

# Design de Sistemas Web

CSS - BÁSICO

---

VINÍCIUS OLIVEIRA

# Unidades de Comprimento

---

A linguagem CSS tem várias unidades diferentes para expressar um comprimento.

Muitas propriedades do CSS recebem valores de "comprimento", como largura (width), margem (margin), preenchimento (padding), tamanho da fonte (font-size) e etc.

O comprimento é um número seguido por uma unidade de comprimento, como 10px, 2em e etc.

Um espaço em branco não pode aparecer entre o número e a unidade de medida. No entanto, se o valor for 0, a unidade de medida pode ser omitida.

Para algumas propriedades do CSS, comprimentos negativos são permitidos.

Existem dois tipos de unidades de comprimento: absoluto e relativo.

# Unidades de Comprimento - Absoluto

---

As unidades de comprimento absoluto são valores fixos e o comprimento é apresentado em tela exatamente com o tamanho definido.

Unidades de comprimento absoluto não são recomendadas para uso na tela, porque os tamanhos de tela podem variar muito.

No entanto, elas podem ser utilizadas se o meio de saída for conhecido, como no layout de uma impressão.

# Unidades de Comprimento - Absoluto

---

As unidades de comprimento absoluto disponíveis em CSS são:

Unidade	Descrição	Conversão
cm	centímetros	
mm	milímetros	
in	polegadas (inches)	1 in = 96 px = 2.54 cm
px	pixels	1 px = 1/96 th of 1 in
pt	pontos	1 pt = 1/72 de 1 in
pc	paicas	1 pc = 12 pt

# Unidades de Comprimento - Relativo

---

Unidades de comprimento relativo especificam um comprimento em relação a outra propriedade de comprimento.

# Unidades de Comprimento - Relativo

---

As unidades de comprimento relativo disponíveis em CSS são:

Unidade	Descrição
em	Relativo ao tamanho da fonte do elemento (2em significa 2 vezes o tamanho da fonte atual)
ex	Relativo à x-altura da fonte atual (raramente utilizado)
ch	Relativo à largura do "0" (zero)
rem	Relativo ao tamanho da fonte do elemento raiz
vw	Relativo a 1% da largura da janela de visualização
vh	Relativo a 1% da altura da janela de visualização
vmin	Relativo a 1% da dimensão menor da janela de visualização
vmax	Relativo a 1% da dimensão maior da janela de visualização
%	Relativo ao elemento pai

# Unidades de Comprimento - Relativo

---

Janela de Exibição = o tamanho da janela do navegador.

Se a janela de visualização tiver 50 cm de largura,  $1vw = 0,5 \text{ cm}$ .

As unidades em e rem são práticas na criação de layout perfeitamente escalável!

Em engenharia de software, escalabilidade é uma característica desejável em todo o sistema, em uma rede ou em um processo, que indica sua habilidade de manipular uma porção crescente de trabalho de forma uniforme, ou estar preparado para crescer.

# Cores

---

As cores em CSS podem ser especificadas das seguintes maneiras:

- um nome de cor predefinido válido em inglês - como “red”,
- um valor Hexadecimal - como “#ff0000”,
- um valor RGB - como “rgb (255,0,0)”,
- um valor RGBA - como “rgba (255,0,0,0.3)”
- um valor HSL - como “hsl(120, 100%, 50%)”,
- um valor HSLA - como “hsla(120, 100%, 75%, 0.3)”,



# Cores – Nome de cor predefinido

---

Nomes de 140 cores em inglês são predefinidos na especificação de cores para linguagens HTML e CSS.

Seguem **alguns** exemplos de cores em inglês que podem ser utilizadas:

Aqua	DeepPink	LightCoral
Black	Fuchsia	Lime
Blue	GhostWhite	Olive
Brown	Gold	Purple
Chocolate	HotPink	Silver
Coral	IndianRed	Snow
DarkOrange	Indigo	Tomato
BlueViolet	Ivory	YellowGreen

# Cores – Nome de cor predefinido

---

Exemplo de definição de diferentes cores com nomes predefinidos em inglês:

```
#p1 {background-color: "gold";}
```

```
#p2 {background-color: "pink";}
```

```
#p3 {background-color: "chocolate";}
```

# Cores – Nome de cor predefinido

---

Veja a tabela com os 140 nomes de cores predefinidos em inglês em :

[https://www.w3schools.com/colors/colors\\_names.asp](https://www.w3schools.com/colors/colors_names.asp)

# Cores – Hexadecimal

---

Uma cor hexadecimal é especificada com: #RRGGBB, onde os inteiros hexadecimais RR (vermelho), GG (verde) e BB (azul) especificam os componentes da cor.

Todos os valores devem estar entre 00 e FF.

Por exemplo, o valor hexadecimal #0000FF é renderizado como azul, porque a cor azul é definida com o seu valor mais alto (FF) e as outras cores são definidas com o valor 00.

As cores são passadas em notação Hexadecimal, sendo 00 o menor valor e FF (255 em decimal) o maior valor.

# Cores – Hexadecimal

---

Exemplo de definição de diferentes cores em hexadecimal:

```
#p1 {background-color: #ff0000;} /* vermelho (red) */
```

```
#p2 {background-color: #00ff00;} /* verde (green) */
```

```
#p3 {background-color: #0000ff;} /* azul (blue) */
```

# Cores – Hexadecimal

---

Veja a tabela com as cores nomeadas classificadas por valor hexadecimal em:

[https://www.w3schools.com/colors/colors\\_hex.asp](https://www.w3schools.com/colors/colors_hex.asp)

# Cores – RGB

---

Um valor de cor RGB é especificado com a função `rgb()`, que possui a seguinte sintaxe:

`rgb(red, green, blue)` - em português: `rgb (vermelho, verde, azul)`

Cada parâmetro (vermelho, verde e azul) define a intensidade da cor e pode ser um número inteiro entre 0 e 255 ou um valor percentual (de 0% a 100%).

Por exemplo, o valor `rgb (0,255,0)` é renderizado como verde, porque o parâmetro `green` é definido com o seu valor mais alto (255) e os outros são definidos com o valor 0.

Os valores a seguir definem cores iguais: `rgb(0,0,255)` e `rgb(0%, 0%, 100%)`.

# Cores – RGB

---

Exemplo de definição de diferentes cores em RGB:

```
#p1 {background-color: rgb(255, 0, 0);} /* vermelho (red) */
```

```
#p2 {background-color: rgb(0, 255, 0);} /* verde (green) */
```

```
#p3 {background-color: rgb(0, 0, 255);} /* azul (blue) */
```



# Cores – RGBA

---

Os valores de cores RGBA são uma extensão dos valores de cores RGB com um canal alfa, que especifica a opacidade do objeto.

Uma cor RGBA é especificada com a função `rgba()`, que possui a seguinte sintaxe:

`rgba(red, green, blue, alpha)` - em português: `rgba(vermelho, verde, azul, alfa)`

O parâmetro alfa é um número entre 0,0 (totalmente transparente) e 1,0 (totalmente opaco).

# Cores – RGBA

---

Exemplo de definição de diferentes cores em RGB com opacidade:

```
#p1 {background-color: rgba(255, 0, 0, 0.3);} /* vermelho com opacidade */
```

```
#p2 {background-color: rgba(0, 255, 0, 0.3);} /* verde com opacidade */
```

```
#p3 {background-color: rgba(0, 0, 255, 0.3);} /* azul com opacidade */
```

# Cores – HSL

---

HSL significa matiz, saturação e luminosidade e representa uma representação de cores de coordenadas cilíndricas.

Um valor de cor HSL é especificado com a função `hsl()`, que possui a seguinte sintaxe:

`hsl(hue, saturation, lightness)` - em português `hsl (matiz, saturação, luminosidade)`

Matiz é um grau na roda de cores (de 0 a 360) - 0 (ou 360) é vermelho, 120 é verde, 240 é azul.

Saturação é um valor percentual; 0% significa um tom de cinza e 100% é a cor total.

Luminosidade também é uma porcentagem; 0% é preto, 100% é branco.

# Cores – HSL

---

Exemplo de definição de diferentes cores em HSL:

```
#p1 {background-color: hsl(120, 100%, 50%);} /* verde */
```

```
#p2 {background-color: hsl(120, 100%, 75%);} /* verde claro */
```

```
#p3 {background-color: hsl(120, 100%, 25%);} /* verde escuro */
```

```
#p4 {background-color: hsl(120, 60%, 70%);} /* verde pastel */
```

# Cores – HSLA

---

Os valores de cores do HSLA são uma extensão dos valores de cores do HSL com um canal alfa que especifica a opacidade do objeto.

Um valor de cor do HSLA é especificado com a função `hsla()`, que possui a seguinte sintaxe:

`hsla(hue, saturation, lightness, alpha)` - `hsla` (matiz, saturação, luminosidade, alfa)

O parâmetro alfa é um número entre 0,0 (totalmente transparente) e 1,0 (totalmente opaco).

# Cores – HSLA

---

Exemplo de definição de diferentes cores em HSL com opacidade:

```
#p1 {background-color: hsla(120, 100%, 50%, 0.3);} /* verde com opacidade */  
#p2 {background-color: hsla(120, 100%, 75%, 0.3);} /* verde claro com  
opacidade */  
#p3 {background-color: hsla(120, 100%, 25%, 0.3);} /* verde escuro com  
opacidade */  
#p4 {background-color: hsla(120, 60%, 70%, 0.3);} /* verde pastel com  
opacidade */
```

# Família da fonte

---

A propriedade *font-family* define a fonte para um elemento.

A propriedade *font-family* pode suportar várias fontes como valor, ou seja, a propriedade terá uma espécie de reserva de fontes para o sistema, dessa forma, se o browser não suportar a primeira fonte definida ele tentará utilizar a próxima fonte.

Ex:

```
h1 {  
    font-family: "Arial", "Crimson Text", "Times New Roman";  
}
```

# Família da fonte

---

Existem dois tipos de nomes de família:

- family-name (nome da família): O nome da família da fonte são os nomes da fonte em si, como já estamos acostumados a utilizar em editores de texto.
- generic-family (família genérica): A família genérica é um grupo de fontes com um estilo em comum.



# Família da fonte – Nome da Família

---

family-name (nome da família): O nome da família da fonte são os nomes da fonte em si, como já estamos acostumados a utilizar em editores de texto.

O nome da família deverá ser citado na propriedade *font-family* do CSS, ou seja, estará “**entre aspas**” no código.

Quando forem colocadas no estilo *inline*, o nome da família deverá ser utilizado entre ‘**apóstrofes**’ no código.

Recomenda-se que o nome da família da fonte esteja no início da propriedade.

Exemplos de nome de família de fontes são “Arial”, “Calibri”, “Times New Roman”, etc.

# Família da fonte – Família Genérica

---

generic-family (família genérica): A família genérica é um grupo de fontes com um determinado estilo em comum.

As famílias genéricas são palavras-chave do CSS e não devem ser citadas na propriedade *font-family*, ou seja, sempre deverão estar **sem aspas** no código.

Uma família de fontes genérica deve ser a última opção na lista de nomes de famílias de fontes, tornando-se uma solução alternativa, um meio de preservar uma parte da intenção do autor da folha de estilo no caso de nenhuma das fontes especificadas estar disponível.

# Família da fonte – Família Genérica

---

Os nomes de família de fontes genéricas em CSS são:

- serif: fontes normais com serifas,
- sans-serif: fontes normais sem serifas,
- monospace: fontes com largura fixa,
- cursive: fontes que simulam manuscrito,
- fantasy: fontes decorativas, para títulos, etc.

# Família da fonte – Família Genérica

---

A tabela abaixo mostra os nomes das fontes genéricas e o nome da fonte similar:

Nome genérico	Similar a
serif	Times New Roman
sans-serif	Arial
monospace	Courier New
cursive	Script
fantasy	Comic

# Família da fonte – Família Genérica

---

Exemplos de fontes:

- serif: “Times New Roman”, “Georgia”, “Palatino”,
- sans-serif: “Arial”, “Helvetica”, “Verdana”,
- monospace: “Courier”, “Andale Mono”, “FreeMono”,
- cursive: “Comic Sans”, “Apple Chancery”, “Bradley Hand”,
- fantasy: “Impact”, “Luminari” e “Chalkduster”.

# Família da fonte – Família Genérica

---

**font-family:serif**

**A Serif headline**

Serif paragraph text.

**font-family:sans-serif**

**A Sans-Serif headline**

Sans-Serif paragraph text.

**font-family:monospace**

**A Monospace headline**

Monospace paragraph text.

**font-family:cursive**

**A Cursive headline**

Cursive paragraph text.

**font-family:fantasy**

**A Fantasy headline**

Fantasy paragraph text.

# Família da fonte

---

As fontes são lidas sempre da esquerda para direita, ou seja, a primeira fonte será a mais a esquerda e a última fonte será a mais a direita.

Cada valor (nesse caso a nome da família ou família genérica) deverá ser separado por vírgula.

Relembrando: primeiro recomenda-se utilizarmos os nomes da família e depois utilizamos as famílias genéricas.

Exemplo:

```
body {  
    font-family: "Crimson Text", "Times New Roman", serif;  
}
```

# Família da fonte

---

Exemplos:

```
h1 {  
    font-family: "Times New Roman", Times, serif;  
}
```

```
h2 {  
    font-family: Arial, Helvetica, sans-serif;  
}
```



# Fonte – Tamanho

---

Ser capaz de gerenciar o tamanho do texto é importante no design da web.

No entanto, não é recomendável utilizar ajustes de tamanho de fonte para fazer com que os parágrafos pareçam cabeçalhos, ou cabeçalhos fiquem parecidos com parágrafos.

Nesse caso, utilize as tags HTML adequadas, como `<h1>` - `<h6>` para títulos e `<p>` para parágrafos.

# Fonte – Tamanho

---

A propriedade *font-size* define o tamanho da fonte para um elemento.

O valor do tamanho da fonte pode ser um tamanho absoluto ou relativo.

Nota: Se você não especificar um tamanho de fonte, o tamanho padrão para o texto normal, como parágrafos, será 16px (16px = 1em).

# Fonte – Tamanho

---

O valor do tamanho da fonte pode ser um tamanho absoluto ou relativo.

## **Tamanho absoluto:**

- Define um tamanho específico para o texto.
- Não permite que um usuário altere o tamanho do texto em todos os navegadores (ruim por motivos de acessibilidade).
- O tamanho absoluto é útil quando o tamanho físico da saída é conhecido.

## **Tamanho relativo:**

- Define o tamanho relativo em relação aos elementos.
- Permite que um usuário altere o tamanho do texto nos navegadores.

# Fonte – Tamanho

---

É possível definir o tamanho da fonte no CSS das seguintes maneiras:

Maneira	Descrição
Tamanhos definidos	Tamanho definido pelo CSS através de palavras-chave
Tamanho fixo	Tamanho definido
Percentual	Tamanho definido

# Fonte – Tamanho

---

A linguagem CSS possui tamanhos definidos através de palavras-chave:

Palavra-chave	Descrição
medium	Define o tamanho da fonte para um tamanho médio. Este é o valor padrão.
xx-small	Define o tamanho da fonte para um tamanho xx-pequeno.
x-small	Define o tamanho da fonte para um tamanho extra pequeno.
small	Define o tamanho da fonte para um tamanho pequeno.
large	Define o tamanho da fonte para um tamanho grande.
x-large	Define o tamanho da fonte para um tamanho extra grande.
xx-large	Define o tamanho da fonte para um tamanho xx-grande.
smaller	Define o tamanho da fonte para um tamanho menor que o elemento pai.
larger	Define o tamanho da fonte para um tamanho maior que o elemento pai.

# Fonte – Tamanho

---

A linguagem CSS possui tamanhos definidos através de palavras-chave:

Palavra-chave	Função
medium	Um tamanho de fonte de aproximadamente 10 pontos
xx-small	50% menor que a fonte x-small
x-small	50% menor que a fonte small
small	50% menor que a fonte medium
large	50% maior que a fonte medium
x-large	50% maior que a fonte large
xx-large	50% maior que a fonte x-large
smaller	50% menor que a fonte do elemento pai
larger	50% maior que a fonte do elemento pai

# Fonte – Tamanho

---

Os valores medium, xx-small, x-small, small, large, x-large e xx-large são considerados valores de tamanho absoluto, ou seja, esses valores independem de outros elementos.

Enquanto os valores smaller e larger são considerados valores de tamanho relativo, ou seja, o tamanho depende do elemento pai.

# Fonte – Tamanho

---

Define o tamanho da fonte para um tamanho fixo.

O CSS aceita as seguintes unidades de medida abaixo como tamanho fixo:

Unidade	Descrição	Conversão
cm	centímetro	
mm	milímetros	
in	polegadas (inches)	1 in = 96 px = 2.54 cm
px	pixels	1 px = 1/96 th of 1 in
pt	pontos	1 pt = 1/72 de 1 in
pc	paicas	1 pc = 12 pt



# Fonte - Estilo

---

A propriedade font-style é usada principalmente para especificar um texto em itálico ou oblíquo.

Esta propriedade pode receber três valores diferentes:

- normal - o texto é mostrado normalmente
- italic - o texto é mostrado em itálico
- oblique - o texto é "inclinado" (oblíquo é muito semelhante ao itálico, mas menos suportado)

# Fonte - Estilo

---

O itálico ou obliquo do texto pode ser configurado da seguinte maneira:

```
p {  
    font-style: valor;  
}
```

# Fonte - Estilo

---

Itálico: Um design diferente mas complementar.

*Itálico: Um design diferente mas complementar.*

Itálico: Um design diferente mas complementar.

*Itálico: Um design diferente mas complementar.*

Itálico: Um design diferente mas complementar.

*Itálico: Um design diferente mas complementar.*

Itálico é um tipo de letra com postura diferente de seu familiar básico, o tipo romano. Itálicos apresentam designs complementares, com características únicas e, normalmente, larguras e espaçamentos diferentes do tipo romano. Itálicos desenhados para uma família tipográfica sans serif normalmente se parecem com seu irmão regular, mas apresentam caracteres com desenho diferentes, como o a, g, e f caixa baixa.

Oblíquo: Uma versão inclinada da romana.

*Oblíquo: Uma versão inclinada da romana.*

Oblíquo: Uma versão inclinada da romana.

*Oblíquo: Uma versão inclinada da romana.*

Oblíquo: Uma versão inclinada da romana.

*Oblíquo: Uma versão inclinada da romana.*

Oblíquo, em contrapartida, é simplesmente uma versão inclinada dos tipos romanos, sem apresentar características diferentes, além da postura. Oblíquos são mais comumente encontrados em famílias sem serifa, embora nem toda sans serif tenha uma versão oblíqua.

# Fonte - Estilo

---

Nos exemplos abaixo, o parágrafo é configurado com três estilos diferentes:

```
p {  
  font-style: normal;  
}
```

```
p {  
  font-style: italic;  
}
```

```
p {  
  font-style: oblique;  
}
```

# Fonte - Espessura (Peso)

---

A propriedade font-weight define o peso (a espessura) de um texto (o quanto os caracteres ficarão grossos ou finos na exibição de um texto).

A espessura do texto pode ser configurada da seguinte maneira:

```
p {  
  font-weight : valor;  
}
```

# Fonte - Espessura (Peso)

---

A espessura de um texto pode ser configurada com os seguintes valores:

Valor	Descrição
normal	Define caracteres normais. Este é o padrão.
bold	Define caracteres espessos.
bolder	Define caracteres mais claros.
lighter	Define caracteres mais claros.
número	Deve ser igual um número inteiro de 100 – 900. Quanto menor o número, mais fino ele será e quanto maior o número, mais grosso ele será. O texto normal tem um valor de 400. O texto em negrito tem um valor de 700.

# Fonte - Espessura (Peso)

---

O valor de cada número corresponde aos seguintes pesos:

Valor	Peso
100	Thin
200	Extra Light (Ultra Light)
300	Light
400	Normal
500	Medium
600	Semi Bold (Demi Bold)
700	Bold
800	Extra Bold (Ultra Bold)
900	Black (Heavy)

# Fonte - Espessura (Peso)

---

Nos exemplos abaixo, o parágrafo é configurado com três espessura diferentes:

```
p {  
  font-weight: normal;  
}
```

```
p {  
  font-weight: bold;  
}
```

```
p {  
  font-weight: 900;  
}
```



# Fonte - Variant

---

A propriedade font-variant especifica se um texto deve ou não ser exibido em uma fonte de letras minúsculas (small-caps).

Em uma fonte de letras minúsculas, todas as letras minúsculas são convertidas em letras maiúsculas.

No entanto, as letras maiúsculas convertidas aparecem em um tamanho de fonte menor do que as letras maiúsculas originais no texto.

# Fonte - Variant

---

Pode-se atribuir dois tipos diferentes de valores a propriedade:

- normal - O navegador exibe uma fonte normal. Este é o padrão.
- small-caps - O navegador exibe uma fonte de letras minúsculas.

A propriedade pode ser configurada da seguinte maneira:

```
p {  
  font-weight : valor;  
}
```

# Fonte - Variant

---

Nos exemplos abaixo são configurados os dois valores possíveis para propriedade:

```
p {  
  font-variant: normal;  
}
```

```
p {  
  font-variant: small-caps;  
}
```

Este cabeçalho com letras normais

ESTE PARÁGRAFO COM LETRAS EM "SMALL-CAPS"

# Fonte - Forma abreviada

---

Para encurtar o código, também é possível especificar as propriedades de fonte (exceto cor) em uma única propriedade.

Isso é chamado de propriedade abreviada.

A propriedade abreviada para fontes de forma resumida é simplesmente a palavra reserva *font*.

# Fonte - Forma abreviada

---

É possível declarar todas ou algumas das propriedades. Os valores `size` e `family` são obrigatórios quando se usa a declaração abreviada. Os demais valores são facultativos e se forem omitidos será adotado o valor padrão (inicial) ou o valor herdado do elemento-pai.

Os valores `style`, `variant`, `weight` e `size`, podem ser declarados em qualquer ordem.

# Fonte - Forma abreviada

---

Abaixo é utilizada a propriedade de fonte de forma abreviada:

```
p {  
    font: italic small-caps bold 14px "Comic Sans MS", sans-serif;  
}
```

Nessa propriedade é utilizada uma fonte em formato itálico, padrão small-caps, negrito, tamanho 14px, nome da família "Comic Sans MS" e família genérica sans-serif.

# Background (Plano de fundo)

---

As propriedades de plano de fundo do CSS são usadas para definir os efeitos de segundo plano para os elementos da página.

Propriedades de fundo do CSS:

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

# Background - Cor

---

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

O fundo de um elemento é o tamanho total do elemento, incluindo preenchimento e borda (mas não a margem).

Dica: use uma cor de fundo e uma cor de texto que facilitam a leitura do texto.



# Background - Cor

---

A cor de fundo de uma página é configurada da seguinte maneira:

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

Lembrando que é possível especificar uma cor em CSS de diferentes maneiras como um valor hexadecimal, um valor RGB e um nome de cor em inglês.

Além disso, é possível passar o valor igual a transparente que deixa um plano de fundo com a cor transparente.

# Background - Cor

---

No exemplo abaixo, os elementos `<h1>`, `<p>` e `<div>` têm cores de fundo diferentes, utilizando padrões diferentes para diferentes seletores:

```
h1 {
```

```
  background-color:  
  #ff0000;
```

```
}
```

```
div {
```

```
  background-color:  
  lightblue;
```

```
}
```

```
p {
```

```
  background-color:  
  rgb(0,255,0);
```

```
}
```

# Background - Imagem

---

A propriedade `background-image` define uma ou mais imagens de fundo para um elemento.

Por padrão, uma imagem de plano de fundo é colocada no canto superior esquerdo de um elemento e repetida verticalmente e horizontalmente.

Dica: O plano de fundo de um elemento é o tamanho total do elemento, incluindo preenchimento e borda (mas não a margem).

Dica: Sempre defina uma cor de segundo plano a ser usada se a imagem não estiver disponível.

# Background - Imagem

---

A imagem de fundo de uma página pode ser configurada da seguinte maneira:

```
body {  
    background-image: valor;  
}
```

# Background - Imagem

---

Pode-se atribuir um dos valores abaixo a propriedade:

Valor	Descrição
<code>url('URL')</code>	O URL para a imagem. Para especificar mais de uma imagem, separe as URLs com uma vírgula.
<code>none</code>	Nenhuma imagem de fundo será exibida. Este é o padrão.
<code>linear-gradient()</code>	Define um gradiente linear como imagem de fundo. Defina pelo menos duas cores (de cima para baixo).
<code>radial-gradient()</code>	Define um gradiente radial como a imagem de fundo. Defina pelo menos duas cores (do centro para as bordas).
<code>repeating-linear-gradient()</code>	Repete um gradiente linear.
<code>repeating-radial-gradient()</code>	Repete um gradiente radial.

# Background - Imagem

---

Os valores da propriedade devem ser separados por vírgula (imagens, cores...).

A imagem da URL pode ser um arquivo nos formatos PNG, SVG, JPG, GIF, WEBP ou outros possíveis formatos.

As cores utilizadas para os gradientes podem ser especificadas de diferentes maneiras em CSS como um valor hexadecimal, um valor RGB e um nome de cor em inglês.

# Background - Image

---

Seguem exemplos de especificação de imagem/gradientes como plano de fundo em CSS:

```
body {  
    background-image:  
    url("imagem.jpg");  
}
```

```
body {  
    background-image:  
    radial-gradient(red,  
    yellow, green);  
}
```

```
body {  
    background-image:  
    repeating-linear-gradient  
    (red, yellow 10%, green  
    20%);  
}
```

# Background - Repeat

---

A propriedade background-repeat define se/como uma imagem de fundo será repetida.

Por padrão, uma imagem de fundo é repetida de forma vertical e horizontal.

```
body {  
    background-image: url("endereço imagem.extensão");  
    background-repeat: valor;  
}
```



# Background - Repeat

---

Pode-se atribuir um dos quatros valores abaixo a esta propriedade:

Valor	Descrição
repeat	A imagem de fundo é repetida na vertical e na horizontal. A última imagem será cortada se não couber. Este é o padrão.
repeat-x	A imagem de fundo é repetida apenas horizontalmente.
repeat-y	A imagem de fundo é repetida apenas verticalmente.
no-repeat	A imagem de fundo não é repetida. A imagem será mostrada apenas uma vez.
space	A imagem de fundo é repetida o máximo possível sem recorte. A primeira e a última imagem são fixadas em cada lado do elemento e o espaço em branco é distribuído uniformemente entre as imagens.
round	A imagem de fundo é repetida e comprimida ou esticada para preencher o espaço (sem lacunas).

# Background - Repeat

---

No exemplo abaixo, temos a propriedade de repetição de imagem de plano fundo sendo aplicada de diferentes formas:

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

```
body {  
    background-image:  
    url("imagem.jpg");  
    background-repeat: repeat-x;  
}
```

```
body {  
    background-image:  
    url("imagem.jpg");  
    background-repeat: round;  
}
```

# Background - Attachment

---

A propriedade `background-attachment` define se uma imagem de plano de fundo rola com o resto da página ou se é fixa.

O comportamento da figura ao rolar a página pode ser configurado da seguinte maneira:

```
body {  
    background-image: url("endereço imagem.extensão");  
    background-attachment: valor;  
}
```

# Background - Attachment

---

Pode-se atribuir um dos valores abaixo a propriedade:

Valor	Descrição
scroll	A imagem de fundo rolará com a página. Este é o padrão.
fixed	A imagem de fundo não rolará com a página.
local	A imagem de fundo rolará com o conteúdo do elemento.

# Background - Attachment

---

No exemplo abaixo, temos a propriedade de comportamento de rolagem da imagem de plano fundo sendo aplicada de diferentes formas:

<pre>body {   background-image:   url("imagem.jpg");    background-attachment: scroll; }</pre>	<pre>body {   background-image:   url("imagem.jpg");    background-attachment: fixed; }</pre>	<pre>body {   background-image:   url("imagem.jpg");    background-attachment: local; }</pre>
--	---	---

# Background - Posição

---

A propriedade `background-position` define a posição inicial de uma imagem de plano de fundo.

Dica: Por padrão, uma imagem de fundo é colocada no canto superior esquerdo de um elemento e repetida de forma vertical e horizontal.

O comportamento da figura ao rolar a página pode ser configurado da seguinte maneira:

```
body {  
    background-image: url("endereço imagem.extensão");  
    background-position: valor;  
}
```

# Background - Posição

---

Pode-se atribuir um dos valores abaixo a propriedade:

Valor	Descrição
Palavra-chave	São utilizadas palavras chaves para o alinhamento horizontal e vertical da imagem.
X% y%	O primeiro valor é a posição horizontal e o segundo valor é a posição vertical. O canto superior esquerdo é 0% 0%. O canto inferior direito é 100% 100%. Se você especificar apenas um valor, o outro valor será 50%. O valor padrão é: 0% 0%.
xpos ypos	O primeiro valor é a posição horizontal e o segundo valor é a vertical. O canto superior esquerdo é 0 0. Unidades podem ser em pixels (0px 0px) ou quaisquer outras unidades aceitas pelo CSS. Se você especificar apenas um valor, o outro valor será 50%. Você pode misturar % e posições.

# Background - Posição

---

Pode-se utilizar as seguintes palavras-chave abaixo para o alinhamento da imagem:

Palavra-chave	Alinhamento
left top	Alinhamento Horizontal esquerdo e alinhamento Vertical parte superior.
left center	Alinhamento Horizontal esquerdo e alinhamento Vertical parte central.
left bottom	Alinhamento Horizontal esquerdo e alinhamento Vertical parte inferior.
right top	Alinhamento Horizontal direito e alinhamento Vertical parte superior.
right center	Alinhamento Horizontal direito e alinhamento Vertical parte central.
right bottom	Alinhamento Horizontal direito e alinhamento Vertical parte inferior.
center top	Alinhamento Horizontal centro e alinhamento Vertical parte superior.
center center	Alinhamento Horizontal centro e alinhamento Vertical parte central.
center bottom	Alinhamento Horizontal centro e alinhamento Vertical parte inferior.



# Background - Posição

---

No exemplo abaixo, temos a propriedade de comportamento de rolagem da imagem de plano fundo sendo aplicada de diferentes formas:

```
body {  
    background-image:  
    url("imagem.jpg");  
  
    background-position: bottom  
    right;  
}
```

```
body {  
    background-image:  
    url("imagem.jpg");  
  
    background-position: 50% 50%;  
}
```

```
body {  
    background-image:  
    url("imagem.jpg");  
  
    background-position: 50px 150px;  
}
```

# Background - Forma abreviada

---

Para encurtar o código, também é possível especificar todas as propriedades de segundo plano em uma única propriedade.

Isso é chamado de propriedade abreviada.

A propriedade abreviada para background de forma resumida é simplesmente a palavra reserva *background*.

Como valor da propriedade abreviada deverão ser usados os valores das propriedades apresentadas para o background.

# Background - Forma abreviada

---

Quando for utilizada a propriedade abreviada, a ordem em que os valores da propriedade devem ser utilizados são:

1. background-color
2. background-image
3. background-repeat
4. background-attachment
5. background-position

Não importa se um dos valores da propriedade está faltando, desde que os outros estejam nessa ordem.

# Background - Forma abreviada

---

Abaixo é utilizada a propriedade de background de forma abreviada:

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

Nesse plano de fundo é utilizada a cor de fundo #ffffff, a imagem "imagem.jpg", sem repetir a imagem com a imagem alinhada a direita na parte superior da página.