

## Tema 6: CSS (Cascating Styles Sheets - Folhas de Estilo em Cascata)

### O que são CSS

A CSS (Folhas de Estilo em Cascata) é uma linguagem usada para **estilizar documentos de forma leve e elegante**, ela possibilita determinar o comportamento de um determinado documento em telas de diferentes dispositivos. Ela é altamente compatível com o HTML, fazendo parte de praticamente todas as páginas da web, fazemos o uso das css para estilizar os documentos HTML, visando sempre a rapidez e eficácia na visualização dos elementos. As CSS **permitem aplicar estilos de modo seletivo em elementos de documentos HTML**.

Por exemplo, para selecionar todas as tags (elementos) título de uma página e fazer qualquer tipo de alteração e estilização, com as CSS não há necessidade de modificar elemento individualmente (como era feito antigamente), posso simplesmente determinar no código CSS **que os títulos terão um determinado comportamento**, e a alteração é feita em todos. Para que a página em HTML possa fazer uso dos estilos em CSS, é necessário inserir na tag <head> **um link para a página em CSS onde estão configurados os estilos**. Veja no exemplo:

```
<html>
<head>
  <title>Formulários</title>
  <link rel="stylesheet" type="text/css" href="estilo.css">
</head>
<body>
  <!-- Conteúdo aqui-->
</body>
</html>
```

Podemos notar na linha 4 da imagem acima, indicando a inserção da página estilo.css no atributo href, fazendo assim o link para que essa página seja usada esses estilos. O link é uma **tag utilizada para indicar ao html que uma página externa será adicionada, nesse caso usamos o essa tag <link rel="stylesheet"> indicando que esse link é relativo às folhas de estilos do tipo (type) text/css. E por último fazemos o uso do href=" nome\_da\_pagina.css", para indicar o local da página do estilo CSS**.

Vamos ver um exemplo simples do uso do html na tag <p> do html:

```
p {
  color: blue;
}
```

Podemos notar nesse exemplo onde o elemento <p> foi configurado para que **tenha a cor blue(azul)** em todos os elementos que a página está linkada. Toda estrutura é chamada de conjunto de regras (em geral usamos o termo “regra”, por ser mais curto). Vamos conhecer a anatomia do conjunto de regras do CSS.

**SELETOR (SELECTOR)** – O nome do elemento (tag) em HTML no começo do conjunto de regras. Ele é responsável por selecionar o(s) elemento(s) a serem aplicados os estilos (nesse caso, os elementos <p>). Para dar estilo a um outro elemento, é só mudar o seletor.

**DECLARAÇÃO (DECLARATION)** – Uma regra simples como `color: red;` especifica quais das propriedades do elemento deseja estilizar. Nesse caso irá mudar a cor do elemento para vermelho.

**PROPRIEDADES (PROPERTY)** – Forma pela qual usamos para estilizar um elemento em HTML. Nesse caso, `color` é uma propriedade dos elementos `<p>`. Em CSS, podemos escolher quais propriedades queremos afetar com sua regra.

**VALOR DA PROPRIEDADE (PROPERTY VALUE)** – Com o valor da propriedade podemos escolher uma, dentre muitas aparências possíveis para uma determinada propriedade (cada propriedade em particular, possui inúmeros valores de distintos).

`p` = Seletor;  
`color: red` = Declaração;  
`color` = Propriedades;  
`red` = valor da propriedade.

Acima nota-se a estrutura do conjunto de regras que se faz necessário para a aplicação do CSS, temos primeiro que indicar o seletor (nesse caso o elemento `<p>`), para definir as propriedades desse seletor sempre devemos definir a propriedade entre chaves ( `{ }` ), depois a declaração da propriedade seguida de dois pontos ( `:` ), e atribuindo o valor da propriedade, finalizando a linha de cada declaração com o ponto e vírgula ( `;` ). Devemos lembrar que essa regra deve ser usada em todos os elementos que serão estilizados com o CSS, mudando algumas definições pontualmente.

Em casos que é necessário a estilização de mais de uma propriedade, é só definir na linha de baixo uma nova propriedade, declarando o valor dela. Veja o exemplo abaixo:

```
p {  
    color: red;  
    width: 500px;  
    border: 1px solid black;  
}
```

Este código define a cor do texto como vermelho, a largura como 500 pixels e uma borda de 1 pixel sólida preta para o elemento especificado.

Podemos perceber que definimos o **estilo do elemento `<p>` com o uso de três propriedades diferentes.**

```
html {  
    background-color: skyblue;  
}  
  
p {  
    color: black;  
    font-family: 'Trebuchet MS', 'Lucida Sans Unicode', 'Lucida Grande', Arial, sans-serif;  
}
```

```
h1 {
  color: teal;
  text-shadow: slategrey;
}
```

Nesse exemplo acima, podemos notar que foram inseridos no CSS **três elementos, o elemento HTML, onde é aplicado em todo o documento HTML, o elemento p e o elemento h1**. Podemos fazer diversas alterações deixando nossas páginas muito mais atrativas visualmente com o CSS. É possível também definir um mesmo conjunto de regras para múltiplos seletores, vamos ver um exemplo:

```
html {
  background-color: skyblue;
}

p, h1, li {
  color: red;
}
```

Na imagem acima definimos **três seletores com o mesmo conjunto de regras**, ou seja, mesma propriedade e valor para os três, apenas separando-os com uma vírgula ( , ). Nesse exemplo determinamos que os elementos <p>, <li> e <h1>, serão apresentados na página em vermelho.

## Diferentes Tipos de Seletores

Existem diversos tipos de seletores diferentes, vamos conhecer os seletores de elementos de um determinado tipo nos documentos HTML. Porém é preciso fazer seleções mais específicas, veja alguns dos tipos mais comuns de seletores.

Nome do Seletor	Tipo	Exemplo
Tag ou Seletor de Tipo	Todos os elementos HTML de determinado tipo.	<code>p`   "Seleciona p"</code>
Seletor de ID	O elemento na página com o ID especificado. É uma boa prática usar um elemento por ID.	<code>#id`   Seleciona &lt;p id="meu-id"&gt;</code>
Seletor de Classe	O(s) elemento(s) na página com a classe especificada. Várias instâncias de classe podem aparecer em uma página.	<code>.class`   Seleciona &lt;p class="minha-classe"&gt;</code>
Seletor de Atributo	O(s) elemento(s) na página com o atributo especificado.	<code>img[src]`   Seleciona &lt;img src="minha_imagem.png"&gt;. Lembrando seleciona o atributo src e não o img</code>
Pseudo-Classe	O(s) elemento(s) especificado(s), mas somente quando estiver no estado especificado, como por exemplo, quando o mouse está sobre ele.	<code>a:hover`   Seleciona &lt;a&gt;, mas somente quando o mouse está em cima do link.</code>

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Exemplo de Seletores CSS</title>
  <style>
```

```

/* Tag ou Seletor de Tipo */
p {
    color: blue;
}

/* Seletor de ID */
#meuld {
    background-color: yellow;
}

/* Seletor de Classe */
.minhaClasse {
    font-weight: bold;
}

/* Seletor de Atributo */
img[src] {
    border: 2px solid green;
}

/* Pseudo-Classe */
a:hover {
    text-decoration: underline;
    color: red; /* Adicionada nova propriedade 'color' */
}
</style>
</head>
<body>

<!-- Tag ou Seletor de Tipo -->
<p>Isso é um parágrafo.</p>

<!-- Seletor de ID -->
<div id="meuld">
    Este é um elemento com ID "meuld".
</div>

<!-- Seletor de Classe -->
<p class="minhaClasse">Este parágrafo tem a classe "minhaClasse".</p>

<!-- Seletor de Atributo -->


<!-- Pseudo-Classe -->
<a href="#">Isso é um link</a>

</body>
</html>

```

## Fontes e Textos

No CSS podemos vincular **fontes em diferentes formas, uma delas é um vínculo online de fonte**, nesse caso usando o google fonts podemos inserir uma ou mais famílias de fontes com esse

método, vamos ver no código:

```
<link rel="stylesheet" href="http://fonts.googleapis.com/css?family=Open+Sans">
```

Nota-se no código acima que vinculamos uma folha de estilo que **baixa a família de fontes Open Sans junto com a página web**, permitindo que seja definido seus elementos HTML usando a própria folha de estilos. No CSS é possível estilizar várias propriedades de um mesmo elemento de texto, vamos conhecer agora esses elementos.

**Na tabela abaixo estão as possibilidades que podemos definir os estilos das fontes**, importante que na hora da prática você faça testes para descobrir qual melhor estilo para seu documento.

PROPRIEDADE	VALOR DA PROPRIEDADE
Font	font-style, font-variant, font-weight, font-size, font-family, caption, icon, menu, message-box, smallcaption, status-bar
Font-size	xx-small, x-small, small, medium, large, x-large, xx-large, smaller, larger, inherit, length (%)
Font-size-adjust	None, inherit, number (%)
Font-stretch	normal, wider, narrower, ultra-condensed, extra-condensed, condensed, semi-condensed, semi-expanded, expanded, extra-expanded, ultra-expanded, inherit
Font-style	normal, italic, oblique, inherit
Font-variant	normal, small-caps, inherit
Font-weight	normal, bold, bolder, lighter, 100, 200, 300, 400, 500, 600, 700, 800, 900, inherit
Font-family	Family-name, generic-family, inherit

A seguir vamos ver a tabela de **propriedades de texto**:

PROPRIEDADE	VALOR DA PROPRIEDADE
direction	ltr, rtl, inherit
hanging-punctuation	None, [start, end, adjacent]
letter-spacing	normal, length (%)
punctuation-trim	None, [start, end, adjacent]
text-align	start, end, left, right, center, justify
text-align-last	start, end, left, right, center, justify
text-decoration	none, underline, overline, line-through, blink
text-emphasis	none, [accent, dot, circle, disc] [before, after, length (%)]
text-indent	Length (%)
text-justify	auto, inter-word, inter-ideograph, inter-cluster, distribute, kashida, Tibetan
text-outline	None, color, length
text-shadow	None, color, length
text-transform	none, capitalize, uppercase, lowercase

text-wrap	normal, unrestricted, none, suppress	
unicode-bidi	normal,    embed,    bidi-override	
white-space	normal, pre, nowrap, pre-wrap, pre-line	
white-space-collapse	preserve, collapse, preserve-breaks, discard	
word-break	normal, keep-all, loose, break-strict, break-all	
word-spacing	normal,    length    (%)	
word-wrap	normal,    break-word	

Na tabela anterior é mostrado a propriedade de texto com seus respectivos valores, dessa forma é possível entender que cada propriedade tem um valor específico com uma ação específica. **Não é necessário que saiba todas, o importante é usá-las de acordo achar melhor em sua folha de estilo, por isso é recomendado que faça testes com cada um desses valores de cada propriedade**, assim é possível visualizar o efeito de cada uma delas.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Exemplo de Propriedades de Texto e Fonte</title>
  <style>
    body {
      font-family: 'Open Sans', sans-serif; /* Exemplo de font-family */
      direction: ltr; /* Exemplo de direction */
      text-align: center; /* Exemplo de text-align */
    }

    h1 {
      font-size: 24px; /* Exemplo de font-size */
      letter-spacing: 2px; /* Exemplo de letter-spacing */
      text-transform: uppercase; /* Exemplo de text-transform */
    }

    p {
      text-decoration: underline; /* Exemplo de text-decoration */
      word-spacing: 5px; /* Exemplo de word-spacing */
    }

    .highlight {
      text-shadow: 2px 2px 2px #333; /* Exemplo de text-shadow */
    }
  </style>
  <link rel="stylesheet" href="http://fonts.googleapis.com/css?family=Open+Sans">
</head>
<body>

  <h1>Exemplo de Propriedades de Texto e Fonte</h1>
```

<p>Este é um parágrafo com <span class="highlight">algumas partes destacadas</span>.</p>

</body>

</html>

## Desenvolvimento e Aplicação

Vamos começar a utilizar o CSS na prática para fixar melhor esse conjunto de regras e seu modo de aplicação.

<!DOCTYPE html>

<html lang="pt-br">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Estilos em CSS</title>

<link rel="stylesheet" href="seuestilo.css"> <!-- Substitua "seuestilo.css" pelo nome do seu arquivo CSS

-->

</head>

<body>

<h1>Aplicando Folhas de Estilo no HTML</h1>

 <!-- Substitua "sua-imagem.png" pelo caminho da sua imagem -->

<p>Assim como o HTML, o CSS não é realmente uma linguagem de programação.

Também não é uma linguagem de marcação - é uma linguagem de folhas de estilos.

Isso significa que o CSS permite aplicar estilos seletivamente a elementos em documentos HTML.</p>

</body>

</html>

## CSS:

### Alterando a cor de fundo:

html {

background-color: #424142; /\* Substitua #424142 pela cor desejada \*/

}

### Alterando o background da página:

body {

width: 600px;

margin: 0 auto;

background-color: #0e4d58; /\* Substitua #0e4d58 pela cor desejada \*/

padding: 0 20px 20px 20px;

border: 3px solid black;

}

Agora para o elemento <body>. Há algumas declarações aqui, então vamos passar por elas uma a

uma:

**width: 600px;** — Isso força o corpo a ter 600 pixels de largura.

**margin: 0 auto;** — Quando você define dois valores em uma propriedade como margin ou padding, o primeiro valor afeta a parte superior do elemento e a parte inferior (tornando-os 0 nesse caso), e no segundo valor os lados esquerdos e direito (aqui, auto é um valor especial que divide o espaço horizontal uniformemente entre esquerda e direita). Você também pode utilizar um, três ou quatro valores, como documentado aqui.

**background-color;** — Como antes, isso define a cor de fundo do elemento.

**padding: 0 20px 20px 20px;** — Temos quatro valores definidos no padding, para criar um pouco de espaço em torno do nosso conteúdo. Dessa vez, estamos definindo sem padding na parte superior do corpo, e 20 pixels no lado esquerdo, na parte inferior e no lado direito. Os valores definem a parte superior, o lado direito, a parte inferior e o lado esquerdo, nessa ordem. Como com a margin, você também pode usar um, dois, ou três valores.

**border: 3px solid black;** — Isso simplesmente define uma borda preta sólida de 3 pixels de largura em todos os lados do corpo.

```
h1 {
    margin: 0;
    padding: 20px 0;
    color: #f08902; /* Substitua #f08902 pela cor desejada */
    text-shadow: 2px 2px 2px black; /* Adicione a cor para o efeito de sombra do texto */
    text-align: center;
    font-family: 'Cambria', 'Cochin', 'Georgia', 'Times', 'Times New Roman', serif;
}
```

**Margin:** Definimos o margin do h1 em 0, para tirar os espaços da margem do texto.

**Padding:** O padding no texto é importante para espaçar dos outros elementos da página.

**Color:** Definimos a cor do texto.

**Text-shadow** — Uma propriedade bastante interessante que usaremos aqui é o text-shadow, que aplica uma sombra ao conteúdo de texto do elemento. Seus quatro valores são os seguintes:

- O primeiro valor em pixel define o deslocamento horizontal da sombra do texto — até onde ele se move: um valor negativo deve movê-la para a esquerda.
- O segundo valor em pixel define o deslocamento vertical da sombra do texto — o quanto ela se move para baixo, neste exemplo; um valor negativo deve movê-la para cima.
- O terceiro valor em pixel define o raio de desfoque da sombra — um valor maior significará uma sombra mais borrada.
- O quarto valor define a cor base da sombra.

**Text-align:** - Essa propriedade centraliza o texto.

**Font-family:** - Definimos uma família de fontes, ele funciona em ordem, nesse caso se a fonte Cambria não estiver instalada no computador ou no servidor (caso o site já esteja hospedado), ele define a fonte Georgia para uso, e assim sucessivamente.

```
img {
    display: block;
    margin: 0 auto;
}
```

Com essa propriedade centralizamos a imagem para melhorar a aparência. Nós poderíamos usar



novamente o truque margin: 0 auto que aprendemos anteriormente para o corpo, mas também precisamos fazer outra coisa. O elemento **<body>** é em nível de bloco, o que significa que ocupa espaço na página e pode ter margens e outros valores de espaçamento aplicados a ele. Imagens, por outro lado, **são elementos em linha**, o que significa que não podem ter margens. Então, para aplicar margens a uma imagem, temos que dar o comportamento de nível de bloco a imagem usando display: block.

```
p {  
    margin: 25px 25px;  
    padding: 25px;  
    text-align: justify;  
    font-size: 25px;  
    color: #ffffff; /* Substitua #ffffff pela cor desejada */  
    text-shadow: 2px 2px 1px black; /* Adicione a cor para o efeito de sombra do texto */  
    font-family: 'Verdana', 'Geneva', 'Tahoma', sans-serif;  
}
```

Nos parágrafos aqui definimos propriedades para melhor alocar o texto na página, nota-se que as propriedades são bem parecidas com as do elemento <h1>;

**Margin:** - Definimos o margin com 25 pixels dos lados, para melhorar e espaçar o texto.

**Padding:** - Com o padding deixamos um espaço interno entre o bloco do texto.

**Text-align:** – Aqui nessa propriedade definimos que o parágrafo seja justificado.

**Font-size:** – Com a font-size deixamos a fonte do parágrafo com tamanho de 25 pixels.

**Color:** - Mudamos a cor do texto para branco, dando destaque no texto e contrastando com o fundo.

**Text-shadow:** - Colocamos uma sombra em preto no texto para dar um contraste com a cor branca.

**Font-family:** – Definimos a família de fontes para nosso parágrafo, assumindo a ordem de importância em caso a fonte não esteja instalada, automaticamente assume a próxima fonte da sequência.

Podemos notar **que as definições no código CSS são de muita importância para uma página HTML, sem o uso do CSS é praticamente inviável fazer o mesmo estilo em HTML puro**, por isso o uso do CSS se fez muito importante desde a sua criação.

RESPONDA:

- 1) O que é CSS?
- 2) O que faz um CSS dentro de um documento em HTML?
- 3) Qual a sintaxe para incorporar uma folha de estilo dentro de uma página em HTML?
- 4) Quais os principais tipos de seletores em CSS?
- 5) Qual a função da propriedade text-align?
- 6) Qual propriedade CSS preciso usar para mudar a cor do texto de um elemento <p> do HTML?
- 7) Para mudar a cor do background do HTML, qual a sintaxe?
- 8) Como funciona a propriedade text-shadow?
- 9) As propriedades \_\_\_\_\_ e \_\_\_\_\_ são essenciais para definir as dimensões de um elemento, como largura e altura.
- 10) Qual a diferença entre margin e padding?

## **\*\*Respostas:\*\***

### **1) \*\*O que é CSS?\*\***

- CSS (Cascading Style Sheets) é uma linguagem de estilo usada para estilizar documentos HTML. Ela permite determinar o layout e a aparência de elementos HTML em uma página da web. O CSS separa a apresentação do conteúdo, facilitando a manutenção e o design consistente em várias páginas.

### **2) \*\*O que faz um CSS dentro de um documento em HTML?\*\***

- O CSS é utilizado para aplicar estilos e layouts aos elementos HTML em uma página. Ele controla aspectos como cores, fontes, espaçamentos, tamanhos, posicionamento e outros atributos visuais, permitindo a criação de designs atraentes e consistentes em toda a página ou site.

### **3) \*\*Qual a sintaxe para incorporar uma folha de estilo dentro de uma página em HTML?\*\***

- A sintaxe para incorporar uma folha de estilo dentro de uma página HTML é através da tag `` no elemento ``. Exemplo:

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

### **4) \*\*Quais os principais tipos de seletores em CSS?\*\***

- Os principais tipos de seletores em CSS incluem:
  - Seletor de Tipo (Tag)
  - Seletor de ID
  - Seletor de Classe
  - Seletor de Atributo
  - Pseudo-Classe

### **5) \*\*Qual a função da propriedade text-align?**

- A propriedade `text-align` é usada para definir o alinhamento horizontal do texto dentro de um elemento. Os valores comuns são `left` (esquerda), `right` (direita), `center` (centro) e `justify` (justificado).

### **6) \*\*Qual propriedade CSS preciso usar para mudar a cor do texto de um elemento ` ` do HTML?**

- A propriedade CSS para mudar a cor do texto é `color`. Exemplo:

```
```css
p {
  color: blue;
}
```
```

### **7) \*\*Para mudar a cor do background do HTML, qual a sintaxe?**

- A sintaxe para mudar a cor do background é usando a propriedade `background-color`. Exemplo:

```

    ``css
    html {
        background-color: skyblue;
    }
    ``

```

#### 8) **Como funciona a propriedade text-shadow?**

- A propriedade `text-shadow` adiciona uma sombra ao texto. Seus valores definem o deslocamento horizontal, deslocamento vertical, raio de desfoque e a cor da sombra. Exemplo:

```

    ``css
    h1 {
        text-shadow: 2px 2px 2px black;
    }
    ``

```

#### 9) **As propriedades \_\_\_\_\_ e \_\_\_\_\_ são essenciais para definir as dimensões de um elemento, como largura e altura.**

- As propriedades `width` e `height` são essenciais para definir as dimensões de um elemento.

#### 10) **Qual a diferença entre margin e padding?**

- `Margin` é a distância entre a borda do elemento e as bordas externas dos elementos adjacentes, afetando o espaçamento entre os elementos. `Padding` é a distância entre o conteúdo do elemento e sua borda, afetando o espaçamento interno do elemento. Em resumo, `margin` lida com espaçamento externo, enquanto `padding` lida com espaçamento interno.

#### **ANEXOS:**

**Abaixo estão as propriedades de texto listadas na tabela com explicações curtas ao lado de cada uma:**

1. **`direction`:** Define a direção do texto, podendo ser da esquerda para a direita (`ltr`), da direita para a esquerda (`rtl`) ou herdada (`inherit`).
2. **`hanging-punctuation`:** Controla a pontuação que "pendura" para fora da caixa do texto. Pode ser configurado como `None` ou especificar valores para o início, fim e adjacentes.
3. **`letter-spacing`:** Define o espaçamento entre caracteres, podendo ser normal ou um valor em porcentagem.
4. **`punctuation-trim`:** Controla a pontuação que é cortada na quebra de linha. Pode ser configurado como `None` ou especificar valores para o início, fim e adjacentes.
5. **`text-align`:** Define o alinhamento horizontal do texto, como início, fim, esquerda, direita, centro ou justificado.
6. **`text-align-last`:** Define o alinhamento do último bloco de texto em um contêiner, similar ao `text-align`.

7. **`text-decoration`**: Adiciona decorações ao texto, como underline, overline, line-through ou blink.
8. **`text-emphasis`**: Adiciona ênfase ao texto, como accent, dot, circle ou disc, com opções para antes, depois e comprimento.
9. **`text-indent`**: Define o recuo da primeira linha do texto em relação à margem.
10. **`text-justify`**: Controla a justificação do texto entre as linhas, podendo ser automática, entre palavras, ideogramas, clusters, distribuída, kashida ou tibetana.
11. **`text-outline`**: Adiciona um contorno ao texto, com opções para nenhum, cor ou comprimento.
12. **`text-shadow`**: Adiciona uma sombra ao texto, permitindo especificar a cor e o comprimento do desfoque.
13. **`text-transform`**: Transforma o texto para maiúsculas, minúsculas ou capitalizado.
14. **`text-wrap`**: Controla como o texto deve ser quebrado em várias linhas, podendo ser normal, sem restrições, nenhum ou suprimido.
15. **`unicode-bidi`**: Controla a direção do texto bidirecional, podendo ser normal, incorporada ou substituída.
16. **`white-space`**: Define como o espaço em branco deve ser tratado, podendo ser normal, pré-formatado, sem quebra, com quebra pré-formatada ou preenchido.
17. **`white-space-collapse`**: Controla o colapso de espaços em branco, podendo preservar, colapsar, preservar quebras ou descartar.
18. **`word-break`**: Define como as quebras de palavra devem ser tratadas, podendo ser normal, manter tudo, solto, rigoroso ou quebrar tudo.
19. **`word-spacing`**: Define o espaçamento entre palavras, podendo ser normal ou um valor em porcentagem.
20. **`word-wrap`**: Controla como as palavras devem ser quebradas quando não cabem na largura disponível, podendo ser normal ou quebrar a palavra.

**Abaixo estão as propriedades de fonte listadas na tabela com explicações curtas ao lado de cada uma:**

1. **`Font`**: Esta propriedade compreende várias subpropriedades, incluindo `font-style`, `font-variant`, `font-weight`, `font-size`, e `font-family`. Essas subpropriedades são usadas para definir o estilo, variante, peso, tamanho e família da fonte, respectivamente. Outras subpropriedades, como `caption`, `icon`, `menu`, `message-box`, `smallcaption` e `status-bar`, também podem ser usadas para definir a fonte em contextos específicos.

2. **Font-size**: Define o tamanho da fonte, podendo ser especificado em valores fixos (`xx-small`, `x-small`, `small`, `medium`, `large`, `x-large`, `xx-large`), valores relativos (`smaller`, `larger`), ou valores percentuais (`length (%)`). A opção `inherit` herda o tamanho da fonte do elemento pai.
3. **Font-size-adjust**: Controla o ajuste do tamanho da fonte em relação à altura da fonte disponível. Pode ser configurado como `None`, `inherit`, ou um número em percentagem.
4. **Font-stretch**: Define a largura da fonte, podendo ser normal, mais ampla (`wider`), mais estreita (`narrower`), ou valores específicos de ultra-condensada a ultra-expandida. A opção `inherit` herda a largura da fonte do elemento pai.
5. **Font-style**: Define o estilo da fonte, podendo ser normal, itálico (`italic`), ou oblíquo (`oblique`). A opção `inherit` herda o estilo da fonte do elemento pai.
6. **Font-variant**: Define a variante da fonte, podendo ser normal ou small caps (`small-caps`). A opção `inherit` herda a variante da fonte do elemento pai.
7. **Font-weight**: Define a espessura ou peso da fonte, podendo ser normal, negrito (`bold`), mais negrito (`bolder`), mais leve (`lighter`), ou valores numéricos de 100 a 900. A opção `inherit` herda o peso da fonte do elemento pai.
8. **Font-family**: Define a família da fonte, especificando uma ou mais famílias de fontes. Pode incluir nomes específicos de família (`Family-name`), famílias genéricas (`generic-family`), ou herdar a família da fonte do elemento pai (`inherit`).