

Desenvolvimento para Web

Módulo Básico





ESTILIZAÇÃO

- **Bordas Arredondadas**
 - **Propriedades CSS para estabelecer bordas arredondadas**
 - ***border-top-left-radius***: raio de **borda superior esquerda** do box
 - ***border-top-right-radius***: raio da **borda superior direita** do box
 - ***border-bottom-right-radius***: raio da **borda inferior direita** do box
 - ***border-bottom-left-radius***: raio da **borda inferior esquerda** do box
 - ***border-radius***: **sintaxe abreviada** para definir o raio da borda nos **quatro cantos** do box
 - Os **valores CSS** para essas **propriedades** são as **medidas CSS** de **comprimento**, como **px**, **em**, **pt** etc. e **porcentagem**

- **Bordas Arredondadas**

- Para demonstrar a **aplicação** de diferentes **tipos** de **borda arredondada**, usaremos **boxes retangulares simples**
- Podemos **declarar um** ou **dois** valores para a **curvatura** de cada um dos cantos de um **box individualmente**, obtendo **efeitos diferentes**

`border-top-left-radius: 80px;`

ou

`border-top-left-radius: 80px 40px;`

- **Bordas Arredondadas**

- **Declarar um só valor produz *curvaturas iguais* nos sentidos *x* e *y*, e declarar dois valores produz *curvatura igual* ao primeiro valor declarado no sentido *x* e ao segundo valor declarado no sentido *y***

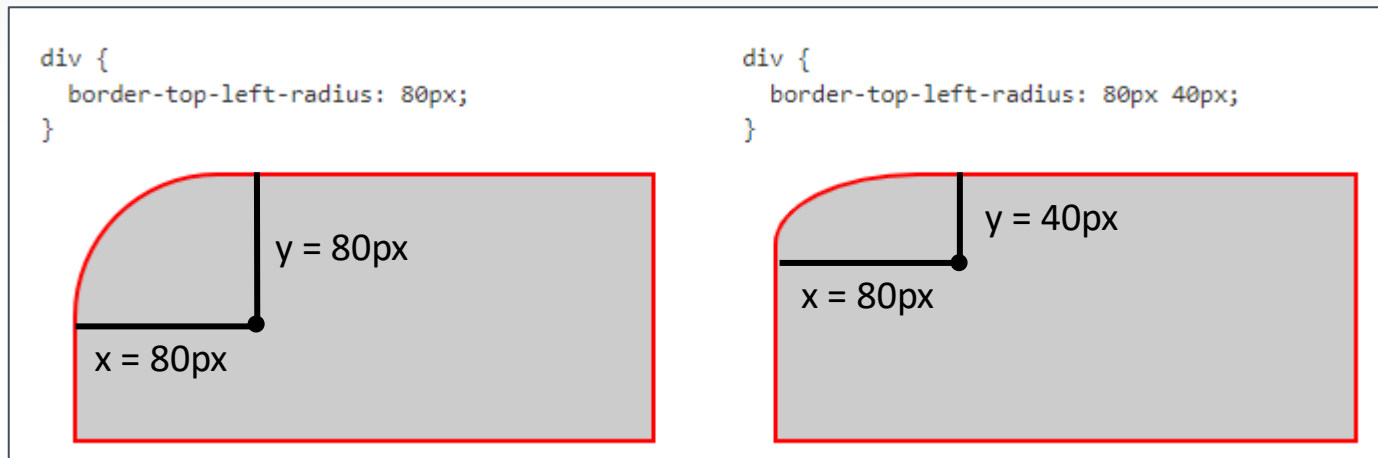


Figura 1 – Bordas arredondadas

- **Bordas Arredondadas**

- Para aplicar **bordas arredondadas** aos **quatro cantos** do **box**, podemos ainda usar a **declaração abreviada** **border-radius** que **admite um, dois, três ou quatro valores**
- **border-radius: raio;**
 - Os **quatro cantos** serão **arredondados igualmente** com **raio de curvatura** com **valor igual a raio**

```
div {  
  border-radius: 15px;  
}
```



- **Bordas Arredondadas**

- **`border-radius: raio1 raio2;`**

- Os cantos **`top-left`** e **`bottom-right`** serão arredondados igualmente com **raio de curvatura** com **valor igual a `raio1`**; e os cantos **`top-right`** e **`bottom-left`** serão arredondados igualmente com **raio de curvatura** com **valor igual a `raio2`**

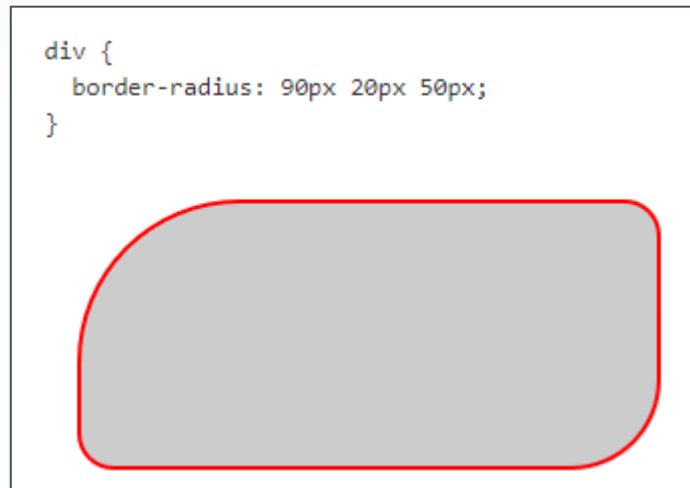
```
div {  
  border-radius: 90px 20px;  
}
```



- **Bordas Arredondadas**

- ***border-radius: raio1 raio2 raio3;***

- Os cantos *top-left* será **arredondado** com **raio** de **curvatura** com **valor igual a raio1**, os cantos *top-right* e *bottom-left* serão **arredondados igualmente** com **raio** de **curvatura** com **valor igual a raio2** e o canto *bottom-right* será **arredondado** com **raio** de **curvatura** com **valor igual a raio3**



- **Bordas Arredondadas**

- ***border-radius: raio1 raio2 raio3 raio4;***

- *Os raios de curvatura declarados serão aplicados na seguinte ordem: top-left, top-right, bottom-right e bottom-left*

```
div {  
  border-radius: 90px 70px 50px 30px;  
}
```



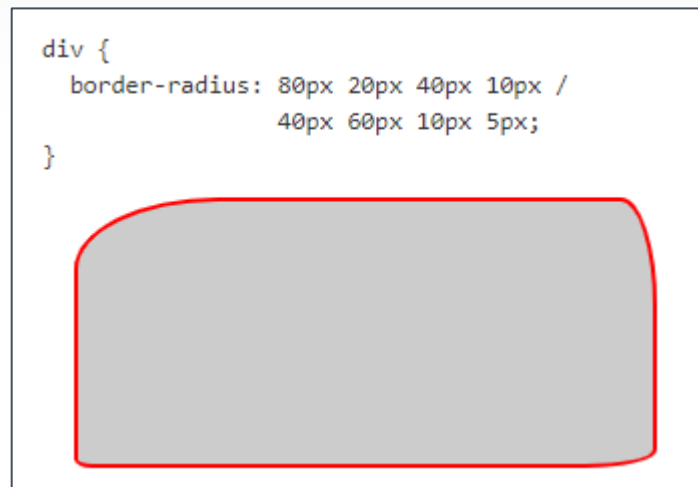
- **Bordas Arredondadas**

- A *declaração simplificada* `border-radius` também *admite a sintaxe para declarar valores para curvatura da borda segundo os eixos **x** e **y***
- *Nesse caso devemos separar as declarações para os eixos **x** e **y** com uma barra (/)*

```
border-radius: 80px 20px 40px 10px / 40px 60px 10px 5px;
```

- **Bordas Arredondadas**

- A *declaração simplificada* `border-radius` também *admite a sintaxe para declarar valores para curvatura da borda segundo os eixos **x** e **y***
- Nesse caso *devemos separar as declarações para os eixos **x** e **y** com uma barra (/)*



- **Bordas Arredondadas**

- *Essa sintaxe admite que se declare de **um** a **quatro valores** para **qualquer uma das duas curvaturas** **x** e **y***

`border-radius: 80px / 40px 60px 10px 5px;`

`border-radius: 80px 20px 40px / 40px 60px;`

`border-radius: 80px 20px / 10px;`

- **Sombras**

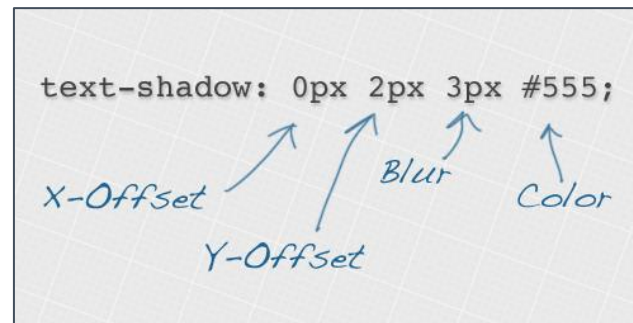
- **Sombra em Texto**

- A **propriedade CSS** para **declarar sombra em texto** é **text-shadow**
 - A **sintaxe** para **definir sombra** em **texto** consiste em um **valor CSS** que é **opcional** para **cor**, seguido de **duas unidades de medida CSS** (**obrigatórias**), que **definem** os **deslocamentos** da **sombra** na **horizontal** e **vertical**, e **opcionalmente** mais uma **unidade de medida CSS** que **define** o efeito **blur** (**borrado**) da **cor**
 - **Todos os valores** devem ser **separados por espaço em branco**
 - A **ordem de declaração** dos **valores** é a **seguinte**: **cor**, **deslocamento horizontal**, **deslocamento vertical** e **blur**

- **Sombras**

- **Sombra em Texto**

- Os valores para **deslocamento** da **sombra** podem ser **positivos** ou **negativos**
 - Valores **positivos** indicam **deslocamento horizontal** para a **direita** e **vertical** para **baixo**, e **negativos** indicam **deslocamento** nos **sentidos inversos**
 - **Não** sendo **declarado valor** para a **cor**, a **sombra** será na **mesma cor** do **texto** do **box**



- **Sombras**

- **Sombra em Texto**

- *É possível declarar simultaneamente **quatro** posicionamentos de **sombra** para um mesmo box, usando as **quatro** possíveis combinações de sinais (**positivo** e **negativo**) para as duas unidades de medida CSS obrigatórias para definir deslocamentos*
 - *Nesse caso as declarações dos valores da sombra devem ser **separadas** por **vírgula***

- **Sombras**

- **Sombra em Texto**

- *Exemplos de sintaxe da regra CSS para aplicar **sombra** em **texto***

```
h1 { text-shadow: red 5px 5px; }  
h1 { text-shadow: red 5px 5px 5px; }  
h1 { text-shadow: red -5px 5px; }  
h1 { text-shadow: red 5px -5px; }  
h1 { text-shadow: red -5px -5px; }
```


- **Sombras**

- **Sombra em Texto**

- **Exemplos de sintaxe da regra CSS para aplicar *sombra* em *texto***

```
h1 {  
  text-shadow: red 3px 3px 5px,  
              blue -6px -6px,  
              green -25px 15px 5px;  
}
```

- **Sombras**

- **Sombra em Texto**

- **Exemplos de sintaxe da regra CSS para aplicar *sombra* em *texto***

```
h1 {text-shadow: red 5px 0;}
```

HTML5
bkBank

```
h1 {text-shadow: red 0 5px;}
```

HTML5
bkBank

```
h1 {text-shadow: red 5px 5px;}
```

HTML5
bkBank

- **Sombras**

- **Sombra em Texto**

- **Exemplos de sintaxe da regra CSS para aplicar *sombra* em *texto***

```
h1 {text-shadow: red 5px 5px 5px;}
```

HTML5
bkBank

```
h1 {text-shadow: red 5px 5px 15px;}
```

HTML5
bkBank

```
h1 {text-shadow: red 2px 2px 4px;}
```

HTML5
bkBank

- **Sombras**

- **Sombra em Texto**

- **Exemplos de sintaxe da regra CSS para aplicar *sombra* em *texto***

```
h1 {text-shadow: red -5px 5px;}
```

HTML5
bkBank

```
h1 {text-shadow: red 5px -5px;}
```

HTML5
bkBank

```
h1 {text-shadow: red -5px -5px;}
```

HTML5
bkBank

- **Sombras**

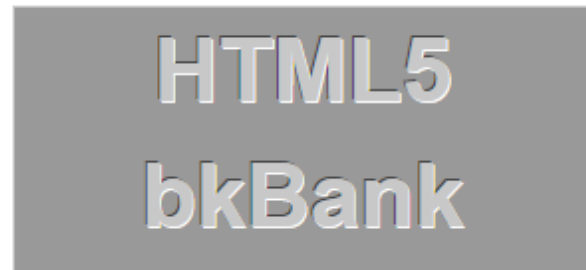
- **Sombra em Texto**

- **Exemplos de sintaxe da regra CSS para aplicar *sombra* em *texto***

```
div{ background: #ddd; }  
h1 {  
  text-shadow: red 3px 3px 5px,  
              blue -6px -6px,  
              green -25px 15px 5px;  
}
```



```
div { background: #999; }  
h1 {  
  color: #c8c8c8;  
  text-shadow: 1px 1px white,  
              -1px -1px #444;  
}
```

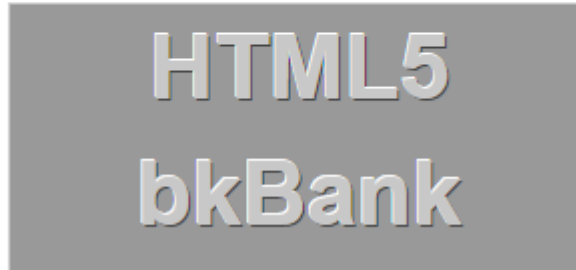


- **Sombras**

- **Sombra em Texto**

- **Exemplos de sintaxe da regra CSS para aplicar *sombra* em *texto***

```
div { background: #999; }  
h1 {  
  color: #c8c8c8;  
  text-shadow: -1px -1px white,  
              1px 1px #444;  
}
```



- **Sombras**

- **Sombra em Box**

- A **propriedade CSS** para declarar sombra em box é **box-shadow**
 - A **sintaxe** para definir **sombra** em **box** consiste da **palavra-chave inset** (**opcional**), seguida de **duas unidades de medida CSS** (**obrigatórias**), que **definem** os **deslocamentos** da **sombra** na **horizontal** e **vertical**, e **opcionalmente** mais uma **unidade de medida CSS** que **define** uma **expansão** (**valor positivo**) ou uma **contração** (**valor negativo**) da **sombra**
 - **Todos os valores** devem ser **separados por espaço em branco**
 - A **palavra-chave opcional inset**, quando declarada, faz com que a **sombra** seja **interna** ao **box**

- **Sombras**

- **Sombra em Box**

- Valores **positivos** indicam deslocamento **horizontal** para a **direita** e **vertical** para **baixo**, e **negativos** indicam deslocamento nos **sentidos inversos**
 - **Não** sendo declarado valor para a **cor**, a sombra será na **mesma cor do texto do box**
 - É possível declarar simultaneamente **quatro** posicionamentos de **sombra** para um mesmo box, usando as **quatro** possíveis combinações de sinais (**positivo** e **negativo**) para as duas unidades de medida CSS obrigatórias para definir deslocamentos
 - Nesse caso as declarações dos valores da sombra devem ser **separadas** por **vírgula**

- **Sombras**

- **Sombra em Box**

- *Exemplos de sintaxe da regra CSS para aplicar **sombra** em **box***

```
div { box-shadow: red 5px 5px; }  
div { box-shadow: #f90 5px 5px 5px; }  
div { box-shadow: #f90 5px 5px 5px 15px; }  
div { box-shadow: inset #f90 -10px 10px 15px; }
```

- **Sombras**

- **Sombra em Box**

- *Exemplos de sintaxe da regra CSS para aplicar **sombra** em **box***

```
div {  
  box-shadow: #f90 10px 0 8px -4px,  
             #f90 -10px 0 8px -4px;  
}
```

- **Sombras**

- **Sombra em Box**

- *Exemplos de sintaxe da regra CSS para aplicar **sombra** em **box***

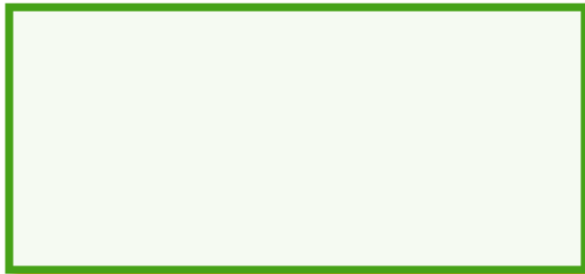
```
div {  
  box-shadow: #f90 0 5px 4px -2px,  
             #f90 0 -5px 4px -2px;  
}  
  
div { box-shadow: #f90 0 0 10px 5px; }
```

- **Sombras**

- **Sombra em Box**

- **Exemplos de sintaxe da regra CSS para aplicar *sombra* em *box***

```
div {box-shadow: #f90 5px 5px 5px;}
```



```
div {box-shadow: #f90 5px 5px 5px 15px;}
```



```
div {box-shadow: inset #f90 -10px 10px 15px;}
```

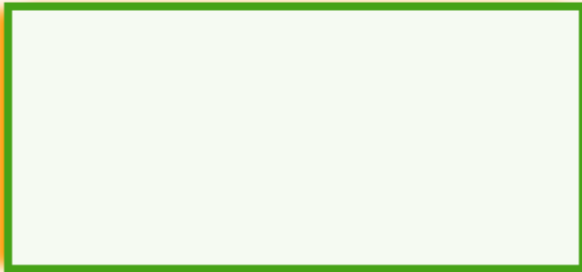


- **Sombras**

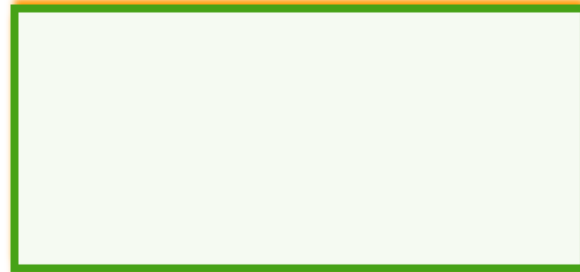
- **Sombra em Box**

- **Exemplos de sintaxe da regra CSS para aplicar *sombra* em *box***

```
div {  
  box-shadow: #f90 10px 0 8px -4px,  
             #f90 -10px 0 8px -4px;  
}
```

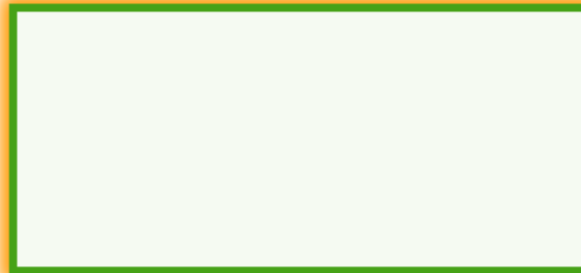


```
div {  
  box-shadow: #f90 0 5px 4px -2px,  
             #f90 0 -5px 4px -2px;  
}
```



- **Sombras**
 - **Sombra em Box**
 - *Exemplos de sintaxe da regra CSS para aplicar **sombra** em **box***

```
div {  
  box-shadow: #f90 0 0 10px 5px;  
}
```



- **Opacidade**

- Uma **propriedade CSS** que se **destina a definir a opacidade (transparência) de um elemento e seus elementos descendentes e bordas**
- O **valor possível para essa propriedade é um número compreendido entre 0 (zero) e 1**
- O **valor 0 (zero) representa transparência total**
- O **valor 1 representa opacidade total**
- Um **valor igual a 0.9 (90% opaco ou 10% transparente)**

- **Opacidade**

- Aplicar **opacidade** em um **elemento** faz com que **todos** os **elementos** nele contidos **herdem** o **valor** da **opacidade** a ele aplicada
- Considere o texto de um parágrafo inserido em um **div**

```
<div>  
  <p class="um"> Texto do parágrafo um </p>  
  <p class="dois"> Texto do parágrafo dois </p>  
</div>
```

#HTML

- **Opacidade**

- *Considere o texto de um parágrafo inserido em um **div***

```
div {  
  opacity: 0.5;  
  color: black;  
}  
  
p.um { opacity: 0.5 }  
p.dois { opacity: 1 }
```

CSS

- **Opacidade**

- **Considere o texto de um parágrafo inserido em um *div***

O texto dos dois parágrafos herdará a opacidade 0.5, definida para o elemento div

A regra de estilo que define a opacidade para o p.um faz com que o texto tenha opacidade $0.5 * 0.5 = 0.25$

A regra de estilo que define a opacidade para o p.dois **não** produz **nenhum efeito**, pois **não** é possível aumentar a opacidade herdada, somente diminuí-la, como foi feito com p.um

CSS

- *Opacidade*

- *O uso de cores **RGBA** ou **HSLA** para aplicar opacidade, ao contrário do uso de **opacity**, **não** faz com que os elementos descendentes herdem a opacidade*

```
<div>  
  <p class="um"> Texto do parágrafo um </p>  
  <p class="dois"> Texto do parágrafo dois </p>  
</div>
```

#HTML

- *Opacidade*

- *O uso de cores **RGBA** ou **HSLA** para aplicar opacidade, ao contrário do uso de **opacity**, **não** faz com que os elementos descendentes herdem a opacidade*

```
div { color: rgba(0, 0, 0, 0.5); }  
p.um { color: rgba(0, 0, 0, 0.5); }  
p.dois { color: rgba(0, 0, 0, 1); }
```

CSS

- **Opacidade**

- *O uso de cores **RGBA** ou **HSLA** para aplicar opacidade, ao contrário do uso de **opacity**, **não** faz com que os elementos descendentes herdem a opacidade*

Nesse exemplo, diferentemente do anterior, p.um terá **opacidade** 0.5 e p.dois será totalmente opaco

Isso porque aplicar opacidade com uso RGBA (ou HSLA) **não** implica herança CSS

CSS

- **Opacidade**

- Criamos **três containers** e neles inserimos um **conteúdo textual** e uma **imagem** do tipo **PNG** transparente
- Para o **primeiro container**, **não** definimos opacidade (equivalente em definir opacidade igual a **1**)
- No **segundo** definimos opacidade igual a **0.6** e no **terceiro** **0.4**



- *Opacidade*

```
div { /* opacidade padrão - opaco */  
  opacity:1;  
  background-image: url(logo.png);  
  background-size: contain;  
  background-position: center bottom;  
  background-repeat: no-repeat;  
}
```



```
div {  
  opacity: 0.7;  
  background-image: url(logo.png);  
  background-size: contain;  
  background-position: center bottom;  
  background-repeat: no-repeat;  
}
```



- *Opacidade*

```
div {  
  opacity: 0.4;  
  background-image: url(logo.png);  
  background-size: contain;  
  background-position: center bottom;  
  background-repeat: no-repeat;  
}
```

Conteúdo textual na cor vermelha
dentro do container com fundo preto



- **Gradientes**

- É possível **criar gradientes** de cores para serem usados como **imagens** de **fundo**
- **Gradiente CSS** é uma **imagem** que **apresenta** uma **transição suave** entre **duas** ou **mais cores**
- A **transição** das **cores** pode ser **segundo** um **eixo** (**uma linha reta**) dando **origem** a uma **gradiente linear**, ou **segundo círculos concêntricos** criando um **gradiente radial**
- As **funcionalidades CSS** para **definição** de **gradientes** possibilitam que se **crie transição abrupta** de **cor** em lugar de **transição suave padrão**

- **Gradientes**

- **Gradiente Linear**

- A **transição** de uma **cor** para **outra** é feita ao **longo** de **um eixo** que **determina a direção** da **transição**
 - A **sintaxe** para **criar** um **gradiente linear** a ser **aplicado** como **valor** para uma **propriedade CSS** que admita **imagem** como **valor** (**background-image** ou **list-style-image**), usa **uma função CSS** denominada **linear-gradient()** cujos **parâmetros** são:

linear-gradient(**direção**, **cor1 stop**, **cor2 stop**, **cor3 stop**, ..., **corn stop**)

- **Gradientes**
 - **Gradiente Linear**

`linear-gradient(direção, cor1 stop, cor2 stop, cor3 stop, ..., corn stop)`

- *Repare que os parâmetros são separados por vírgula*
- **direção**: define o eixo do **gradiente linear** que pode ser declarado com uso de uma medida de ângulo CSS ou da palavra inglesa **to** seguida de um **lado** ou de um **canto**
- **cor**: define as **cores** do **gradiente** e devem ser declaradas **duas** ou **mais cores**
- **stop**: define uma **distância medida** sobre o eixo do **gradiente** em que será feita a **transição de cores**

- **Gradientes**

- **Gradiente Linear**

- *Exemplos: para a 1ª e 4ª regras CSS, o gradiente foi declarado **sem** uso do parâmetro **stop**, e para as demais, **com** uso do parâmetro*

```
/* sem stop */
```

```
background-image: linear-gradient(to left, black, white)
```

```
/* com stop */
```

```
background-image: linear-gradient(to right, white, black 20%)
```

```
/* com stop */
```

```
background-image: linear-gradient(to right, white, black 180px)
```

- **Gradientes**

- **Gradiente Linear**

- *Exemplos: para a 1ª e 4ª regras CSS, o gradiente foi declarado **sem** uso do parâmetro **stop**, e para as demais, **com** uso do parâmetro*

```
/* sem stop */
```

```
background-image: linear-gradient(to left, black, white, black, white)
```

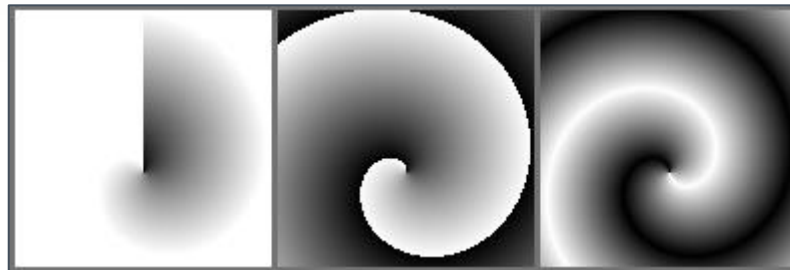
```
/* com stop */
```

```
background-image: linear-gradient(to right, black 33%,  
                                pink 33%, pink 67%,  
                                gray 67%)
```

- **Gradientes**

- **Gradiente Radial**

- A **transição** de uma **cor** para **outra** se faz **segundo linhas radiais**, a **partir de um ponto chamado centro** do **gradiente**
 - A **forma** das **linhas radiais** pode ser **circular** ou **elíptica**, criando-se **dois tipos de gradiente radial**



- **Gradientes**

- **Gradiente Radial**

- A *sintaxe* para *criar* um **gradiente radial** a ser aplicado como **valor** para **uma propriedade CSS** que **admita imagem** como **valor** (*background-image* ou *list-style-image*), **usa uma função CSS** denominada **radial-gradient()** cujos **parâmetros são**:

radial-gradient(coordenadas do centro,
forma tamanho,
cor1 stop, cor2 stop, cor3 stop, ..., corn stop)

- **Gradientes**
 - **Gradiente Radial**

`radial-gradient`(*coordenadas do centro, forma tamanho, cor1 stop, cor2 stop, cor3 stop, ..., corn stop*)

- *Repare que os parâmetros são separados por vírgula*
- *coordenadas do centro: define as coordenadas do centro do gradiente e admite os mesmos valores usados para definir a propriedade background-position (na omissão, center será o padrão)*

- **Gradientes**
 - **Gradiente Radial**

radial-gradient(coordenadas do centro, forma tamanho,
cor1 stop, cor2 stop, cor3 stop, ..., corn stop)

- **forma**: admite os valores **circle** e **ellipse** definindo o gradiente **circular** ou **elíptico**, e o parâmetro **tamanho** define como o gradiente se desenvolve em relação aos limites do box em que é aplicado
- Os valores possíveis são: **closest-side**, **farthest-side**, **closest-corner**, **farthest-corner**, **contain** e **cover**. É permitido definir um **valor explícito** com uso de coordenadas (o valor padrão é **cover**)

- **Gradientes**

- **Gradiente Radial**

- *Segue o significado de cada um desses valores:*

- **closest-side**: se a forma do gradiente for **circle**, ele deverá estender-se até o lado do container mais próximo do centro. Se a forma do gradiente for **ellipse**, ele deverá estender-se até os lados **horizontal** e **vertical** do container, mais próximo do **centro**
 - **farthest-side**: se a forma do gradiente for **circle**, ele deverá estender-se até o lado do container mais afastado do centro. Se a forma do gradiente for **ellipse**, ele deverá estender-se até os lados **horizontal** e **vertical** do container, mais afastados do **centro**

- **Gradientes**

- **Gradiente Radial**

- *Segue o significado de cada um desses valores:*

- **closest-corner**: se a forma do gradiente for **circle**, ele deverá estender-se até o canto do container mais próximo do **centro**. Se a forma do gradiente for **ellipse**, ele deverá estender-se até o canto do container mais próximo do **centro**, mas mantendo o mesmo **aspect-ratio**
 - **farthest-corner**: se a forma do gradiente for **circle**, ele deverá estender-se até o canto do container mais afastado do **centro**. Se a forma do gradiente for **ellipse**, ela deverá estender-se até o canto do container mais afastado do **centro**, mas mantendo o mesmo **aspect-ratio**

- **Gradientes**
 - **Gradiente Radial**
 - *Segue o significado de cada um desses valores:*
 - **contain**: esse valor produz o mesmo efeito de **closest-side**
 - **cover**: esse valor produz o mesmo efeito de **farthest-corner**

- **Gradientes**

- **Gradiente Radial**

- Exemplos de **gradientes radiais** como **imagem de fundo de um elemento *div***

```
.um {  
  background-image: -moz-radial-gradient(40px 60px, circle closest-side,  
                                         white, black, white);  
  background-image: -webkit-radial-gradient(40px 60px, circle closest-side,  
                                             white, black, white);  
  background-image: radial-gradient(40px 60px, circle closest-side, white,  
                                    black, white);  
}
```

- **Gradientes**

- **Gradiente Radial**

- Exemplos de **gradientes radiais** como **imagem de fundo de um elemento div**

```
.dois {  
  background-image: -moz-radial-gradient(40px 60px, circle farthest-side,  
                                         white, black, white);  
  background-image: -webkit-radial-gradient(40px 60px, circle farthest-side,  
                                             white, black, white);  
  background-image: radial-gradient(40px 60px, circle farthest-side, white,  
                                    black, white);  
}
```

- **Gradientes**

- **Gradiente Radial**

- Exemplos de **gradientes radiais** como **imagem de fundo de um elemento div**

```
.tres {  
  background-image: -moz-radial-gradient(40px 60px, circle closest-corner,  
                                         white, black, white);  
  background-image: -webkit-radial-gradient(40px 60px, circle closest-  
                                             corner, white, black, white);  
  background-image: radial-gradient(40px 60px, circle closest-corner, white,  
                                    black, white);  
}
```

- **Gradientes**

- **Gradiente Radial**

- Exemplos de **gradientes radiais** como **imagem de fundo de um elemento *div***

```
.quatro {  
  background-image: -moz-radial-gradient(40px 60px, circle farthest-corner,  
                                         white, black, white);  
  background-image: -webkit-radial-gradient(40px 60px, circle farthest-  
                                             corner, white, black, white);  
  background-image: radial-gradient(40px 60px, circle farthest-corner,  
                                    white, black, white);  
}
```


- **Gradientes**

- **Gradiente Radial**

- Exemplos de **gradientes radiais** como **imagem de fundo de um elemento *div***

```
.cinco {  
  background-image: -moz-radial-gradient(40px 60px, circle contain,  
                                         white, black, white);  
  background-image: -webkit-radial-gradient(40px 60px, circle contain,  
                                             white, black, white);  
  background-image: radial-gradient(40px 60px, circle contain,  
                                    white, black, white);  
}
```

- **Gradientes**

- **Gradiente Radial**

- Exemplos de **gradientes radiais** como **imagem de fundo de um elemento *div***

```
.seis {  
  background-image: -moz-radial-gradient(40px 60px, circle cover,  
                                         white, black, white);  
  background-image: -webkit-radial-gradient(40px 60px, circle cover,  
                                             white, black, white);  
  background-image: radial-gradient(40px 60px, circle cover,  
                                    white, black, white);  
}
```

- **Gradientes**

- **Gradiente Radial**

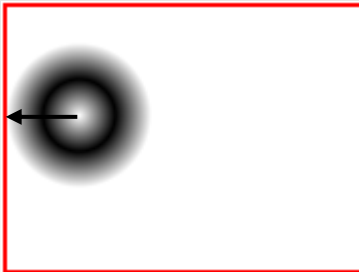
- Exemplos de **gradientes radiais** como **imagem de fundo de um elemento *div***

```
.sete {  
  background-image: -moz-radial-gradient(40px 60px, 100px 60px, white,  
                                         black, white);  
  background-image: -webkit-radial-gradient(40px 60px, 100px 60px, white,  
                                             black, white);  
  background-image: radial-gradient(40px 60px, circle 100px 60px, white,  
                                    black, white);  
}
```

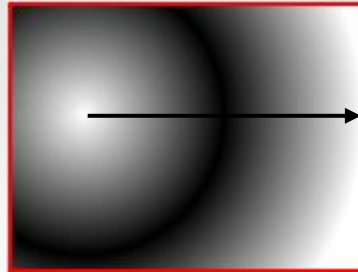
- **Gradientes**

- **Gradiente Radial**

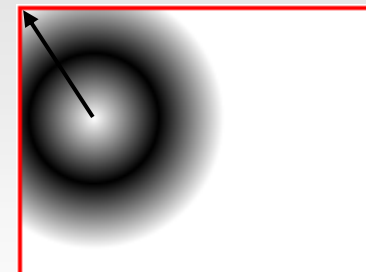
div.um | closest-side



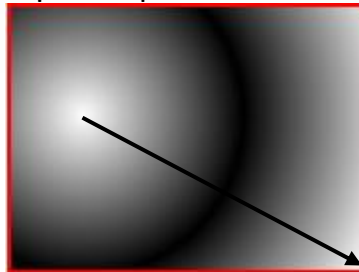
div.dois | farthest-side



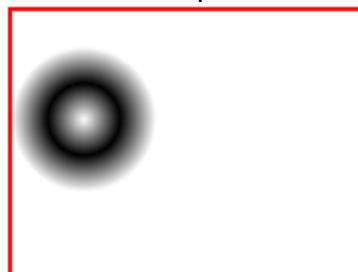
div.tres | closest-corner



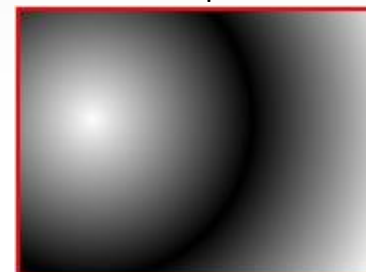
div.quatro | farthest-corner



div.cinco | contain



div.seis | cover



div.sete | 100px 60px

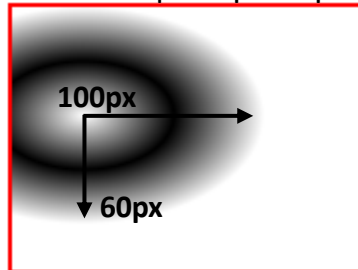
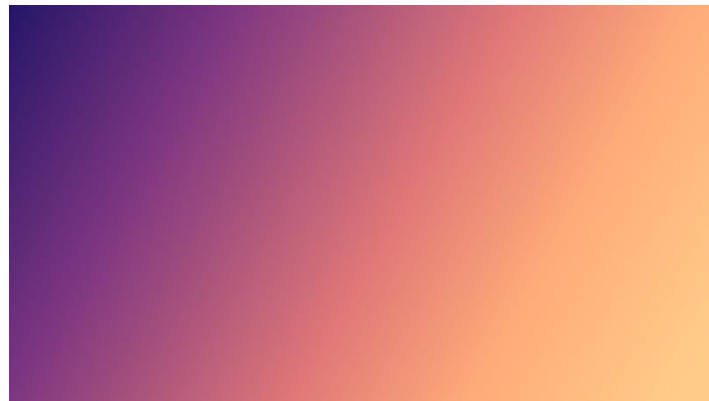


Figura 2 – Efeito do parâmetro tamanho 60

- **Gradientes**

- **Gradiente Repetido**

- *Existe uma funcionalidade para **criar gradientes** que permite definir um **gradiente** e fazer com que ele se **repita indefinidamente***
 - *A **repetição** de **gradientes** só é possível **naqueles** para os quais se definiu o **parâmetro stop**, que pode ser aplicado tanto a **gradientes lineares** como a **gradientes radiais***



- **Gradientes**

- **Gradiente Repetido**

- **Exemplo:**

- Aplicação de **gradiente repetido** em vários boxes
 - Nos boxes **div#um** e **div#tres**, aplicou-se um **gradiente simples**, e nos **div#dois** e **div#quatro**, o mesmo gradiente com repetição
 - Nos **div#cinco** e **div#seis**, aplicaram-se **gradientes** com **repetição linear** e **radial**

```
.um { background-image: linear-gradient(to right, black,  
                                         white 10px, black 20px);  
}
```

- **Gradientes**
 - **Gradiente Repetido**
 - **Exemplo:**

```
.dois {background-image: repeating-linear-gradient(to right, black,  
                                                    white 10px, black 20px);  
}  
.tres {background-image: radial-gradient(50% 50% ellipse, black, white  
                                         10px, black 20px);  
}  
.quatro {background-image: repeating-radial-gradient(50% 50% ellipse,  
                                                     black, white 10px, black 20px);  
}
```

- **Gradientes**
 - **Gradiente Repetido**
 - **Exemplo:**

```
.cinco {background-image: repeating-linear-gradient(-45deg, black,  
white 2px, black 5px, white 20px);  
}  
.seis {background-image: repeating-radial-gradient(100% 50% ellipse,  
black, white 10px, black 20px);  
}
```


- **Gradientes**

- **Gradiente Repetido**

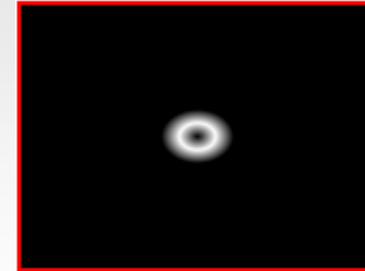
div.um
(simples)



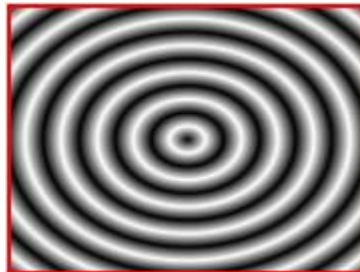
div.dois
(div#um repetido)



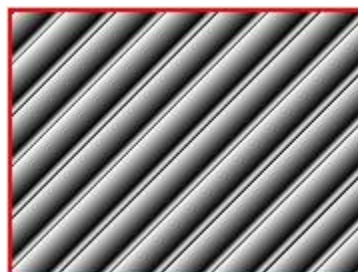
div.tres
(simples)



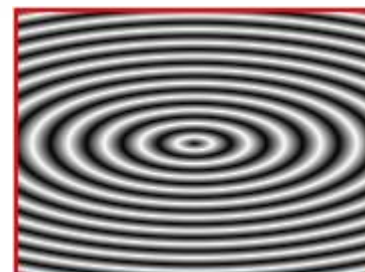
div.quatro
(div#tres repetido)



div.cinco
(45deg repetido)



div.seis
(ellipse repetida)



- **Propriedade Background**

- A **propriedade CSS background** é a forma **abreviada** de **declarar várias propriedades destinadas a estilizar o fundo de um box**
- **background-color**: **define uma cor de fundo para o box**
- **background-image**: **define uma imagem de fundo para o box**
- **background-repeat**: **define como se dá a repetição da imagem de fundo**
- **background-attachment**: **define se a imagem permanece fixa ou rola com o conteúdo do box**

- **Propriedade Background**
 - *background-position*: define a **posição** da **imagem** no **box**
 - *background-clip*: define a **área** do **box** na qual será **aplicada** a **imagem**
 - *background-origin*: define a **posição** de **origem** da **imagem** no **box**
 - *background-size*: define as **dimensões** da **imagem** no **box**
 - *background*: **forma abreviada** de **declarar uma** ou **mais propriedades** descrita anteriormente

- ***Propriedade Background***

- ***background-color***

- *Define uma cor de fundo para o box*
 - *Os valores possíveis para essa propriedade são: valor de cor CSS, como #f00, red e rgb(255, 0, 0)*
 - *O valor padrão é a cor transparent*

```
background-color: #ffc9d4;  
background-color: blue;  
background-color: rgba(255, 120, 150, 0.7);  
background-color: hsla(0, 100%, 50%, 0.6);  
background-color: rgb(20%, 40%, 38%);
```

- **Propriedade Background**

- **background-image**

- Define uma **imagem** de **fundo** para o **box**
 - Observe que usamos **aspas duplas** no endereço da imagem. Alternativamente podemos usar **aspas simples** ou simplesmente **não usar aspas**

```
background-image: url("/imagens/imagem.jpg");  
background-image: linear-gradient(...);  
background-image: radial-gradient(...);
```

- **Propriedade Background**

- **background-repeat**

- Define como a **imagem** de **fundo** se repete no **box**
 - **background-repeat: repeat;** - a **imagem repete-se** na **horizontal** e na **vertical** (valor padrão)
 - **background-repeat: no-repeat;** - a **imagem não** se **repete**
 - **background-repeat: repeat-x;** - a **imagem repete-se** na **horizontal**

- **Propriedade Background**

- **background-repeat**

- **background-repeat: repeat-y;** - a imagem **repete-se** na **vertical**
 - **background-repeat: space;** - a imagem **toca** as **quatro bordas internas** do **box**, e são **espaçadas** de modo a se **distribuírem igualmente**
 - **background-repeat: round;** - a imagem **toca** as **quatro bordas internas** do **box**, e são **redimensionadas** de modo a **preencherem o fundo tocando umas nas outras**

- **Propriedade Background**

- **background-attachment**

- Define o comportamento, se **fixo** ou **não**, da **imagem** de **fundo** no **box**
 - **background-attachment: scroll**; - a **imagem não** permanece **fixa** em relação à **viewport** e **rola** com o **conteúdo** (valor padrão)
 - **background-attachment: fixed**; - a **imagem** permanece **fixa** em relação à **viewport** e **não rola** com o **conteúdo**
 - **background-attachment: local**; - a **imagem** permanece **fixa** em relação à **viewport**, todavia, quando aplicada ao **box** de um **elemento**, **rola** com o **conteúdo** se o **elemento** tiver um mecanismo de **rolagem** definido por **overflow: scroll**

- **Propriedade Background**

- **background-position**

- Define as **coordenadas x** e **y** da **imagem** de **fundo**
 - Valores de **medida CSS** de **comprimento**, **porcentagem** ou as **palavras-chave top, right, bottom e left**
 - O valor padrão é **0 0** ou **left top**
 - Pode-se declarar apenas **um valor** para a **coordenada**, que será considerada a **coordenada horizontal**, e, nesse caso, a **coordenada vertical** será considerada **center**

- **Propriedade Background**

- **background-position**

- O ponto de referência, tanto para o início da contagem das coordenadas no box como na imagem a ser inserida como fundo, é o canto superior esquerdo do box e da imagem
 - Se as coordenadas forem definidas com uso de porcentagem, o ponto de referência na imagem é aquele cujas coordenadas na imagem são iguais às porcentagens declaradas
 - Exemplo: 50% 50%, a referência na imagem para posicioná-la é o centro dela mesma

- ***Propriedade Background***
 - ***background-position***

```
background-position: 40px 50px;  
background-position: 100px;  
background-position: 40% 20%;  
background-position: right top;  
background-position: center;
```

- **Propriedade Background**

- **background-clip**

- Define a área de preenchimento da imagem de fundo no box
 - **background-clip: border-box;** - a **imagem ocupa** até a **área** das **bordas** do **box**, inclusive, se **houver uma** (padrão)
 - **background-clip: padding-box;** - a **imagem ocupa** até a **área** de **padding** do **box**, inclusive, se **houver uma** e **não** ocupa a **área** de **bordas**
 - **background-clip: content-box;** - a **imagem ocupa** até a **área** de **conteúdo** do **boxe** **não** ocupa as **áreas** de **padding** e **bordas** se **houver**

- **Propriedade Background**

- **background-origin**

- Define a **origem** de **preenchimento** da **imagem** de **fundo** do **box**
 - **background-origin: border-box;** - a **imagem** tem por **origem** o **canto superior esquerdo** do **box**, considerando as **bordas** do **box**, se houver (padrão)
 - **background-origin: padding-box;** - a **imagem** tem por **origem** o **canto superior esquerdo** do **box** sem considerar as **bordas** do **box**, se houver
 - **background-origin: content-box;** - a **imagem** tem por **origem** o **canto superior esquerdo** do **box**, sem considerar o **padding** e as **bordas** do **box**, se houver

- **Propriedade Background**

- **background-size**

- Define as **dimensões** da **imagem** de **fundo** do **box**
 - Os **valores possíveis** são uma **medida** de **comprimento CSS**, a **porcentagem** e as **palavras-chave**: **auto**, **contain** e **cover**, sendo **auto** o **valor padrão** (dimensões originais da imagem)
 - **background-size**: **150px 75px**; - a **imagem** terá as **dimensões** de **150px** versus **75px**
 - **background-size**: **250px**; - a **imagem** terá **largura** igual a **250px** e uma **altura** tal que o **aspect ratio** (relação entre **largura** e **altura**) seja **presentado**

- **Propriedade Background**

- **background-size**

- **background-size: 80% 10%;** - a **imagem** terá as **dimensões** de **80%** da **largura** do **box** e **10%** da **altura** do **box**
 - **background-size: 60%;** - a **imagem** terá **largura** igual a **60%** da **largura** do **box** e uma **altura** tal que o **aspect ratio** (relação entre **largura** e **altura**) seja **preservado**
 - **background-size: auto;** - a **imagem** terá suas **dimensões originais** (valor padrão)
 - **background-size: contain;** - a **imagem** terá as **dimensões** de modo que sua **maior dimensão ocupe toda a extensão** do **box** e a **menor dimensão** seja tal que o **aspect ratio** seja **preservado**

- **Propriedade Background**

- **background-size**

- **background-size: cover;** - a **imagem** terá as **dimensões** de modo que sua **menor dimensão ocupe toda a extensão do box** e a **maior dimensão** seja tal que o **aspect ratio** (relação entre **largura** e **altura**) seja **preservado**

- **Propriedade Background**

- **background**

- É a maneira **abreviada** de **declarar todas as oito propriedades** para **definição** de **fundo**
 - **Sintaxe:**

```
seletor {  
  background: background-image background-position /  
              background-size background-repeat  
  background-attachment: background-origin background-clip  
                        background-color;  
}
```

- **Propriedade Background**

- **Múltiplas Imagens de Fundo**

- Podemos definir **quantas imagens de fundo** quisermos em um **mesmo box**, **posicionando e dimensionando** cada uma delas **individualmente**
 - **Cada uma** das **imagens de fundo** definidas para um **mesmo box** **cria** um **layer** com uma **coordenada z** (similar ao z-index), **definindo sua visibilidade**
 - Os **layers** são **posicionados** em uma **pilha** de **layers** colocados uns **sobre outros** fazendo com que os **layers** com **maior coordenada z** **se sobreponham** àqueles com **menor coordenada z**

- ***Propriedade Background***

- ***Múltiplas Imagens de Fundo***

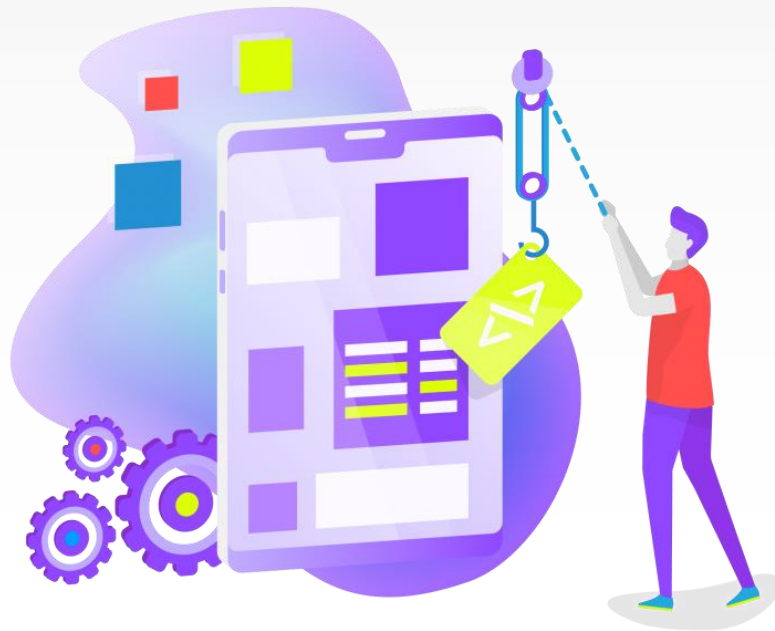
- A **ordem** de **empilhamento**, ou seja, a **coordenada z** é definida pela **ordem** em que a **imagem** é declarada dentro da regra CSS
 - A **primeira imagem** declarada recebe a maior **coordenada z**, a **segunda** recebe a **segunda maior coordenada**, e assim por diante, até a **última imagem declarada** que recebe a **menor coordenada z**
 - A **sintaxe** para **declarar múltiplas imagens de fundo** consiste em **separar os valores das propriedades** com uma **vírgula**

- ***Propriedade Background***

- ***Múltiplas Imagens de Fundo***

- ***Exemplo de regra CSS para definir três imagens como fundo de um elemento div***

```
div {  
  background-image: url(triangulo.png), url(circulo.png), url(retangulo.png);  
  background-repeat: repeat-x, no-repeat, repeat-y;  
  background-position: right 120px, center, 0 0;  
  background-color: #f0c;  
}
```



EXERCÍCIOS

Referências

Silva, M. S. Fundamentos de HTML5 e CSS3. 1. ed. São Paulo: Novatec, 2015.

Duckett, J. HTML e CSS Projete e Construa Websites. 1. ed. Rio de Janeiro: Alta Books, 2011.

Hyslop, B., Castro, E. HTML and CSS: Visual Quickstart Guide. 8. ed. Barcelona: Peachpit Press, 2013.

