



# DISCIPLINA: LPW

PROFESSOR: JOEL RAMOS

## CONTEÚDO – TERCEIRO TRIMESTRE 2024

### Pseudo-Seletores CSS

Pseudo-seletores são utilizados para aplicar estilos a elementos com base em um estado ou característica específica, sem precisar modificar o HTML. Nesta seção, vamos explorar alguns pseudo-seletores comuns do CSS e como utilizá-los.

#### Conteúdo Posterior e Anterior

CSS permite que você personalize as fontes de texto de várias maneiras. As fontes são uma parte crítica de um site e podem afetar sua legibilidade, estética e até mesmo o desempenho da página.

Nesta seção, vamos abordar as principais propriedades que controlam as fontes em CSS, incluindo o tamanho, a família, o estilo, a espessura (ou peso) e a variação. Veremos também como usar um seletor abreviado para compactar essa configuração.

#### Tamanho da Fonte

A propriedade font-size define o tamanho da fonte em pixels, em pontos ou em outras unidades. É importante lembrar que o tamanho da fonte pode afetar a legibilidade e a aparência da página. É importante usar tamanhos de fonte adequados para diferentes partes da página, como cabeçalhos e parágrafos. O código abaixo mostra um exemplo de como definir o tamanho da fonte.

```
p {  
  font-size: 16px;  
}
```

Neste exemplo, definimos o tamanho da fonte para 16 pixels para todos os elementos <p> da página.

#### Família de Fontes

A propriedade font-family define a família de fontes usada no texto. É possível definir várias fontes alternativas para garantir que o texto seja exibido corretamente em diferentes navegadores e sistemas operacionais.

```
p {  
  font-family: "Helvetica Neue", Helvetica, Arial, sans-serif;  
}
```

Neste exemplo, definimos a fonte Helvetica Neue como a fonte preferida, mas se ela não estiver disponível, usaremos Helvetica ou Arial, ou qualquer fonte sans-serif disponível.

#### Estilo da Fonte

A propriedade font-style define o estilo da fonte como normal, itálico ou oblíqua. A fonte em itálico pode ser usada para enfatizar palavras-chaves ou frases.

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

Neste exemplo, definimos o estilo itálico para todos os elementos <p> da página.

## **Espessura (Peso) da Fonte**

A propriedade font-weight define a espessura da fonte como normal, negrito ou mais leve. É uma maneira fácil de enfatizar palavras-chaves ou frases.

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

Neste exemplo, definimos a fonte em negrito para todos os elementos <a> da página.

## **Variação da Fonte**

A propriedade font-variant define a variação da fonte como normal ou small-caps. A variação small-caps é útil para enfatizar palavras ou frases, tornando-as em letras maiúsculas menores do que as maiúsculas normais.

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

Neste exemplo, definimos a variação small-caps para todos os elementos <p> da página.

## **Propriedade Abreviada**

A propriedade font permite definir de forma abreviada várias propriedades de fontes em uma única linha de código. Ela inclui a font-size, font-family, font-style, font-weight e font-variant.

```
h1 {  
  font: italic small-caps bold 24px/1.5  
  "Helvetica Neue", Helvetica, Arial, sans-serif;  
}
```

Neste exemplo, definimos várias propriedades de fontes para o elemento <h1>, incluindo font-style em itálico, font-variant em small-caps, font-weight em negrito, font-size em 24 pixels com uma altura de linha de 1,5 e font-family com a preferência para Helvetica Neue, seguida por Helvetica, Arial ou qualquer fonte sans-serif disponível.

## **Textos**

Nesta seção, vamos explorar várias propriedades CSS que afetam o estilo de texto em um documento web. Veremos como essas propriedades afetam a aparência do texto e como elas podem ser usadas para criar efeitos visuais interessantes.

### **Alinhamento Horizontal**

A propriedade CSS text-align é usada para definir o alinhamento horizontal do texto em um elemento. Ela pode ter os valores left, right, center e justify que, respectivamente, alinham o texto à esquerda, à direita, no centro ou distribui o texto uniformemente na largura disponível, criando espaçamento adicional entre as palavras, se necessário.

```
p {  
  text-align: left;  
}
```

Neste exemplo, a propriedade text-align é aplicada ao elemento <p>, fazendo com que todo o texto dentro do elemento seja alinhado à esquerda.

### **Direção da Escrita**

As propriedades CSS direction e unicode-bidi são usadas para controlar a direção do texto em um elemento. A propriedade direction pode ter os valores ltr (esquerda para a direita) ou rtl (direita para a esquerda).

A propriedade unicode-bidi é usada para controlar a direção do texto em elementos que contêm texto com diferentes direções de fluxo, como texto em árabe ou hebraico. Ela pode ter os valores normal para texto esquerda-para-direita, ou bidi-override para texto direita-para-esquerda.

```
h1 {  
  direction: rtl;  
  unicode-bidi: bidi-override;
```

```
}
```

Neste exemplo, as propriedades `direction` e `unicode-bidi` são aplicadas ao elemento `h1`, fazendo com que todo o texto dentro do elemento seja exibido da direita para a esquerda.

### **Decorações**

A propriedade CSS `text-decoration` é usada para adicionar ou remover decorações de texto, como sublinhado, riscado ou sobrelinhado. Ela pode ter um os valores `none`, `underline`, `overline` e `line-through`.

```
a {  
  text-decoration: line-through;  
}
```

Neste exemplo, a propriedade `text-decoration` é aplicada ao elemento `<a>`, fazendo com que todos os links no documento tenham uma linha riscando o texto.

### **Transformações**

A propriedade CSS `text-transform` é usada para transformar textos. Ela pode ter os valores `none`, `uppercase`, `lowercase` e `capitalize` que, respectivamente, exclui qualquer transformação do texto, converte em letras maiúsculas, converte em letras minúsculas e converte a primeira letra de cada palavra em letra maiúscula.

```
h2 {  
  text-transform: uppercase;  
}
```

Neste exemplo, a propriedade `text-transform` é aplicada ao elemento `h2`, fazendo com que todo o texto dentro do elemento seja convertido para letras maiúsculas.

### **Indentação**

A propriedade CSS `text-indent` é usada para definir a indentação da primeira linha de um bloco de texto.

```
p {  
  text-indent: 20px;  
}
```

Neste exemplo, a propriedade `text-indent` é aplicada ao elemento `<p>`, fazendo com que a primeira linha do parágrafo seja indentada em 20 pixels.

### **Espaçamento Entre Letras**

A propriedade CSS `letter-spacing` é usada para ajustar o espaçamento entre letras em um texto.

Ela pode ter um valor positivo ou negativo para aumentar ou diminuir o espaçamento entre as letras. `h3`

```
{  
  letter-spacing: 2px;  
}
```

Neste exemplo, a propriedade `letter-spacing` é aplicada ao elemento `h3`, aumentando o espaçamento entre as letras em 2 pixels.

### **Altura da Linha**

A propriedade CSS `line-height` é usada para definir a altura da linha de um bloco de texto. O valor pode ser especificado em pixels, em ou porcentagem.

```
p {  
  line-height: 1.5em;  
}
```

Neste exemplo, a propriedade `line-height` é aplicada ao elemento `<p>`, aumentando a altura da linha para 1,5 vezes o tamanho da fonte.

### **Espaçamento Entre Palavras**

A propriedade CSS `word-spacing` é usada para ajustar o espaçamento entre palavras em um texto. Ela pode ter um valor positivo ou negativo para aumentar ou diminuir o espaçamento entre as palavras.

```
h4 {  
  word-spacing: 4px;  
}
```

Neste exemplo, a propriedade word-spacing é aplicada ao elemento h4, aumentando o espaçamento entre as palavras para 4 pixels.

### **Quebras no Espaço em Branco**

A propriedade CSS white-space é usada para definir como o espaço em branco é tratado em um elemento. Ele pode assumir os valores normal, nowrap, pre e pre-wrap. O valor normal trata vários espaços em branco como um único espaço, e quebra linhas automaticamente quando o texto atinge a largura máxima do elemento. O valor nowrap impede que o texto quebre em várias linhas. O valor pre preserva os espaços em branco e quebras de linha no texto exatamente como eles são no código fonte. O valor pre-wrap preserva os espaços em branco e quebras de linha, mas permite que o texto seja quebrado em várias linhas.

```
p {  
  white-space: pre-wrap;  
}
```

Neste exemplo, a propriedade white-space é aplicada ao elemento <p>, preservando os espaços em branco e quebras de linha no texto.

### **Sombras**

A propriedade CSS text-shadow é usada para adicionar sombras ao texto em um elemento. Ela pode ter um ou mais valores, separados por vírgulas, que definem a posição da sombra, a cor da sombra e o tamanho da sombra.

```
h5 {  
  text-shadow: 2px 2px #000000;  
}
```

Neste exemplo, a propriedade text-shadow é aplicada ao elemento h5, adicionando uma sombra preta de 2 pixels de tamanho à direita e abaixo do texto.

### **Links**

Os links são elementos importantes em uma página web, pois permitem que o usuário navegue pelo conteúdo. Porém, é possível ir além da formatação padrão dos links e criar efeitos visuais e de interatividade com CSS. Nesta seção, vamos explorar algumas técnicas para estilizar links usando CSS.

#### **Links Visitados e Não Visitados**

Os pseudo-seletores a:link e a:visited são usados para definir a aparência dos links que ainda não foram visitados e dos links que já foram visitados, respectivamente. É possível personalizar a cor do texto, a cor do fundo, a borda, entre outras propriedades.

```
a:visited {  
  color: #999;  
  text-decoration: line-through;  
}
```

Neste exemplo, definimos que os links visitados terão a cor do texto cinza escuro (#999) e o texto riscado (text-decoration: line-through).

#### **Links Ativos e Sob o Mouse**

Os pseudo-seletores a:hover e a:active são usados para definir a aparência dos links quando o usuário passa o mouse sobre eles e quando eles estão sendo clicados, respectivamente. É possível personalizar a cor do texto, a cor do fundo, a borda, entre outras propriedades.

```
a:hover {
```

```
color: #000;
background-color: #ff0;
}
```

```
a:active {
color: #fff;
background-color: #f00;
}
```

Neste exemplo, definimos que quando o usuário passar o mouse sobre o link, a cor do texto será preta (#000) e o fundo será amarelo (#ff0). Quando o link estiver sendo clicado, a cor do texto será branca (#fff) e o fundo será vermelho (#f00).

Vale mencionar que os pseudo-seletores `a:hover` e `a:active` não funcionam em dispositivos móveis, pois não há eventos de mouse em dispositivos móveis. Ainda, os pseudo-seletores `:hover` e `:active` podem ser usados em outros tipos de elementos HTML, como botões e imagens.

### Transformando um Link em Botão

Usando as técnicas abordadas até aqui, é possível transformar um link em um botão, adicionando bordas, fundo e espaçamento.

```
a.button {
display: inline-block;
padding: 0.5em 1em;
text-align: center;
background-color: #f00;
color: #fff;
text-decoration: none;
border-radius: 0.5em;
box-shadow: 0 0.2em 0.4em rgba(0, 0, 0, 0.4);
}
```

```
a.button:hover {
background-color: rgb(201, 0, 0);
}
```

```
a.button:active {
position: relative;
top: 0.1em;
left: 0.1em;
box-shadow: none;
}
```

Neste exemplo, definimos uma classe `button` para o link e adicionamos propriedades para exibir o link como um botão, incluindo margem interna (`padding`), alinhamento de texto (`text-align`), cor de fundo (`background-color`), cor do texto (`color`), remoção de sublinhado (`text-decoration:none`), cantos arredondados (`border-radius`) e sombra (`box-shadow`).

Essas técnicas de formatação de links em CSS permitem que você crie uma experiência mais agradável e interativa para os usuários da sua página web. Ao usar os pseudo-seletores `a:link`, `a:visited`, `a:hover` e `a:active` e transformar um link em botão, você pode personalizar a aparência dos links de acordo com o design da sua página.

### Listas

Listas são elementos comuns em documentos HTML. Com CSS, podemos modificar a aparência

das listas, tornando-as mais atraentes e personalizadas. Esta seção aborda as técnicas básicas para a formatação de listas usando CSS.

### **Tipo de Lista**

A primeira coisa que podemos modificar é o tipo de lista. O tipo padrão de lista é uma lista com marcadores (bullets), mas também podemos ter listas numeradas. Para mudar o tipo de lista, usamos a propriedade `list-style-type`. Os valores possíveis incluem:

- a) `disc`: o marcador padrão, que é um círculo sólido;
- b) `circle`: um círculo aberto;
- c) `square`: um quadrado sólido;
- d) `decimal`: uma lista numerada com números arábicos;
- e) `lower-alpha`: uma lista numerada com letras minúsculas;
- f) `upper-alpha`: uma lista numerada com letras maiúsculas;
- g) `lower-roman`: uma lista numerada com números romanos em minúsculas;
- h) `upper-roman`: uma lista numerada com números romanos em maiúsculas.

```
ul {  
  list-style-type: square;  
}
```

Neste exemplo, a propriedade `list-style-type` é aplicada a um elemento `<ul>`, alterando o tipo de lista para quadrados sólidos.

### **Imagem do Marcador**

Outra forma de personalizar uma lista é usando imagens como marcadores. Para fazer isso, usamos a propriedade `list-style-image`. O valor desta propriedade é a URL da imagem que será usada como marcador. É importante notar que a imagem é usada em substituição ao marcador padrão.

Portanto, se a imagem não carregar corretamente, nenhum marcador será exibido.

```
ul {  
  list-style-image: url('marcador.png');  
}
```

Neste exemplo, a propriedade `list-style-image` é aplicada a um elemento `ul`, usando a imagem `marcador.png` como marcador.

### **Posição do Marcador**

A propriedade `list-style-position` define a posição do marcador em relação ao conteúdo do item da lista. Por padrão, o marcador é exibido à esquerda do conteúdo do item, mas com essa propriedade é possível movê-lo para a direita ou para dentro do conteúdo do item. Os valores possíveis para a propriedade `list-style-position` são `inside`, onde o marcador é colocado dentro do conteúdo do item da lista, e `outside`, onde o marcador é colocado à esquerda do conteúdo do item da lista.

```
ul {  
  list-style-position: inside;  
}
```

Neste exemplo, a propriedade `list-style-position` é aplicada a um elemento `ul`, movendo o marcador para dentro do conteúdo do item da lista. Essa propriedade é especialmente útil quando usamos imagens como marcadores, pois podemos posicioná-las de forma mais precisa em relação ao conteúdo do item da lista. Por exemplo, se a imagem do marcador tiver um tamanho maior do que o padrão, podemos usar a propriedade `list-style-position: inside` para garantir que o conteúdo do item não fique muito afastado do marcador.

### **Margem, Preenchimento e Indentação**

Por fim, podemos ajustar a margem, o preenchimento e a indentação das listas para deixá-las mais atraentes. Para isso, usamos as propriedades `margin`, `padding` e `text-indent`. A propriedade `margin` define a margem da lista, enquanto a propriedade `padding` define o espaço interno da lista. A propriedade `text-indent` define o recuo da primeira linha do texto da lista.

```
ul {  
  margin: 0;
```

```
padding: 0;  
text-indent: 10px;  
}
```

Neste exemplo, as propriedades margin, padding e text-indent são aplicadas a um elemento <ul>, definindo margem, preenchimento e recuo do texto da lista.

```
ul {  
list-style-image: url('marcador.png');  
margin: 0;  
padding: 0;  
text-indent: 10px;  
}
```