

CSS



CSS

Seletor do tipo "classe" com arquivo externo

Também é possível criar um arquivo **.css** e criar uma ou mais classes ao invés de colocar a formatação dentro do documento HTML.

Para acessar as classes basta no arquivo HTML fazer referência ao link do arquivo **.css**.

A classe que foi criada no arquivo **.css** tem o nome de **.bloco**

Para criar a classe no arquivo **.css** é preciso que haja o “**ponto**” “.” antes do nome da classe. A mesma deve estar entre a **abertura** e o **fechamento das chaves** “**{}**”.

CSS

Classe criada no arquivo .css (.bloco)

Para criar a classe no arquivo .css é preciso que haja o “ponto” “.” antes do nome da classe. A mesma deve estar entre a abertura e o fechamento das chaves “{}”.

Este é o arquivo **estilo.css** criado para integrar com o outro arquivo html

```
.bloco { color: blue;  
        background-color: yellow;  
        text-align: center;  
    }
```



características do bloco

CSS

No arquivo HTML para que a classe seja utilizada em um determinado ponto, é preciso utilizar a sessão `<section>`
`</section>` com a especificação do atributo `class` e o `nome da classe`.

A tag `<link>` define a relação entre o documento atual e um recurso externo. É mais frequentemente usada para vincular a folhas de estilo externas. É um elemento vazio, ele contém apenas atributos.

O atributo `rel` especifica a relação entre o documento atual e o link do outro documento.

CSS

Exemplo:

```
<!DOCTYPE HTML>
```

```
<html>
```

```
  <head>
```

```
    <title>Uso CSS externo</title>
```

```
  </head>
```

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

```
  <section class="bloco">
```

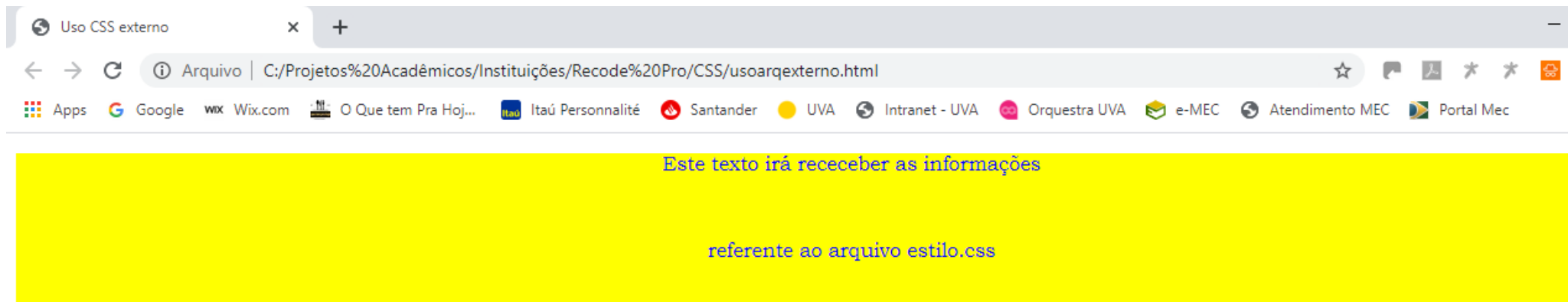
```
    <p>Este texto irá receber as informações</p><br>
```

```
    <p>referente ao arquivo estilo.css</p><br>
```

```
  </section>
```

```
</html>
```

CSS



CSS

Em qual momento devemos utilizar os seletores vistos?

1. Definir que todos os elementos da página terão o mesmo tamanho, é possível utilizar o seletor do tipo do **elemento** como algo mais genérico.
2. Se você pretende destacar apenas um único elemento específico, que não se repete na página, utilize o **id** no elemento e no seletor CSS.
3. Usar as **classes** em elementos que precisam se repetir algumas vezes em uma página, ou quando os elementos precisam acumular estilos de classes diferentes.

Atividade Prática 1

Duração de 25 minutos

1. Criar um arquivo de nome `formatos.css`
 - 1.1 Criar uma formatação de fundo de tela, cor da fonte, tipo de fonte e centralização do texto;
2. Criar um arquivo html de `exercclass.html`;
3. Colocar um título como uso tag `<title>`;
4. Fazer o link para que os parágrafos tenham a formatação do arquivo externo `formato.css`;
5. Criar três parágrafos com texto livre;
6. Trabalhar com formatações que você crie;

CSS

Trabalhando com as larguras e alturas



CSS

Definindo altura e largura

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Altura e Largura</title>
  </head>
  <style>
    body {background-color: silver}
    textarea {width: 200px; height: 200px}
  </style>
  <body>
    <form>
      <textarea>Podemos dizer que a vida possui vários significados</textarea>
    </form>
  </body>
</html>
```

CSS

Definindo altura e largura com três textarea

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Altura e Largura</title>
  </head>
  <style>
    body {background-color: silver}
    h1 {color: crimson}
    textarea {width: 200px; height: 200px}
  </style>
  <body align="center">
    <h1>Emoções que Precisam ser Vistas</h1>
    <form>
      <textarea>Podemos dizer que a vida possui vários significados</textarea>
      <textarea>Por isso, é importante entendermos determinados avisos</textarea>
      <textarea>Dessa forma, nunca seremos pegos de surpresa</textarea>
    </form>
  </body>
</html>
```

CSS

Temos a página com três textarea com o mesmo tamanho.

Emoções que Precisam ser Vistas

Podemos dizer que a vida
possui vários significados

Por isso, é importante
entendermos determinados
avisos

Dessa forma, nunca seremos
pegos de surpresa

Atividade Prática 2

Duração de 25 minutos

1. Criar um arquivo html de textarea.html;
2. Colocar um título como uso tag <title>;
3. Criar uma tag <form> para trabalhar com o textarea;
4. Criar textarea com texto livre;
5. Trabalhar com formatações que você crie;

CSS

Definindo altura e largura com dois formulários

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Uso da Classe</title>
  </head>
  <style>
    body {background-color: silver}
    h1 {color: darkorange}
    textarea {width: 200px; height: 200px}
  </style>
  <body align="center">
    <h1>Emoções que Precisam ser Vistas</h1>
    <form>
      <textarea>Podemos dizer que a vida possui vários significados</textarea>
      <textarea>Por isso, é importante entendermos determinados avisos</textarea>
      <textarea>Dessa forma, nunca seremos pegos de surpresa</textarea>
    </form>
    <form>
      <textarea>Podemos dizer que a vida possui vários significados</textarea>
      <textarea>Por isso, é importante entendermos determinados avisos</textarea>
      <textarea>Dessa forma, nunca seremos pegos de surpresa</textarea>
    </form>
  </body>
</html>
```

Emoções que Precisam ser Vistas

Podemos dizer que a vida possui vários significados

Por isso, é importante entendermos determinados avisos

Dessa forma, nunca seremos pegos de surpresa

Podemos dizer que a vida possui vários significados

Por isso, é importante entendermos determinados avisos

Dessa forma, nunca seremos pegos de surpresa

CSS

Trabalhando com as bordas

Estrutura de espaçamento no CSS é a propriedade **border**, que define as bordas do elemento.

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Bordas</title>
  </head>
  <style>
    body {background-color: silver}
    h1 {color: green}
    textarea {width: 200px; height: 200px; border: medium double}
  </style>
  <body align="center">
    <h1>Emoções que Precisam ser Vistas</h1>
    <form>
      <textarea>Podemos dizer que a vida possui vários significados</textarea>
      <textarea>Por isso, é importante entendermos determinados avisos</textarea>
      <textarea>Dessa forma, nunca seremos pegos de surpresa</textarea>
    </form>
  </body>
</html>
```


Emoções que Precisam ser Vistas

Podemos dizer que a vida
possui vários significados

Por isso, é importante
entendermos determinados
avisos

Dessa forma, nunca seremos
pegos de surpresa

Atividade Prática 3

Duração de 25 minutos

1. Criar um arquivo html de textarea_l.html;
2. Colocar um título como uso tag <title>;
3. Criar duas tags <form> para trabalhar com o textarea;
4. Criar textarea com texto livre para um <form>;
5. Criar textarea com texto livre para outro <form>;
6. Fazer aplicação de bordas estilizadas;
7. Trabalhar com formatações que você crie;

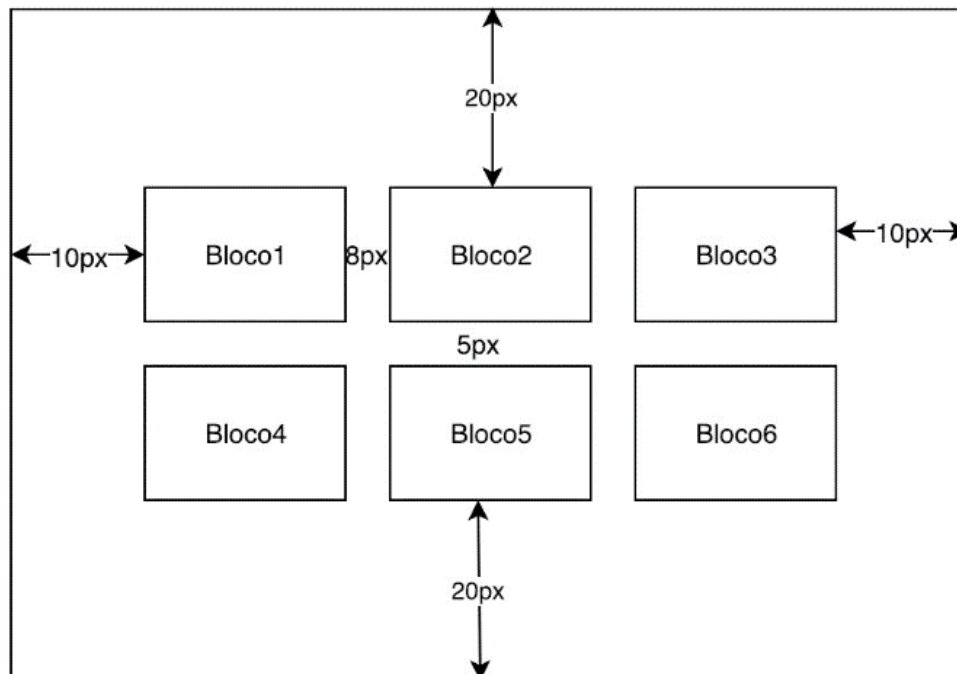
CSS

Trabalhando com Margin e padding

Colocando espaçamentos em seus elementos.

Existem dois tipos de espaçamentos:

- O que separa um bloco do outro, conhecido como margem; e
- O responsável por dar espaço entre o conteúdo de um bloco e as suas bordas, o espaçamento interno.



CSS

Espaçamento Interno com Padding

No espaçamento interno conseguimos incluir uma distância entre o conteúdo criado e as bordas de um bloco (bordas). Para definir esses espaçamentos, utilizamos a propriedade **padding**.

Ela, assim como as bordas, recebe valores numéricos em porcentagem e pixels, além de podermos incluir mais de uma medida (até 4, uma de cada lado) e trabalhar com cada lado separadamente, que são:

- top
- bottom
- right
- left

CSS

Sintaxe:

A propriedade padding pode ser usada usando um, dois, três ou todos os quatro valores. Cada valor é um <comprimento> ou uma <porcentagem>.

Quando um único é valor utilizado, ele se aplica a todos os quatro lados .

Se Dois valores são utilizados no primeiro se aplica às bordas verticais (superior e inferior) e o segundo se aplica às bordas horizontais (esquerda e direita).

Quando três valores são usados o primeiro se aplica à borda superior , o segundo será usado tanto para a borda esquerda quanto para a borda direita. O terceiro será selecionado à borda inferior .

Com os Quatro valores a ordem de aplicação é: superior, direita, inferior e esquerda (de acordo com o movimento dos ponteiros do relógio).

CSS

Exemplo:

padding: 5%; /* padding 5% em todos os lados */

padding: 10px; /* padding 10px em todos os lados */

padding: 10px 20px; /* 10px acima e abaixo */
 /* 20px esquerda e direita */

padding: 10px 3% 20px; /* 10px acima */
 /* 3% esquerda e direita */
 /* 20px abaixo */

padding: 1em 3px 30px 5px; /* acima 1em padding */
 /* direita 3px padding */
 /* abaixo 30px padding */
 /* esquerda 5px padding */

CSS

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Padding</title>
  </head>
  <style>
    body {background-color: silver}
    h1 {color: green}
    textarea {width: 200px; height: 200px; border: medium double}
    .padding {width: 200px; height: 200px; border: medium double; padding: 50px}
  </style>
  <body>
    <h1>Emoções que Precisam ser Vistas</h1>
    <form>
      <textarea>Podemos dizer que a vida possui vários significados</textarea>
    </form>
    <form class="padding">
      <textarea>Por isso, é importante entendermos determinados avisos</textarea>
    </form>
  </body>
</html>
```

Emoções que Precisam ser Vistas

Podemos dizer
que a vida
possui vários
significados

Por isso, é
importante
entendermos
determinados
avisos

Atividade Prática 5

Duração de 25 minutos

1. Criar um arquivo html de padding.html;
2. Colocar um título como uso tag <title>;
3. Criar duas tags <form> para trabalhar com o textarea;
4. Criar textarea com texto livre para um <form>;
5. Criar textarea com texto livre para outro <form> para trabalhar com o padding;
6. Fazer aplicação de bordas estilizadas;
7. Trabalhar com formatações que você crie;

CSS

Margin

Ela define a distância da borda até o início do objeto em questão. Não tem nenhum segredo, as **regras aprendidas em bordas e padding são as mesmas** para essa propriedade.

Para definir as margens, usamos a propriedade **margin**. Você vai notar que, mesmo sem definir **margin** e **padding** no seu elemento, o HTML já tem uma medida padrão definida.

Então, não se preocupe, você pode “zerar”, ou seja, definir as suas próprias medidas.

CSS

A margem segue o **mesmo padrão de bordas** e **padding**, podendo receber valores numéricos em porcentagem e pixels, além de permitir incluir mais de uma medida (até 4, uma para cada lado) e trabalhar com cada lado separadamente.

Mas, lembrando:

- top
- bottom
- right
- left

CSS

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Margin</title>
  </head>
  <style>
    body {background-color: silver}
    h1 {color: green}
    textarea {width: 100px; height: 100px;}
    .margin {width: 100px; height: 100px; margin: 50px}
  </style>
  <body>
    <h1>Emoções que Precisam ser Vistas</h1>
    <form>
      <textarea>Podemos dizer que a vida possui vários significados</textarea>
    </form>
    <form class="margin">
      <textarea>Por isso, é importante entendermos determinados avisos</textarea>
    </form>
  </body>
</html>
```

Emoções que Precisam ser Vistas

Podemos dizer
que a vida
possui vários
significados

Por isso, é
importante
entendermos
determinados
avisos

Atividade Prática 6

Duração de 25 minutos

1. Criar um arquivo html de nome margin.html;
2. Colocar um título como uso tag <title>;
3. Criar duas tags <form> para trabalhar com o textarea;
4. Criar textarea com texto livre para um <form>;
5. Criar textarea com texto livre para outro <form> para trabalhar com a margin;
6. Fazer aplicação de bordas estilizadas;
7. Trabalhar com formatações que você crie;

CSS

Display

A propriedade display permite que você defina a maneira como determinado elemento HTML deve ser exibido, ou não.

Por padrão, os elementos se organizem como blocos. Toda vez que adicionamos um elemento HTML (como h1, p, li, etc), eles se organizam um abaixo do outro.

Essa estrutura de blocos é como estes elementos são exibidos por padrão. Com o CSS, podemos adicionar e modificar todos os estilos existentes, inclusive este.

CSS

Nos permite organizar os elementos com as seguintes opções:

block → A declaração **display: block** define o que já conhecemos, que é a sua opção padrão, ou seja, os elementos do tipo bloco são organizados um abaixo do outro.

inline → A declaração **display: inline** exhibe blocos na mesma linha, fazendo com que ele se adeque ao tamanho do conteúdo que contém.

inline-block → A declaração **display: inline-block** mescla características dos valores inline e block.

none → A declaração **display: none** faz com que o elemento não seja renderizado.

CSS

```
<html>
  <head>
    <style>
      p {color: red;}

      p.ex1 {display: none;}
      p.ex2 {display: inline;}
      p.ex3 {display: block;}
      p.ex4 {display: inline-block;}
    </style>
  </head>
  <body>
    <h1>The display Property</h1>

    <h2>display: none:</h2>
    <div>
      Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam semper diam at erat pulvinar, at
      pulvinar felis blandit. <p class="ex1">HELLO WORLD!</p> Vestibulum.
    </div>

    <h2>display: inline:</h2>
    <div>
      Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam semper diam at erat pulvinar, at
      pulvinar felis blandit. <p class="ex2">HELLO WORLD!</p> Vestibulum..
    </div>
  </body>
</html>
```

CSS

```
<h2>display: block:</h2>
```

```
<div>
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam semper diam at erat pulvinar, at pulvinar felis blandit. <p class="ex3">HELLO WORLD!</p> Vestibulum.

```
</div>
```

```
<h2>display: inline-block:</h2>
```

```
<div>
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam semper diam at erat pulvinar, at pulvinar felis blandit. <p class="ex4">HELLO WORLD!</p> Vestibulum.

```
</div>
```

```
</body>
```

```
</html>
```

CSS

Float

A propriedade **float** faz com que os elementos fiquem flutuando na ordem em que aparecem dentro da sua hierarquia, ou seja, dentro do elemento pai. Os elementos podem flutuar da esquerda para a direita ou da direita para a esquerda.

opções de valores para float:

Left → A declaração **float: left** faz com que o nosso elemento “flutue à esquerda”, logo ele sai do fluxo normal, onde os elementos blocos se posicionam um abaixo do outro e os coloca ao longo do lado direito. Esta propriedade tem um efeito semelhante (mas não igual) à **display: inline** (que encaixa os elementos na mesma linha, enquanto houver espaço).

CSS

Right → A declaração **float: right** faz com que o nosso elemento “flutue à direita”. Este efeito é bem interessante, pois funciona como se o fluxo fosse invertido, iniciando na margem direita em direção à margem esquerda, da mesma forma como os chineses e japoneses escrevem.

CSS

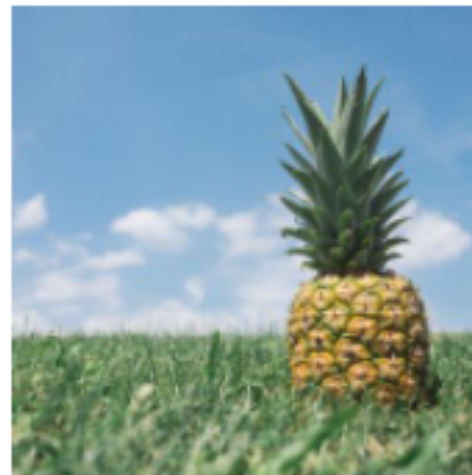
```
<!DOCTYPE html>
<html>
  <head>
    <style>
      img { float: right;}
    </style>
  </head>
  <body>
    <h1>The float Property</h1>
    <p>In this example, the image will float to the right in the text, and the text in the paragraph
    will wrap around the image.</p>

    <p> Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla
    et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor.
    Maecenas nisl est, ultrices nec congue eget, auctor vitae massa. Fusce luctus vestibulum augue
    ut aliquet. Mauris ante ligula, facilisis sed ornare eu, lobortis in odio.
    Praesent convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis dictum nisi,
    sed ullamcorper ipsum dignissim ac. In at libero sed nunc venenatis imperdiet sed ornare
    turpis. Donec vitae dui eget tellus gravida venenatis. Integer fringilla congue eros non
    fermentum. Sed dapibus pulvinar nibh tempor porta. Cras ac leo purus. Mauris quis diam
    velit.</p>
  </body>
</html>
```

The float Property

In this example, the image will float to the right in the text, and the text in the paragraph will wrap around the image.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec congue eget, auctor vitae massa. Fusce luctus vestibulum augue ut aliquet. Mauris ante ligula, facilisis sed ornare eu, lobortis in odio. Praesent convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis dictum nisi, sed ullamcorper ipsum dignissim ac. In at libero sed nunc venenatis imperdiet sed ornare turpis. Donec vitae dui eget tellus gravida venenatis. Integer fringilla congue eros non fermentum. Sed dapibus pulvinar nibh tempor porta. Cras ac leo purus. Mauris quis diam velit.



CSS

Div

A tag `<div>` define uma divisão ou seção em um documento HTML.

É usada como um contêiner para elementos HTML que são estilizados com CSS ou manipulados com JavaScript.

É facilmente estilizada usando o atributo class ou id.

Qualquer tipo de conteúdo pode ser colocado dentro da `<div>` tag!

CSS

Exemplo:

```
<html>
  <head>
    <style>
      .Div {
        border: 5px outset red;
        background-color: lightblue;
        text-align: center;
      }
    </style>
  </head>
  <body>
    <h1>Tratando o Elemento fora da tag Div</h1>

    <div class="Div">
      <h2>Elemento dentro da tag Div</h2>
      <p>fazendo o Uso Devido</p>
    </div>
    <p>Uso fora da tag Div</p>
  </body>
</html>
```


CSS

Tratando o Elemento fora da tag Div

Elemento dentro da tag Div

fazendo o Uso Devido

Uso fora da tag Div

CSS

Exemplo:

```
<html>
<head>
  <style>
    .Div {
      border: 5px outset red;
      background-color: lightblue;
      text-align: center;
    }
    .Div1 {
      border: 10px outset green;
      background-color: yellow;
      text-align: center;
    }
  </style>
</head>
<body>
  <div class="Div">
    <h2>Elemento dentro da tag Div</h2>
    <p>Fazendo o Uso Devido</p>
  </div>

  <div class="Div1">
    <h2>Outro Elemento dentro da tag Div</h2>
    <p>Fazendo Mais um Uso Devido</p>
  </div>
</body>
</html>
```

CSS

Elemento dentro da tag Div

Fazendo o Uso Devido

Outro Elemento dentro da tag Div

Fazendo Mais um Uso Devido

CSS

Position

A propriedade position especifica o tipo de método de posicionamento usado para um elemento.

Existem cinco valores de posição diferentes:

- static
- Relative
- fixed
- absolute
- sticky

Os elementos são posicionados usando as propriedades superior, inferior, esquerda e direita. No entanto, essas propriedades não funcionarão a menos que a position propriedade seja definida primeiro. Eles também funcionam de forma diferente dependendo do valor da posição.

CSS

Relative → A declaração **position: relative** pode ser entendida através do seu nome, ou seja, é uma posição que possui relação com algo, e nesse caso, com o elemento pai. Esta é a opção padrão dos elementos. No módulo anterior aprendemos sobre margens (**margin**) que define a distância dos objetos das suas margens (superior, esquerda, inferior e direita) e que elas podem ser definidas separadamente (margin-top, margin-right, margin-bottom e margin-left).

CSS

Exemplo:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      div.relative { position: relative;
                     left: 30px;
                     border: 3px solid #73AD21;}
    </style>
  </head>
  <body>
    <h2>position: relative;</h2>
    <p>An element with position: relative; is positioned relative to its normal
    position:</p>
    <div class="relative"> This div element has position: relative; </div>
  </body>
</html>
```

CSS

Absolut → A declaração **position: absolut** é muito interessante, ela faz com que o elemento ignore a sua localização hierárquica (seus pais) e passa a fazer referência diretamente à página do navegador, como se estivesse flutuando sobre todos os demais elementos do seu documento HTML. Este estilo é muito utilizado e permite informações dinâmicas, um exemplo disso são as mensagens flutuantes que aparecem no site.

CSS

Exemplo:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      div.relative {position: relative;
        width: 400px;
        height: 200px;
        border: 3px solid #73AD21;}

      div.absolute {position: absolute;
        top: 80px;
        right: 0;
        width: 200px;
        height: 100px;
        border: 3px solid #73AD21;}
    </style>
  </head>
  <body>
    <h2>position: absolute;</h2>
    <p>An element with position: absolute; is positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like fixed):</p>
    <div class="relative">This div element has position: relative;
      <div class="absolute">This div element has position: absolute;</div>
    </div>
  </body>
</html>
```


posição: absoluta;

Um elemento com `posição: absoluto;` é posicionado em relação ao ancestral posicionado mais próximo (em vez de posicionado em relação à janela de visualização, como `fixo`):

Este elemento div tem `position: relative;`

Este elemento div tem
`position: absolute;`

CSS

Static → Não é posicionado de nenhuma maneira especial; está sempre posicionado de acordo com o fluxo normal da página.

Exemplo:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      div.static {position: static;
        border: 3px solid #73AD21;}
    </style>
  </head>
  <body>
    <h2>position: static;</h2>
    <p>An element with position: static; is not positioned in any special way; it
    is always positioned according to the normal flow of the page:</p>
    <div class="static">This div element has position: static;</div>
  </body>
</html>
```

CSS

Fixed → Um elemento com `position: fixed;` é posicionado em relação à janela de visualização, o que significa que ele sempre permanece no mesmo lugar, mesmo se a página for rolada. As propriedades superior, direita, inferior e esquerda são usadas para posicionar o elemento.

Um elemento fixo não deixa uma lacuna na página onde normalmente estaria localizado.

CSS

Exemplo:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      div.fixed {position: fixed;
                  bottom: 0;
                  right: 0;
                  width: 300px;
                  border: 3px solid #73AD21;}
    </style>
  </head>
  <body>
    <h2>position: fixed;</h2>
    <p>An element with position: fixed; is positioned relative to the viewport, which
    means it always stays in the same place even if the page is scrolled:</p>
    <div class="fixed">This div element has position: fixed;</div>
  </body>
</html>
```

CSS

Fixed → A declaração **position: fixed** também é muito legal e funciona quase da mesma forma que o **absolut**: nesta opção o elemento fica fixo na página, onde quisermos que ele fique. Você pode rolar a página para cima e para baixo, mas ele fica sempre no mesmo local que foi definido, em relação à página. Essa posição é muitas vezes utilizada para menus.

CSS – Material Complementar

Tabelas

Para especificar as bordas da tabela em CSS, use a propriedade **border**.

O exemplo abaixo especifica uma borda preta para os elementos `<table>`, `<th>` e `<td>`

```
table, th, td {border: 1px solid black;}
```

Primeiro nome	Último nome
Pedro	Griffin
Lois	Griffin

CSS

Recolher bordas da mesa

A propriedade border-collapse define se as bordas da tabela devem ser reduzidas em uma única borda.

```
table {border-collapse: collapse;}  
table, th, td {border: 1px solid black;}
```

Primeiro nome	Último nome
Pedro	Griffin
Lois	Griffin

CSS

Largura e altura da mesa

A largura e a altura de uma tabela são definidas pelas propriedades **width** e **height**.

O exemplo abaixo define a largura da tabela em 100% e a altura dos elementos <th> em 50px.

```
table {width: 100%;}  
th {height: 50px;}
```

Primeiro nome	Último nome	Poupança
Pedro	Griffin	\$ 100
Lois	Griffin	\$ 150
Joe	Swanson	\$ 300

CSS

Alinhamento horizontal

A propriedade **text-align** define o alinhamento horizontal (como esquerda, direita ou centro) do conteúdo em `<th>` ou `<td>`.

Por padrão, o conteúdo dos elementos `<th>` é alinhado ao centro e o conteúdo dos elementos `<td>` é alinhado à esquerda.

O exemplo a seguir alinha à esquerda o texto em elementos `<th>`.

th {text-align: left;}

Primeiro nome	Último nome	Poupança
Pedro	Griffin	\$ 100
Lois	Griffin	\$ 150
Joe	Swanson	\$ 300

CSS

Alinhamento vertical

A propriedade **vertical-align** define o alinhamento vertical (como superior, inferior ou meio) do conteúdo em <th> ou <td>.

Por padrão, o alinhamento vertical do conteúdo em uma tabela é médio (para os elementos <th> e <td>).

O exemplo a seguir define o alinhamento vertical do texto para baixo para elementos <td>.

td {height: 50px; vertical-align: bottom;}

Primeiro nome	Último nome	Poupança
Pedro	Griffin	\$ 100
Lois	Griffin	\$ 150
Joe	Swanson	\$ 300

CSS

Tabelas Listradas

Para tabelas listradas de zebra, use o **nth-child()** seletor e adicione um **background-color** a todas as linhas da tabela par (ou ímpar).

Exemplo:

```
tr:nth-child(even) {background-color: #f2f2f2;}
```

Primeiro nome	Último nome	Poupança
Pedro	Griffin	\$ 100
Lois	Griffin	\$ 150
Joe	Swanson	\$ 300

CSS

Cor da Tabela

Especifica através da propriedade `background-color` a cor de fundo e a cor do texto dos elementos `<th>`.

Exemplo:

```
th {background-color: #4CAF50; color: white;}
```

Primeiro nome	Último nome	Poupança
Pedro	Griffin	\$ 100
Lois	Griffin	\$ 150
Joe	Swanson	\$ 300

CSS

Tabela Responsiva

Uma tabela responsiva exibirá uma barra de rolagem horizontal se a tela for muito pequena para exibir o conteúdo completo.

Adicione um elemento de contêiner (como `<div>`) ao `overflow-x:auto` redor do elemento `<table>` para torná-lo responsivo:

Exemplo:

```
<div style="overflow-x:auto;">  
  <table>  
    ... table content ...  
  </table>  
</div>
```

CSS

Exemplo de uso:

```
<style>
```

```
    .vermelho{
```

```
        background-color: red;
```

```
    }
```

```
    .amarelo{
```

```
        background-color: yellow;
```

```
    }
```

```
    .azul{
```

```
        background-color: blue;
```

```
    }
```

```
</style>
```

```
<table border="1">
```

```
    <colgroup>
```

```
        <col class="vermelho">
```

```
        <col class="amarelo">
```

```
        <col class="azul">
```

```
    </colgroup>
```

```
    <tr>
```

```
        <th>Nome</th>
```

```
        <th>Idade</th>
```

```
        <th>Profissao</th>
```

```
</tr>
```

CSS

```
<tr>  
  <td>Ted</td>  
  <td>22</td>  
  <td>Estudante</td>  
</tr>  
<tr>  
  <td>Junior</td>  
  <td>26</td>  
  <td>Escritor</td>  
</tr>  
</table>
```

CSS

tabelacol.html



Arquivo | C:/Projetos%20Acadêmicos/Instituições/Recode%20Pro/CSS/tabelacol.html



Apps



Google

WIX

Wix.com



O Que tem Pra Hoj...



Itaú Personalité



Santander



UVA

Nome	Idade	Profissao
Ted	22	Estudante
Junior	26	Escritor