple; width: 100%;">
      <h1>Define o corpo principal da página</h1>
      <p>Geralemente onde colocamos o conteúdo principal, com fotos, artigos, vídeos e etc</p>
    </article>
  </section>

  <footer style="background: rgb(103, 145, 103); padding: 10px; text-align: center;">
    <p>Aqui colocamos um rodapé da página. ideal para colocar contatos, email, telefones e etc</p>
  </footer>
```




```
</body>
```

```
</html>
```

```
<!--
```

Margin

A propriedade margin simplesmente adiciona uma margem ao seu elemento. Você pode utilizar qualquer medida Css (px, pt, em, %...) como tamanho da propriedade margin, além disso você pode atribuir valores negativos, mas tenha cuidado com eles.

Ela deve ser usada da seguinte maneira:

```
#elemento {  
    margin-top: 15px;  
    margin-right: 10px;  
    margin-bottom: 25px;  
    margin-left: 35px;  
}
```

outra forma mais completa é/;

```
#elemento { margin: 15px 10px 25px 35px; }
```

o primeiro valor é responsável pela margem do topo, a segunda pela direita, a terceira pela da base e a quarta pela esquerda. Ex: #elemento { margin: top right bottom left };

E se todos os valores forem iguais você pode declarar da seguinte maneira:

```
#elemento { margin: 10px; }
```

A propriedade margin adiciona um espaço transparente e ele não pode ser preenchido com nenhuma cor. Além disso, ela não pode receber bordas.

Padding

O padding tem um funcionamento muito similar ao do margin, porém ao invés de dar um espaçamento externo, ele dá um interno. Vou usar uma imagem para ficar mais claro:

```
#elemento { background: #f90; width: 200px; height: 200px;  
margin: 25px; padding: 25px; border: 1px solid #fff; }
```



Assim como o margin podemos escrever de diversas formas o padding:

```
#elemento {  
    padding-top: 25px;  
    padding-right: 45px;  
    padding-bottom: 35px;  
    padding-left: 15px;  
}  
#elemento { padding: 25px 15px; } /* Top/bottom - right/left */  
#elemento { padding: 25px; } /* top/right/bottom/left */
```

obs:

tenha cuidado ao utilizar o padding pois ele aumenta a largura do elemento. Caso seu elemento tenha um width de 200px e você utilize um padding-left de 50px ele vai passar a ter uma largura de 250px. Então, sempre que necessário, compense o padding do elemento reduzindo o width ou o height.

-->



Exemplo 10

```
<!DOCTYPE html>
<html lang="pt-br">

<head>
  <meta charset="utf-8">
  <title>Aula 05 e 06 - HTML - Exemplo 10</title>
  <style>
    body {
      font-family: Arial, Helvetica, sans-serif;
    }

    header {
      background-color: #666;
      padding: 30px;
      text-align: center;
      font-size: 20px;
      color: white;
    }

    section {
      display: -webkit-flex;
      display: flex;
    }

    nav {
      width: 200px;
      background: rgb(77, 103, 104);
    }

    footer {
      background: rgb(103, 145, 103);
      padding: 10px;
      text-align: center;
    }
  </style>
</head>

<body>
  <h2>CSS3 e Layout HTML5</h2>
  <p>Exemplo de layout.</p>

  <header>
```



```
<h6>Aqui coloco uma logo ou algo que define o cabeçalho da página</h6>
</header>

<section>
  <nav>
    <p style="text-align: left;">MENU para outras páginas</p>
    <ul>
      <li><a href="#">Página 1</a></li>
      <li><a href="#">página 2</a></li>
      <li><a href="#">página 3</a></li>
      <li><a href="#">página 4</a></li>
      <li><a href="#">página 5</a></li>
    </ul>
  </nav>

  <article style="background: purple; width: 100%;">
    <h1>Define o corpo principal da página</h1>
    <p>Geralemente onde colocamos o conteúdo principal, com fotos, artigos, v
ideos e etc</p>
  </article>
</section>

<footer>
  <p>Aqui colocamos um rodapé da pagina. ideal para colocar contatos, email tel
efones e etc</p>
</footer>

</body>

</html>

<!--
  https://www.w3schools.com/html/tryit.asp?filename=tryhtml_layout_flexbox
```

Margin

A propriedade margin simplesmente adiciona uma margem ao seu elemento. Você pode utilizar qualquer medida Css (px, pt, em, %...) como tamanho da propriedade margin, além disso você pode atribuir valores negativos, mas tenha cuidado com eles.

Ela deve ser usada da seguinte maneira:



```
#elemento {  
    margin-top: 15px;  
    margin-right: 10px;  
    margin-bottom: 25px;  
    margin-left: 35px;  
}
```

outra forma mais completa é/;

```
#elemento { margin: 15px 10px 25px 35px; }
```

o primeiro valor é responsável pela margem do topo, a segunda pela direita, a terceira pela base e a quarta pela esquerda. Ex: #elemento { margin: top right bottom left };;

E se todos os valores forem iguais você pode declarar da seguinte maneira:

```
#elemento { margin: 10px; }
```

A propriedade margin adiciona um espaço transparente e ele não pode ser preenchido com nenhuma cor. Além disso, ela não pode receber bordas.

Padding

O padding tem um funcionamento muito similar ao do margin, porém ao invés de dar um espaçamento externo, ele dá um interno. Vou usar uma imagem para ficar mais claro:

```
#elemento { background: #f90; width: 200px; height: 200px;  
margin: 25px; padding: 25px; border: 1px solid #fff; }
```

Assim como o margin podemos escrever de diversas formas o padding:

```
#elemento {  
    padding-top: 25px;  
    padding-right: 45px;  
    padding-bottom: 35px;  
    padding-left: 15px;  
}  
#elemento { padding: 25px 15px; } /* Top/bottom - right/left */  
#elemento { padding: 25px; } /* top/right/bottom/left */
```

obs:



```
enha cuidado ao utilizar o padding pois ele aumenta a largura do elemento. Caso seu elemento tenha um width de 200px e você utilize um padding-left de 50px ele vai passar a ter uma largura de 250px. Então, sempre que necessário, compense o padding do elemento reduzindo o width ou o height.
```

CSS

Quando falamos em *desenvolvimento de sites*, chamamos *tipo de CSS* a forma com que o *documento HTML* irá usar os *estilos CSS*. Existem *três tipos de folha de estilo CSS* possíveis para utilização em *projetos WEB*: *estilos inline*, *interno(incorporado)* e *externo*.

Estilo inline

Estilo inline é quando os *estilos CSS* são aplicados diretamente dentro da *tag de abertura de um elemento HTML*. No entanto, este *tipo de uso das CSS* não é recomendado e encontra-se em desuso pelos *desenvolvedores WEB*. Veja a seguir um exemplo de utilização deste *tipo de estilo*:

```
<div style="background-color:blue; color: red; width:300px; height:200px;">
```

estilo incorporado

O *estilo incorporado* também não é o mais indicado e recomendado, uma vez que estilos escritos desta forma ficam restritos a um único documento, não sendo possível a sua reutilização em outras *páginas do projeto WEB*.

Neste método, você declara a folha de estilo diretamente dentro da *seção head do documento HTML*, utilizando o *elemento style*. Veja um exemplo a seguir da utilização deste *tipo de estilo*:

```
<html>
  <head>
    <style>
      h2 {
        color:#000;
      }
    </style>
  </head>
```



```
<body>
    <h2>Título Secundário da Página</h2>
</body>
</html>
```



Estilo externo

Este *tipo de uso de CSS* em documentos HTML é o *mais indicado e recomendado*. Neste tipo, os *códigos CSS* ficam em um *documento externo ao documento HTML*, separando assim as camadas de *estrutura e apresentação do projeto WEB*.

Existem duas formas de incorporar folhas de estilos CSS externa ao documento WEB.:

Linkados e Importados

Linkados

Para o *tipo de estilo CSS externo linkado*, usa-se o elemento link para *incorporar o documento CSS externo ao documento HTML*. O elemento link deve ser usado dentro da seção head do documento HTML.

Veja a seguir um exemplo da *utilização do elemento link* para incorporar uma folha de estilo CSS ao documento HTML. Neste exemplo, imagine que o documento CSS esteja dentro de uma pasta chamada “css” e o seu arquivo HTML, “index.html” esteja na raiz, como pode ser visto na ilustração abaixo:

```
<head>  
    <link rel="stylesheet" type="text/css" media="screen" href="css/estilos.css" />  
</head>
```

Partindo do princípio que o documento CSS esteja nomeado como “estilos.css” e que esteja dentro de uma pasta chamada “css”, o *atributo href* deve ficar exatamente com



esta informação: “*css/estilos.css*”, que é a forma com que o navegador entenderá onde se encontra o *arquivo CSS a ser linkado*.

Importados

O *tipo de estilo das CSS importadas* funciona igualmente ao tipo de estilo das CSS linkadas. O que difere aqui é somente a forma com que isto é feito, uma vez que, neste caso, é usada a *diretiva @import* dentro do elemento *style*, na *seção head do documento HTML*, como pode ser visto no exemplo abaixo:

```
<head>
  <style type="text/css">
    @import url("css/estilos.css") screen, projection;
  </style>
</head>
```

Se o desenvolvedor pretende ganhar velocidade, não é recomendável o uso do estilo importadas, pois o *@import* demora mais para carregar a folha de estilo.

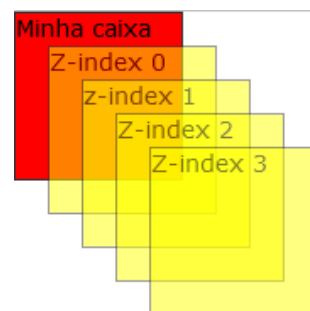


Propriedade z-index

Quando pensamos em posicionamento de elementos em páginas web, a primeira coisa que vem em nossa cabeça são os eixos X e Y que são os eixos horizontais e verticais ou largura e altura, respectivamente.

Todo elemento que tentamos posicionar na página web vamos movendo usando esses dois eixos e assim conseguimos colocar ele onde queremos.

A propriedade z-index trabalha com um eixo que não é muito conhecido e tampouco usado pela maioria dos desenvolvedores, o eixo Z. O eixo Z é o eixo responsável pelo cálculo e posicionamento da profundidade de algum elemento, ou seja, é aquele que irá determinar se o elemento estará mais próximo ou mais afastado da tela.



A propriedade `z-index` só funciona em elementos posicionados (posição: absoluta, posição: relativa, posição: fixa).

Quando criamos elementos com posicionamentos, automaticamente a propriedade z-index “recebe” um valor, conforme são listados no html. Quanto maior for o número dado ao z-index, mais “acima” das outras camadas de posicionamento do no html ele estará.

A propriedade z-index poderá receber os seguintes valores:

- auto

O elemento é desenhado na mesma ordem-z como um elemento com `z-index: 0`. Isto *não* cria uma nova pilha de contexto.

- <inteiro (0, 1, 2, 3,...)>

O elemento é desenhado na ordem-z dada. Isto cria também uma nova pilha de contexto, que significa que todos os descendentes também terão o desenho no mesmo z-index. Dessa forma os z-indexes dos descendentes não são comparados aos z-indexes dos elementos do lado de fora deste elemento.