

HTML - aula 7

Exemplo de formulário

```
<form action="/action_page.php">
  Nome:<br>
  <input type="text" name="nome" value="Fulano"><br>
  Sobrenome:<br>
  <input type="text" name="sobrenome" value="De Tal">
  <br>
  <input type="submit" value="Submit">
</form>
```

Nome:

Sobrenome:

Elemento <form>

O elemento <form> define um formulário que é usado para receber dados do usuário:

```
<form>
```

```
    elementos de formulário
```

```
</form>
```

Elementos de formulários

Existem diferentes tipos de elementos de entrada:

- campos de texto
- caixas de seleção
- botões de rádio
- botões de envio (submit)
- outros

Elementos <input>

O elemento <input> é o elemento de formulário mais importante.

O elemento <input> pode ser mostrado de diversas maneiras, dependendo do tipo do atributo (**type**).

<input type="text">

<input type="radio">

☐

<input type="submit">

Entrada de texto

O campo de entrada de texto é definido da seguinte maneira:

```
<input type="text">
```

Botão de rádio

```
<form action="/action_page.php">  
  <input type="radio" name="sexo" value="m" checked> Masculino<br>  
  <input type="radio" name="sexo" value="f"> Feminino<br>  
</form>
```

☒ Masculino
☐ Feminino

Botão de envio

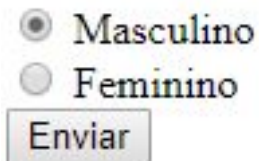
Para definir um botão para enviar os dados do formulário para a página que vai manipular os dados, usamos:

```
<input type="submit">
```


Botão de envio

A página que irá manipular os dados do formulário é aquela especificada no atributo **action** do **form**.

```
<form action="/action_page.php">  
  <input type="radio" name="sexo" value="m" checked> Masculino<br>  
  <input type="radio" name="sexo" value="f"> Feminino<br>  
  <input type="submit">  
</form>
```



☒ Masculino
☐ Feminino

Atributo action

O atributo **action** define a ação que será executada quando o formulário for enviado.

Geralmente, os dados do formulário são enviados para uma página web no servidor quando o usuário apertar o botão **submit**.

Atributo action

No exemplo abaixo, os dados do formulário serão enviados para uma página chamada `"/action_page.php"`. Essa é uma página no servidor que receberá os dados.

```
<form action="/action_page.php">
```

Se o atributo **action** for omitido, a ação será a página atual.

Atributo target

O atributo **target** especifica se o resultado enviado será aberto em uma nova aba do navegador, um **frame** ou na janela atual.

O valor padrão é "**_self**" que significa que o formulário será enviado na janela atual.

Atributo target

Para abrir o resultado em uma nova aba do navegador, use o valor "**_blank**":

Exemplo

```
<form action="/action_page.php" target="_blank">
```

Outros valores possíveis são "_parent" e "_top", ou o nome de um iframe.

Atributo method

O atributo **method** especifica o método HTTP a ser usado para enviar os dados do formulário, as opções são GET ou POST.

The method attribute specifies the HTTP method (GET or POST) to be used when submitting the form data:

Exemplo:

```
<form action="/action_page.php" method="get">
```

Atributo method

Exemplo:

```
<form action="/action_page.php" method="get">
```

ou

```
<form action="/action_page.php" method="post">
```

Quando usar o método GET

O método padrão ao enviar dados de formulário é o GET.

Quando o GET é usado, os dados do formulário enviados podem ser vistos no campo de endereço do navegador.

```
/action_page.php?nome=Fulano&sobrenome=De%20Tal
```


Quando usar o método GET

Notas:

Os dados do formulário são agrupados em pares de nomes/valores.

O comprimento da URL é limitado (cerca de 3000 caracteres).

Nunca use GET para enviar dados importantes já que eles poderão ser vistos na URL.

GET é melhor para enviar dados não seguros.

Quando usar o método POST

Sempre use o método POST se os dados do formulário forem informações pessoais ou sensíveis (senha por exemplo).

O método POST não mostra os dados enviados no campo de endereço do navegador.

Quando usar o método POST

Notas:

POST não possui limitações de tamanho e pode ser usado para enviar grande quantidade de dados.

Atributo name

Cada campo de entrada deve ter um atributo name para ser enviado.

Se o atributo name for omitido, os dados daquele campo não serão enviados.

Atributo name

No exemplo abaixo apenas o sobrenome será enviado.

```
<form action="/action_page.php">
```

```
  Nome:<br>
```

```
  <input type="text" value="Fulano"><br>
```

```
  Sobrenome:<br>
```

```
  <input type="text" name="sobrenome" value="De Tal"><br><br>
```

```
  <input type="submit" value="Enviar">
```

```
</form>
```

Agrupamento de dados usando <fieldset>

O elemento <fieldset> é usado para agrupar dados relacionados em um formulário.

O elemento <legend> define uma legenda para o elemento <fieldset>

Agrupamento de dados usando <fieldset>

```
<form action="/action_page.php">
  <fieldset>
    <legend>Informações Pessoais:</legend>
    Nome:<br>
    <input type="text" name="nome" value="Fulano">
    <br>
    Sobrenome:<br>
    <input type="text" name="sobrenome" value="De Tal">
    <br><br>
    <input type="submit" value="Enviar">
  </fieldset>
</form>
```

Agrupamento de dados usando <fieldset>

Informações Pessoais:

Nome:
Fulano

Sobrenome:
De Tal

Enviar

Exercícios

Elementos de formulário

Elemento <input>

O elemento mais importante é elemento <input>

O elemento <input> pode ser exibido de maneiras diferentes dependendo do atributo **type**.

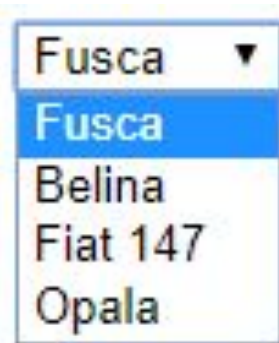
Elemento <input>

Se o atributo **type** for omitido, o padrão é usado (type="text")

Elemento <select>

O elemento <select> define uma lista de seleção:

```
<select name="carros">  
  <option value="fusca">Fusca</option>  
  <option value="belina">Belina</option>  
  <option value="fiat147">Fiat 147</option>  
  <option value="opala">Opala</option>  
</select>
```



Elemento <select>

O elemento **<option>** define uma opção que pode ser selecionada.

Por padrão, o primeiro item na lista é selecionado.

Para definir outro elemento <option> como selecionado utilize o atributo **selected**

```
<option value="fiat147" selected>Fiat 147</option>
```

Elemento <select>

```
<select name="carros">  
  <option value="fusca">Fusca</option>  
  <option value="belina">Belina</option>  
  <option value="fiat147" selected>Fiat 147</option>  
  <option value="opala">Opala</option>  
</select>
```

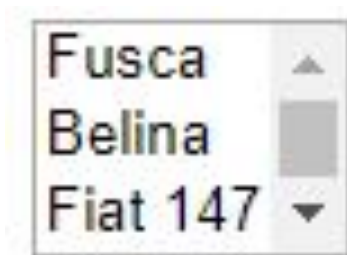


Elemento <select>

Valores visíveis:

Use o atributo **size** para especificar o número de opções visíveis.

```
<select name="carros" size="3">  
  <option value="fusca">Fusca</option>  
  <option value="belina">Belina</option>  
  <option value="fiat147">Fiat 147</option>  
  <option value="opala">Opala</option>  
</select>
```



Elemento <select>

```
<select name="carros" size="3" multiple>
```

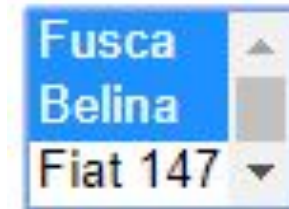
```
  <option value="fusca">Fusca</option>
```

```
  <option value="belina">Belina</option>
```

```
  <option value="fiat147">Fiat 147</option>
```

```
  <option value="opala">Opala</option>
```

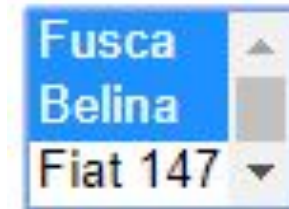
```
</select>
```



Elemento <select>

O atributo **multiple** permite que o usuário selecione mais de um valor.

```
<select name="carros" size="3" multiple>
```



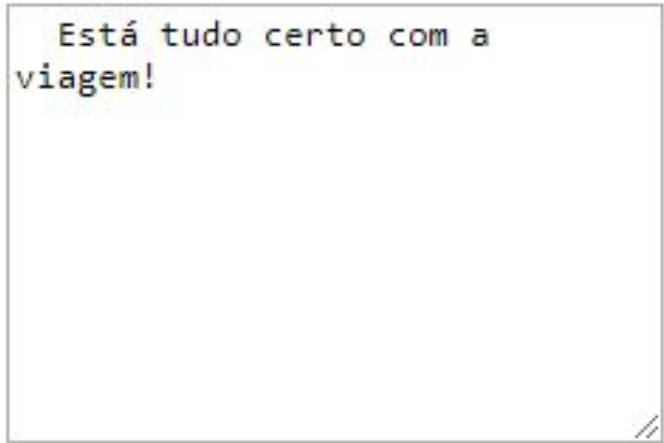
Elemento <textarea>

O elemento <textarea> define um campo de entrada de texto com mais de uma linha.

```
<textarea name="mensagem" rows="10" cols="30">
```

Está tudo certo com a viagem!

```
</textarea>
```



```
Está tudo certo com a  
viagem!
```

Elemento <textarea>

O atributo **rows** especifica o número de linhas visíveis no campo de texto.

O atributo **cols** especifica a largura visível no campo de texto.

```
<textarea name="mensagem" rows="10" cols="30">
```

Elemento <textarea>

É possível usar CSS para controlar a largura e altura:

```
<!DOCTYPE html>
<html>
<head>
<style>
  textarea { width: 100px; height:100px; }
</style>
</head>
<body>
  <textarea name="message" >0 gato morreu!</textarea>
</body>
</html>
```



Elemento button

O elemento `<button>` define um botão clicável.

```
<button type="button">Enviar</button>
```



Elemento `<datalist>`

O elemento **`<datalist>`** especifica uma lista de opções pré-definidas para um elemento `<input>`.

Usuários verão uma lista de opções conforme eles digitam os dados.

Elemento <datalist>

O atributo **list** do elemento <input> precisa se referir ao atributo id do elemento <datalist>

```
<input list="navegadores" name="browser">  
  <datalist id="navegadores">  
    <option value="Internet Explorer">  
    <option value="Firefox">  
    <option value="Chrome">  
    <option value="Opera">  
    <option value="Safari">  
  </datalist>
```



A screenshot of a web browser interface. It features a text input field with a blue border containing the letter 'C'. To the right of the input field is a small downward-pointing triangle icon. Below the input field, a dropdown menu is open, displaying the word 'Chrome' in bold black text. To the right of the dropdown menu is a grey button with the text 'Enviar' in black.

Elemento <output>

O elemento <output> representa o resultado de um cálculo.

Elemento <output>

```
<form oninput="dobro.value=parseInt(numero.value)*2">
```

```
  <input type="number" id="numero" value="0">
```

```
  Dobro: <output name="dobro"></output>
```

```
</form>
```

Dobro: 6

Exercícios

CSS - prioridade

- 1 - CSS inline (dentro de um elemento HTML)
- 2 - CSS externo e interno (na seção <head>)
- 3 - Padrão Navegador..

CSS - cores

Cores são especificadas usando alguns nomes conhecidos (blue,red,...) ou valores dos tipos RGB, HEX, HSL, RGBA, HSLA

Nomes de cores

Os navegadores aceitam cerca de 140 nomes de cores:

https://www.w3schools.com/colors/colors_names.asp

Cor de fundo de elementos

Você pode ajustar a cor de fundo dos elementos HTML usando background-color:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
h1 { background-color:DodgerBlue; }
```

```
p { background-color:Tomato; }
```

Cor de texto

Podemos alterar a cor do texto usando **color**

```
p { color:MediumSeaGreen; }
```


Cor da borda

Podemos ajustar a borda de elementos usando **border**:

```
p { border:2px solid Tomato; }
```

Valores RGB

Uma cor pode ser especificada usando um valor RGB:

```
rgb( vermelho, verde, azul )
```

Cada parâmetro (vermelho, verde, azul) define a intensidade da cor entre 0 e 255.

`rgb(255, 0, 0)` é mostrado como vermelho, o valor vermelho está ajustado para 255.

Valores RGB

Para mostrar a cor preta, todos os parâmetros devem ser ajustados para 0:

```
rgb( 0, 0, 0 );
```

Para mostrar a cor branca, todos os parâmetros devem ser ajustados para 255:

```
rgb(255, 255, 255);
```

Tons de cinza são alcançados com valores iguais para as três cores:

```
rgb( 120, 120, 120 );
```

Valores HEX

É possível especificar uma cor usando valores hexadecimais:

#rrggbb

rr é o valor hexadecimal para vermelho. (mínimo 00, máximo FF)

gg é o valor hexadecimal para verde. (mínimo 00, máximo FF)

bb é o valor hexadecimal para azul. (mínimo 00, máximo FF)

Valores HEX

Para mostrar a cor preta, todos os parâmetros devem ser ajustados para 00:

#000000

Para mostrar a cor branca, todos os parâmetros devem ser ajustados para FF:

#FFFFFF

Tons de cinza são alcançados com valores iguais para as três cores:

#777777

Valores HSL

Uma cor pode ser especificada usando hue (matiz), saturation e lightness na forma:

```
hsl(hue, saturation, lightness)
```

Matiz (hue) é o grau na roda de cores de 0 a 360, 0 é vermelho, 120 é verde, 240 é azul.

Saturação (saturation) em percentual, 0% significa um tom de cinza, 100% é cor completa.

Lightness (luminosidade) em percentual, 0% é preto, 50% nem preto nem branco, 100% é branco.

Saturação - Saturation

Saturação pode ser descrita como a intensidade de uma cor:

100% é a cor pura, sem tons de cinza.

50% é 50% cinza, mas ainda é possível ver a cor.

0% é completamente cinza, não é possível ver a cor.

hsl(0, 100%, 50%)

hsl(0, 75%, 50%)

hsl(0, 50%, 50%)

hsl(0, 25%, 50%)

hsl(0, 0%, 50%)

Luminosidade - Lightness

A luminosidade de uma cor pode ser descrita como a quantidade de luz de uma cor onde:

0% significa ausência de luz (preto),

50% significa 50% luz (nem escuro nem claro),

100% é luminosidade completa (branco).



hsl(0, 100%, 25%)

hsl(0, 100%, 50%)

hsl(0, 100%, 75%)

hsl(0, 100%, 100%)

Tons de cinza - hsl

Tons de cinza são definidos ajustando a matiz e saturação para 0, e ajustando a luminosidade de 0% a 100% para ter tons mais escuros ou claros:



hsl(0, 0%, 25%)



hsl(0, 0%, 50%)



hsl(0, 0%, 75%)



hsl(0, 0%, 100%)

Valor RGBA

Valores de cores RGBA são uma extensão dos valores de cor RGB com um canal alfa - que especifica a opacidade de uma cor.

Um valor RGBA é especificado com:

```
rgba(vermelho, verde, azul, alfa)
```

Valor RGBA

O parâmetro alfa é um número entre 0.0 (totalmente transparente) e 1.0 (sem nenhuma transparência, opaco):

rgba(255, 99, 71, 0)

rgba(255, 99, 71, 0.25)

rgba(255, 99, 71, 0.5)

rgba(255, 99, 71, 0.75)

rgba(255, 99, 71, 1)

Valor HSLA

Valores de cores HSLA são uma extensão dos valores HSL com um canal alfa - que especifica a opacidade de uma cor.

Um valor de cor HSLA é especificado assim:

```
hsla(matiz, saturação, luminosidade, alfa)
```

Valor HSLA

O parâmetro alfa é um número entre 0.0 (totalmente transparente) e 1.0 (opaco):

hsla(9, 100%, 64%, 0)

hsla(9, 100%, 64%, 0.5)

hsla(9, 100%, 64%, 0.75)

hsla(9, 100%, 64%, 1)

Pano de fundo com CSS

Algumas propriedades CSS ajustam o pano de fundo (background):

- background-color
- background-image
- background-repeat
- background-attachment
- background-position

Background-color

A propriedade **background-color** especifica a cor do pano de fundo de um elemento.

A cor de fundo de uma página é ajustada assim:

```
body {  
  
    background-color: lightblue;  
  
}
```

Background-color

A cor pode ser ajustada usando:

Um nome de cor - como **blue**

Um valor HEX - #ff0000

Um valor RGB - rgb(255,0,0)

Um valor RGBA - rgba(255,0,0,1)

Um valor HSL ou HSLA

Background-color

Elementos podem ter cores de fundo diferentes:

```
h1 {  
    background-color: green;  
}
```

```
div {  
    background-color: lightblue;  
}
```

```
p {  
    background-color: yellow;  
}
```

Imagem de fundo

A propriedade **background-image** especifica uma imagem para usar como fundo de um elemento.

O padrão é repetir a imagem para que ela cubra todo o elemento.

A imagem de fundo de uma página pode ser ajustada assim:

```
body {  
  
    background-image: url("papel.gif");  
  
}
```

<https://www.w3schools.com/css/paper.gif>

Imagem de fundo - controle da repetição

Por padrão uma imagem de fundo com **background-image** se repete tanto na horizontal quanto na vertical.

Algumas imagens devem se repetir apenas horizontalmente ou verticalmente para não ficar estranho.

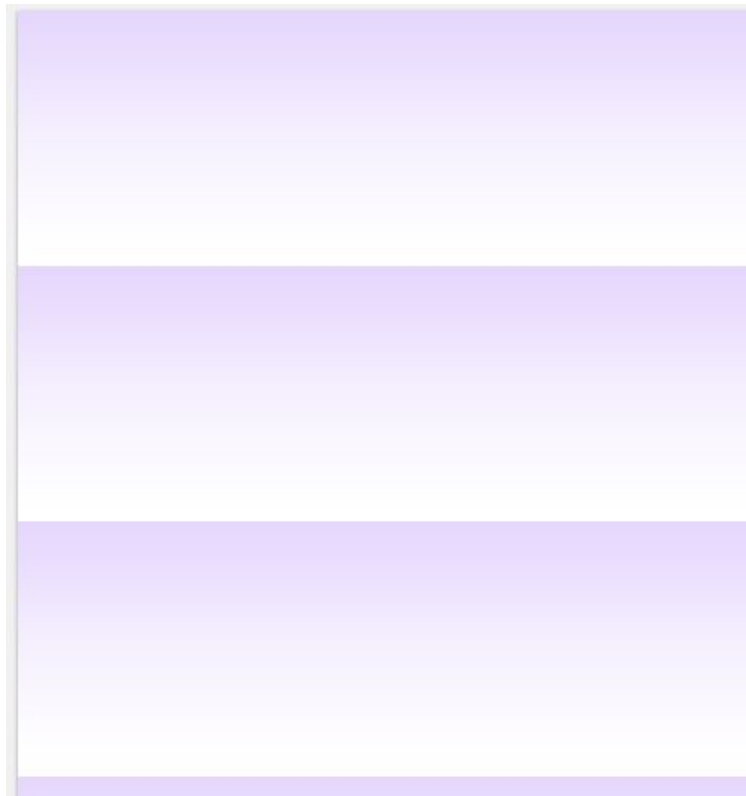


Imagem de fundo - controle da repetição

Para repetir uma imagem apenas na horizontal utilize:

background-repeat: repeat-x;

```
body {
```

```
    background-image: url("gradient_bg.png");
```

```
    background-repeat: repeat-x;
```

```
}
```

https://www.w3schools.com/css/gradient_bg.png

Imagem de fundo - controle da repetição

Para repetir uma imagem apenas na vertical utilize:

background-repeat: repeat-y;

```
body {
```

```
    background-image: url("gradient_bg.png");
```

```
    background-repeat: repeat-y;
```

```
}
```

https://www.w3schools.com/css/gradient_bg.png

Ajustando a posição e repetição

Para mostrar a imagem de fundo apenas uma vez use a propriedade **background-repeat**:

```
body {  
  
    background-image: url("img_tree.png");  
  
    background-repeat: no-repeat;  
  
}
```

Ajustando a posição e repetição

A posição da imagem de fundo é ajustada pela propriedade **background-position**:

```
body {  
  
    background-image: url("img_tree.png");  
  
    background-repeat: no-repeat;  
  
    background-position: right top;  
  
}
```

Posição fixa

Para dizer que uma imagem de fundo deve ser fixa (não deve rolar com a página, use a propriedade **background-attachment**:

```
body {  
  
    background-image: url("img_tree.png");  
  
    background-repeat: no-repeat;  
  
    background-position: right top;  
  
    background-attachment: fixed;  
  
}
```


Background reduzido

Para reduzir o código, é possível ajustar todas as propriedades do pano de fundo em uma única propriedade **background**:

```
body {  
    background: #ffffff url("img_tree.png") no-repeat right top;  
}
```

Exercícios

Bordas CSS

As propriedades de borda do CSS permitem que você especifique:

- o estilo
- largura
- e cor

da borda de um elemento.

Border-style

A propriedade **border-style** especifica que tipo de borda mostrar.

```
p {border-style: dotted;}
```

A propriedade border-style pode ter de um a quatro valores (para borda superior, borda direita, borda inferior, borda esquerda).

```
p {border-style: dotted dashed solid double;}
```

Border-style

Valores permitidos:

dotted - Define uma borda pontilhada

dashed - Define uma borda tracejada

solid - Define uma borda sólida

double - Define uma borda dupla

groove - Define uma borda 3D entalhada. O efeito depende de border-color.

ridge - Define uma borda 3D com estrias. O efeito depende de border-color.

inset - Define uma borda 3D inset. O efeito depende de border-color.

Border-style

outset - Define uma borda 3D outset. O efeito depende de border-color.

none - Define nenhuma borda.

hidden - Define uma borda escondida.

Border-width

A propriedade border-width especifica a largura das quatro bordas.

A largura pode ser ajustada como um tamanho específico (em px, pt, cm, em, etc) ou usando um dos três valores pré-definidos: thin, medium, ou thick.

Border-width

```
p.one {  
    border-style: solid;  
    border-width: 5px;  
}
```

```
<p class="one">Primeiro.</p>  
<p class="two">Segundo.</p>  
<p class="three">Terceiro.</p>
```

```
p.two {  
    border-style: solid;  
    border-width: medium;  
}
```

```
p.three {  
    border-style: solid;  
    border-width: 2px 10px 4px 20px;  
}
```


Border-color

A propriedade **border-color** é usada para ajustar a cor das quatro bordas.

A propriedade border-color pode ter de um a quatro valores (para borda superior, borda direita, borda inferior, borda esquerda).

Se border-color não for ajustada, ela herda a cor do elemento.

Border-color

A cor pode ser ajustada usando:

Um nome de cor - como blue

Um valor HEX - #ff0000

Um valor RGB - rgb(255,0,0)

Um valor RGBA - rgba(255,0,0,1)

Um valor HSL ou HSLA

Lados individuais

É possível especificar uma borda diferente para cada lado.

Em CSS, existem propriedades para ajustar cada uma das bordas (top, right, bottom, and left):

```
p {  
    border-top-style: dotted;  
    border-right-style: solid;  
    border-bottom-style: dotted;  
    border-left-style: solid;  
}
```

Lados individuais

```
p {  
    border-top-style: dotted;  
    border-right-style: solid;  
    border-bottom-style: dotted;  
    border-left-style: solid;  
}
```

equivale à:

```
p {  
    border-style: dotted solid;  
}
```

Lados individuais

Funcionamento:

Com 4 valores:

```
border-style: dotted solid double dashed;
```

borda superior é dotted

borda direita é solid

borda inferior é double

borda esquerda é dashed

Lados individuais

Funcionamento:

Com 3 valores:

```
border-style: dotted solid double;
```

borda superior é dotted

borda direita e esquerda é solid

borda inferior é double

Lados individuais

Funcionamento:

Com 2 valores:

```
border-style: dotted solid;
```

borda superior e inferior é dotted

borda direita e esquerda é solid

Lados individuais

Funcionamento:

Com 1 valor:

```
border-style: dotted;
```

Todas as bordas são dotted.

Esse sistema funciona para border-style, border-width, border-color.

Propriedade resumida - border

É possível resumir as propriedades de borda em uma propriedade **border**

- border-width
- border-style (required)
- border-color

```
p {  
    border: 5px solid red;  
}
```

Propriedade resumida - border

É possível especificar todas as propriedades individuais para apenas um lado:

```
p {  
    border-left: 6px solid red;  
    background-color: lightgrey;  
}
```

Bordas arredondadas

A propriedade **border-radius** é usada para adicionar bordas arredondadas à um elemento:

```
p {  
    border: 2px solid red;  
    border-radius: 5px;  
}
```

Quanto maior o valor mais arredondada a borda.

Exercícios