

HTML 5 Responsivo

Criando Sites Responsivos Com HTML5 do Zero

Módulo 5 – Avançando com CSS 3

Autores:

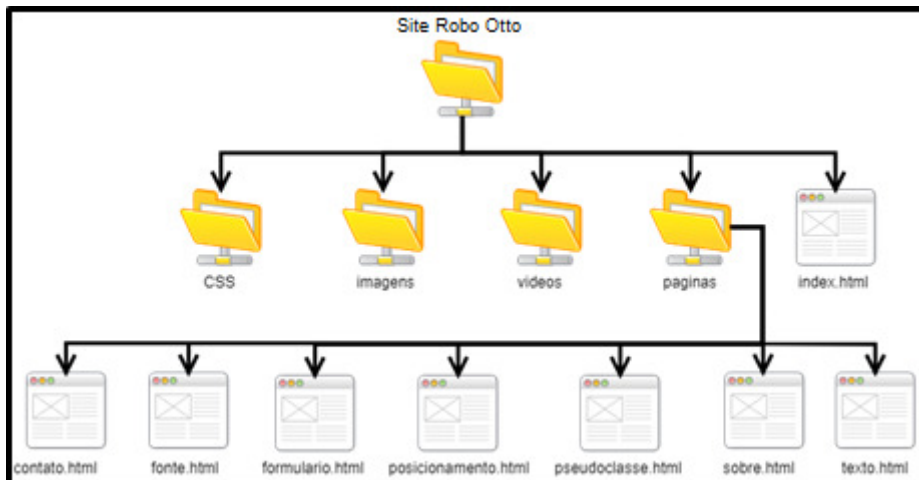
Caique Zaneti Kirilo

Luiz Carlos Machi Lozano

SÃO PAULO, 2020

Passo 47 – Nova paginação

Vamos agora criar uma pasta na nossa estrutura e dentro dela fazer 7 novas páginas html como mostra a imagem a seguir:



Para facilitar nossa vida, por enquanto, vamos colocar um código básico em todas estas páginas para que durante o decorrer dos passos seja possível acessá-las, como mostra o exemplo:

```
1 <!DOCTYPE html>
2 <html lang="pt-br">
3   <head>
4     <title>Robô Otto</title>
5     <meta charset="utf-8">
6     <link rel="stylesheet" href="../CSS/styles.css">
7     <link href="https://fonts.googleapis.com/css?family=Lato" rel="stylesheet">
8   </head>
9   <body>
10    <main>
11      <h1>Título da página</h1>
12    </main>
13  </body>
14 </html>
```

Coloque em cada página respectivamente seu próprio título h1, por exemplo:

```
1 <!DOCTYPE html>
2 <html lang="pt-br">
3   <head>
4     <title>Robô Otto - Posicionamento</title>
5     <meta charset="utf-8">
6     <link rel="stylesheet" href="../CSS/styles.css">
7     <link href="https://fonts.googleapis.com/css?family=Lato" rel="stylesheet">
8   </head>
9   <body>
10    <main>
11      <h1>Posicionamento</h1>
12    </main>
13  </body>
14 </html>
```

Passo 48 – Criando Menus

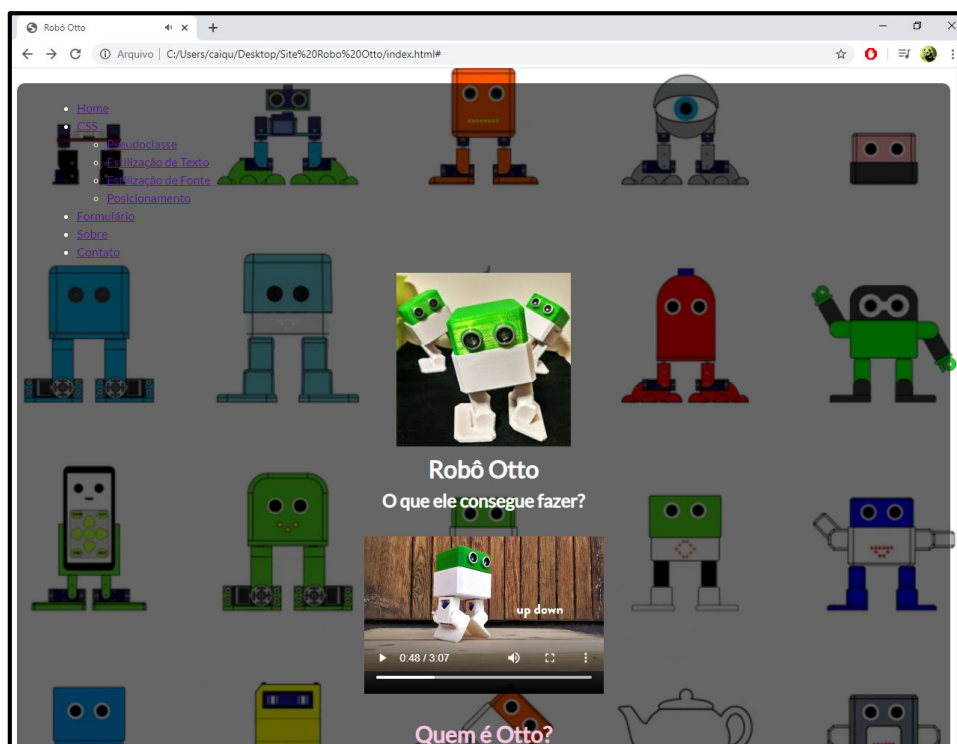
Agora que temos uma grande quantidade de páginas para administrar, devemos criar um menu para conseguir acessá-las mais facilmente, sendo assim, dentro do nosso index vamos colocar o seguinte código básico:

```

10      <main>
11      <ul id="menu">
12          <li><a href="index.html">Home</a></li>
13          <li>
14              <a href="#">CSS</a>
15              <ul>
16                  <li><a href="paginas/pseudoclassee.html">Pseudoclassee </a></li>
17                  <li><a href="paginas/texto.html">Estilização de Texto</a></li>
18                  <li><a href="paginas/fonte.html">Estilização de Fonte</a></li>
19                  <li><a href="paginas/posicionamento.html">Posicionamento</a></li>
20              </ul>
21          </li>
22          <li><a href="paginas/formulario.html">Formulario</a></li>
23          <li><a href="paginas/sobre.html">Sobre</a></li>
24          <li><a href="paginas/contato.html">Contato</a></li>
25      </ul>

```

Note que na realidade fizemos uma lista de links, onde em alguns casos uma lista estava dentro da outra:



Assim como muitas das coisas que fizemos nos capítulos passados, nada faz sentido sem o CSS para dar vida com seus estilos, sendo assim, vamos mudar essa lista para que ela se pareça cada vez mais com um menu. Dentro do nosso

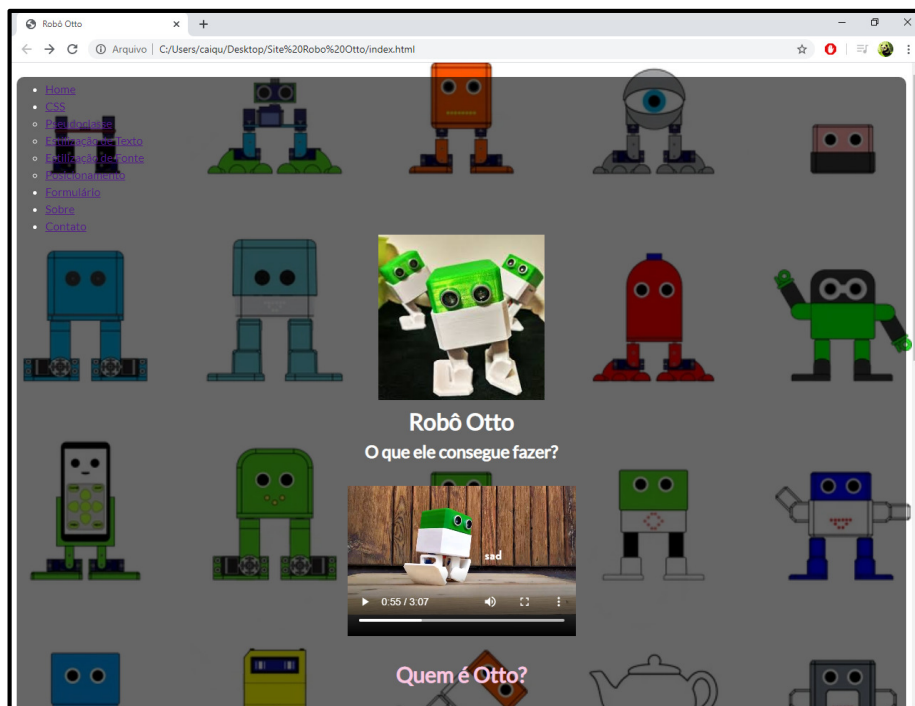
arquivo CSS, vamos começar nossa estilização do “zero” pois iremos criar um id para resetar os estilos da tag para o menu:

```

46 #menu, #menu ul {
47     margin: 0;
48     padding: 0;
49     list-style: none;
50 }

```

Note que desta vez não usamos uma classe, mas sim um id, mas afinal qual a diferença entre CLASS e ID? As classes (class) nada mais são do que uma forma simples de identificar um grupo de elementos, podendo assim formatar VÁRIOS elementos de uma vez só. Já os ids (id) são uma forma de identificar um elemento único, são identificadores particulares de cada elemento e somente ele.



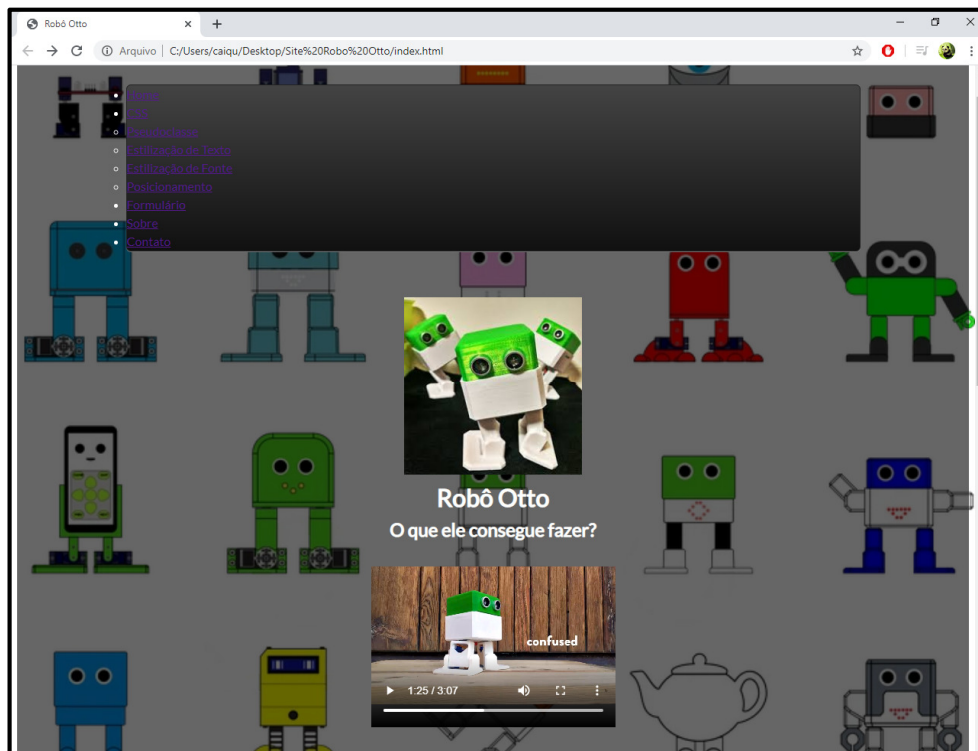
Note que perdemos os estilos base das listas. Imagine que o #menu seja o equivalente ao <main> para o nosso menu, sendo assim, tudo que estiver relacionado a nosso menu começará com o id #menu:

```

52 #menu {
53     width: 960px;
54     margin: 60px auto;
55     border: 1px solid #222;
56     background-color: #111;
57     background-image: linear-gradient(#444, #111);
58     border-radius: 6px;
59     box-shadow: 0 1px 1px #777;
60 }

```

Note que o resultado foi um equivalente as divs que estávamos montando anteriormente:



Ainda não se parece com um menu pois cada item está um embaixo do outro, para mudar essa situação devemos fazer com que os elementos fltuem (float) sendo assim, vamos utilizar um método de para limpar a flutuação sugerido por Nicolas Gallagher (Gallagher, 2011), que será útil no final do nosso menu:

```
62 #menu:before ,
63 #menu:after {
64     content: "";
65     display: table;
66 }
67
68 #menu:after {
69     clear: both;
70 }
71
72 #menu {
73     zoom: 1;
74 }
```

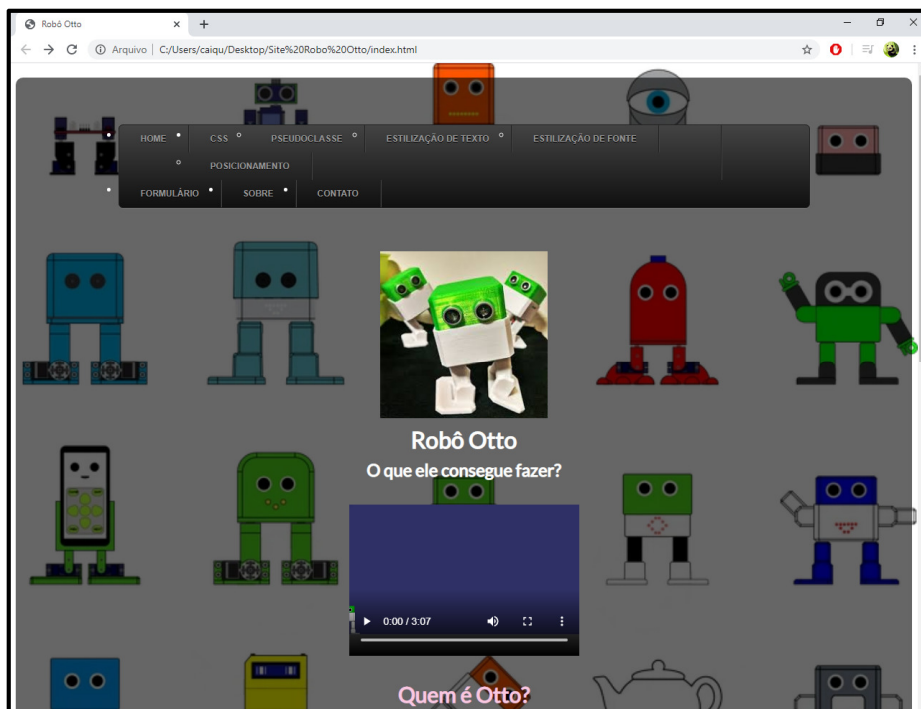
Depois de inserirmos o código para limpar os elementos flutuantes ainda não teremos mudanças visuais significativas, sendo assim vamos fazer os elementos efetivamente flutuarem agora.

```

76 #menu li {
77     float: left;
78     border-right: 1px solid #222;
79     box-shadow: 1px 0 0 #444;
80     position: relative;
81 }
82
83 #menu a {
84     float: left;
85     padding: 12px 30px;
86     color: #999;
87     text-transform: uppercase;
88     font: bold 12px Arial, Helvetica;
89     text-decoration: none;
90     text-shadow: 0 1px 0 #000;
91 }
92
93 #menu li:hover > a {
94     color: #fafafa;
95 }

```

Note que em cada vez mais estamos alterando elementos da lista que criamos para que ela se pareça com um menu, este é o grande pulo do gato neste menu dropdown que estamos criando, o resultado depois de inserirmos este código é algo mais parecido com um menu:



Precisamos agora fazer o tratamento dos submenus que criamos, para isso iremos adicionar o código abaixo.

```

97  #menu ul {
98      margin: 20px 0 0 0;
99      opacity: 0;
100     visibility: hidden;
101     position: absolute;
102     top: 38px;
103     left: 0;
104     z-index: 1;
105     background: #444;
106     background: linear-gradient(#444, #111);
107     box-shadow: 0 -1px 0 rgba(255,255,255,.3);
108     border-radius: 3px;
109     transition: all .2s ease-in-out;
110 }
111
112 #menu li:hover > ul {
113     opacity: 1;
114     visibility: visible;
115     margin: 0;
116 }
117
118 #menu ul ul {
119     top: 0;
120     left: 150px;
121     margin: 0 0 0 20px;
122     box-shadow: -1px 0 0 rgba(255,255,255,.3);
123 }

```

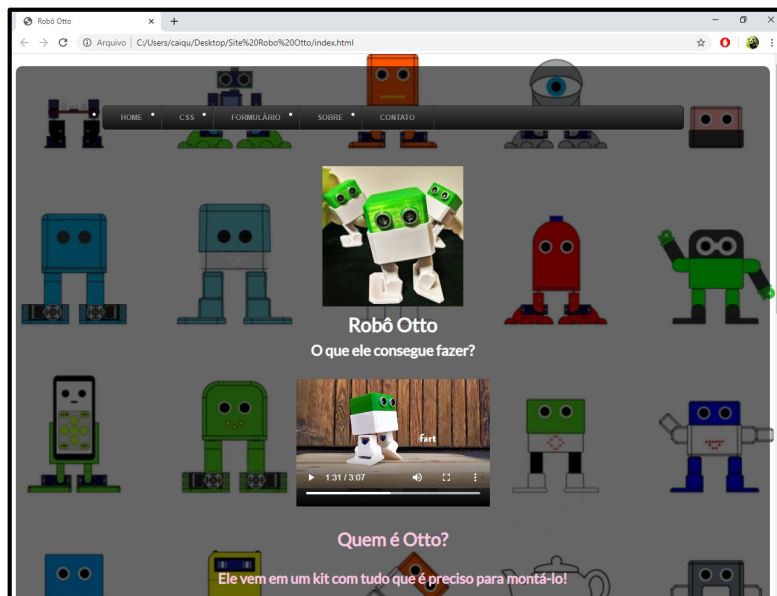
Note que já é possível perceber a real importância do método de limpeza de float que comentamos anteriormente, pois ao clicarmos em um menu que tem submenus, eles aparecem, mas assim que tiramos o ponteiro do mouse de cima ele some. Continuemos com o código:

```

125 #menu ul li {
126     float: none;
127     display: block;
128     border: 0;
129     box-shadow: 0 1px 0 #111, 0 2px 0 #666;
130 }
131
132 #menu ul li:last-child {
133     box-shadow: none;
134 }
135
136 #menu ul a {
137     padding: 10px;
138     width: 130px;
139     display: block;
140     white-space: nowrap;
141     float: none;
142     text-transform: none;
143 }
144
145 #menu ul a:hover {
146     background-color: #0186ba;
147     background-image: linear-gradient(#04acec, #0186ba);
148 }
149
150 #menu ul li:first-child > a {
151     border-radius: 3px 3px 0 0;
152 }

```


Agora sim temos um menu de respeito, que ainda precisa de polimentos, mas já possui cara de menu.



Vamos finalizar então nosso menu com o código abaixo, onde são mais polimentos de estilo sendo adicionados.

```
156 #menu ul li:first-child > a:after {
157     content: '';
158     position: absolute;
159     left: 40px;
160     top: -6px;
161     border-left: 6px solid transparent;
162     border-right: 6px solid transparent;
163     border-bottom: 6px solid #444;
164 }
165
166 #menu ul ul li:first-child a:after {
167     left: -6px;
168     top: 50%;
169     margin-top: -6px;
170     border-left: 0;
171     border-bottom: 6px solid transparent;
172     border-top: 6px solid transparent;
173     border-right: 6px solid #3b3b3b;
174 }
175
176 #menu ul li:first-child a:hover:after {
177     border-bottom-color: #04acec;
178 }
179
180 #menu ul ul li:first-child a:hover:after {
181     border-right-color: #0299d3;
182     border-bottom-color: transparent;
183 }
184
185 #menu ul li:last-child > a {
186     border-radius: 0 0 3px 3px;
187 } /*
```

Agora sim temos um menu para chamar de nosso.

Passo 49 – Pseudoclasse e pseudo-elemento

No capítulo anterior nós utilizamos pseudo-elementos na estilização do nosso conteúdo. Um pseudo-elemento é um identificador adicionado a um seletor que permite que seja estilizado apenas uma parte específica do elemento selecionado. É útil quando queremos mudar por exemplo a primeira letra do título, ou a primeira linha do parágrafo.

Sua sintaxe é bem simples

```
1  seletor::pseudo-elemento {  
2      propriedade: valor;  
3  }
```

Você pode utilizar apenas um pseudo-elemento em um seletor. Ele deve aparecer depois da declaração de um elemento simples.

Abaixo segue uma lista de pseudo-elementos mais utilizados:

::after cria um pseudo-elemento que é o último filho do elemento selecionado.

::before cria um pseudo-elemento que é o primeiro filho do elemento atingido

::cue esse pseudo-elemento possibilita a customização de legendas (em vídeos por exemplo)

::first-letter esse pseudo-elemento seleciona a primeira letra da primeira linha de um bloco, se não for precedida por qualquer outro conteúdo (como imagens ou tabelas embutidas) em sua linha.

::first-line aplica estilos à primeira linha de um elemento de nível de bloco. Observe que o comprimento da primeira linha depende de muitos fatores, incluindo a largura do elemento, a largura do documento e o tamanho da fonte do texto.

::selection aplica estilos à parte de um documento destacado pelo usuário (como clicar e arrastar o mouse sobre o texto).

::slotted () representa qualquer elemento que tenha sido colocado em um slot dentro de um modelo HTML.

::backdrop pode ser utilizado para criar um pano de fundo que esconde o conteúdo por trás de um elemento posicionado no topo de uma pilha de elementos. Por exemplo, para o elemento que é mostrado em tela cheia conforme descrito nessa especificação.

::placeholder ele customiza o texto, possibilitando a criação de uma moldura em torno do texto.

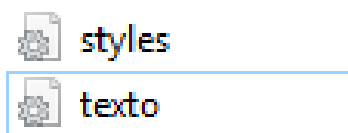
::marker seleciona a caixa do marcador de um item da lista, que normalmente contém um marcador ou um número.

::spelling-error representa um segmento de texto que o user agent sinalizou como digitado incorretamente.

::grammar-error representa um segmento de texto que o user agent sinalizou como gramaticalmente incorreto.

Passo 50 – Estilizando o texto

Chegou o momento de trabalharmos um pouco mais com a estilização do nosso texto, vamos então usar uma página específica para testarmos essas novas funções. Começaremos a codificar nossa página de estilização de texto com um css específico dela também



O código para nosso arquivo texto.css será:

```

1  <!DOCTYPE html>
2  <html lang="pt-br">
3  <head>
4      <title>Robô Otto</title>
5      <meta charset="utf-8">
6      <link rel="stylesheet" href="../CSS/texto.css">
7      <link href="https://fonts.googleapis.com/css?family=Lato" rel="stylesheet">
8  </head>
9  <body>
10     <main>
11
12         <h1> Estilização de Texto </h1>
13         <h2> Estilização de Texto </h2>
14         <h3> Estilização de Texto </h3>
15         <h4> Estilização de Texto </h4>
16
17     </main>
18 </body>
19 </html>

```

Text Alignment

A propriedade text-align é usada para definir o alinhamento horizontal de um texto. Um texto pode ser alinhado à esquerda ou à direita, centrado ou justificado.

O exemplo a seguir mostra o alinhamento ao centro e o texto alinhado à esquerda e à direita. (o alinhamento à esquerda é o padrão, se a direção do texto for da esquerda para a direita. E o alinhamento à direita é padrão se a direção do texto for da direita para a esquerda). Quando a propriedade text-align é configurada para "justificar", cada linha é esticada para que cada linha tenha

largura igual. E as margens esquerda e direita são retas (como em revistas e jornais)

```
1  h1 {  
2      text-align: center;  
3  }  
4  
5  h2 {  
6      text-align: left;  
7  }  
8  
9  h3 {  
10     text-align: right;  
11 }  
12  
13 h4 {  
14     text-align: justify;  
15 }
```

Text Decoration

O `text-decoration` é usada para definir ou remover decorações do texto.

`text-decoration: none;` é frequentemente usado para remover sublinhados dos links.

Os outros valores para o `text-decoration` efetivamente mudam o texto.

```
1  h1 {  
2      text-align: center;  
3      text-decoration: overline;  
4  }  
5  
6  h2 {  
7      text-align: left;  
8      text-decoration: line-through;  
9  }  
10  
11 h3 {  
12     text-align: right;  
13     text-decoration: underline;  
14 }  
15  
16 h4 {  
17     text-align: justify;  
18     text-decoration: none;  
19 }
```

Text Transformation

A propriedade `text-transform` é usada para especificar letras maiúsculas e minúsculas em um texto. Pode ser usado para transformar tudo em letras maiúsculas ou minúsculas, ou capitalizar a primeira letra de cada palavra.

```

1  h1 {
2      text-align: center;
3      text-decoration: overline;
4      text-transform: uppercase;
5  }
6
7  h2 {
8      text-align: left;
9      text-decoration: line-through;
10     text-transform: lowercase;
11 }
12
13 h3 {
14     text-align: right;
15     text-decoration: underline;
16     text-transform: capitalize;
17 }
18
19 h4 {
20     text-align: justify;
21     text-decoration: none;
22 }

```

Text Indentation

A propriedade `text-indent` é usada para especificar o recuo da primeira linha de um texto.

```

19  h4 {
20      text-align: justify;
21      text-decoration: none;
22      text-indent: 50px;
23  }

```

Letter Spacing

A propriedade de espaçamento entre letras é usada para especificar o espaço entre os caracteres em um texto. O exemplo a seguir demonstra como aumentar ou diminuir o espaço entre os caracteres.

```

1  h1 {
2      text-align: center;
3      text-decoration: overline;
4      text-transform: uppercase;
5      letter-spacing: 3px;
6  }
7
8  h2 {
9      text-align: left;
10     text-decoration: line-through;
11     text-transform: lowercase;
12     letter-spacing: -3px;
13 }

```

Line Height

A propriedade `line-height` é usada para especificar o espaço entre as linhas.

```

15  h3 {
16      text-align: right;
17      text-decoration: underline;
18      text-transform: capitalize;
19      line-height: 0.8;
20  }
21
22  h4 {
23      text-align: justify;
24      text-decoration: none;
25      text-indent: 50px;
26      line-height: 1.8;
27  }

```

Text Direction

A propriedade `direction` é usada para alterar a direção do texto de um elemento.

```

1  h1 {
2      text-align: center;
3      text-decoration: overline;
4      text-transform: uppercase;
5      letter-spacing: 3px;
6      direction: rtl;
7  }

```

Word Spacing

A propriedade `word-spacing` é usada para especificar o espaço entre as palavras em um texto. O exemplo a seguir demonstra como aumentar ou diminuir o espaço entre as palavras.

```

9  h2 {
10     text-align: left;
11     text-decoration: line-through;
12     text-transform: lowercase;
13     letter-spacing: -3px;
14     word-spacing: 10px
15 }
16
17 h3 {
18     text-align: right;
19     text-decoration: underline;
20     text-transform: capitalize;
21     line-height: 0.8;
22     word-spacing: -5px
23 }

```

Text Shadow

A propriedade `text-shadow` adiciona sombra ao texto. O exemplo a seguir especifica a posição da sombra horizontal (3px), a posição da sombra vertical (2px) e a cor da sombra (vermelha).

```

25  h4 {
26      text-align: justify;
27      text-decoration: none;
28      text-indent: 50px;
29      line-height: 1.8;
30      text-shadow: 3px 2px red;
31  }

```

Abaixo temos uma lista com propriedades de texto e sua funcionalidade:

Propriedade	Descrição
color	Define a cor do texto
direction	Especifica a direção do texto / direção da escrita
letter-spacing	Aumenta ou diminui o espaço entre os caracteres em um texto
line-height	Define a altura da linha
text-align	Especifica o alinhamento horizontal do texto
text-decoration	Especifica a decoração adicionada ao texto
text-indent	Especifica o recuo da primeira linha em um bloco de texto
text-shadow	Especifica o efeito de sombra adicionada ao texto
text-transform	Controla a capitalização do texto
text-overflow	Especifica como o conteúdo excedido que não é exibido deve ser sinalizado para o usuário
unicode-bidi	Usado junto com a propriedade direction para definir ou retornar se o texto deve ser substituído para suportar vários idiomas no mesmo documento
vertical-align	Define o alinhamento vertical de um elemento
white-space	Especifica como o espaço em branco dentro de um elemento é manipulado
word-spacing	Aumenta ou diminui o espaço entre palavras em um texto

Passo 51 - Estilizando fontes

Existem dois tipos de nomes de famílias de fontes:

generic family - um grupo de famílias de fontes com aparência semelhante (como "Serif" ou "Monospace")

font family - uma família de fontes específica (como "Times New Roman" ou "Arial")

Generic family	Font family	Descrição
Serif	Times New Roman Georgia	A fonte Serif têm pequenas linhas nas extremidades de alguns caracteres
Sans-serif	Arial Verdana	“Sans” - essas fontes não têm as linhas nas extremidades dos caracteres

Monospace	Courier New Lucida Console	Todos os caracteres monospace têm a mesma largura
-----------	---------------------------------------	--

Vamos então novamente criar um novo arquivo css chamado fonte.css e vamos colocar os exemplos deste passo lá, para nossa página html, colocaremos um texto base parecido com o anterior:

```

1  <!DOCTYPE html>
2  <html lang="pt-br">
3  <head>
4      <title>Robô Otto</title>
5      <meta charset="utf-8">
6      <link rel="stylesheet" href="../CSS/texto.css">
7      <link href="https://fonts.googleapis.com/css?family=Lato" rel="stylesheet">
8  </head>
9  <body>
10     <main>
11
12         <h1> Estilização de Texto </h1>
13         <h2> Estilização de Texto </h2>
14         <h3> Estilização de Texto </h3>
15         <h4> Estilização de Texto </h4>
16
17     </main>
18 </body>
19 </html>

```

Font Family

O conjunto de fontes de um texto é definida com a propriedade **font-family**. A propriedade **font-family** deve conter vários nomes de fontes como um sistema de "fallback". Se o navegador não suportar a primeira fonte, ele tentará a próxima fonte e assim por diante. Comece com a fonte desejada e termine com uma família genérica, para permitir que o navegador escolha uma fonte semelhante na família genérica, se nenhuma outra fonte estiver disponível. **Nota:** Se o nome de uma família de fontes for mais de uma palavra, ela deverá estar entre aspas, como: "Times New Roman". Mais de uma família de fontes é especificada em uma lista separada por vírgulas:

```

1  h1 {
2      font-family: "Times New Roman", Times, serif;
3  }

```

Font Style

A propriedade **font-style** é usada principalmente para especificar texto em itálico. Esta propriedade tem três valores:

- normal - O texto é mostrado normalmente
- italic - O texto é mostrado em itálico
- oblique - The text is "leaning" (oblique is very similar to italic, but less supported)

```
1 h1 {  
2   font-family: "Times New Roman", Times, serif;  
3   font-style: normal;  
4 }  
5  
6 h2 {  
7   font-style: italic;  
8 }  
9  
10 h3 {  
11   font-style: oblique;  
12 }
```

Font Size

A propriedade **font-size** define o tamanho do texto. Ser capaz de gerenciar o tamanho do texto é importante no design da web. No entanto, você não deve usar ajustes de tamanho de fonte para fazer com que os parágrafos pareçam cabeçalhos, ou cabeçalhos parecidos com parágrafos. Sempre use as tags HTML adequadas, como `<h1>` - `<h6>` para títulos e `<p>` para parágrafos. O valor do tamanho da fonte pode ser um tamanho absoluto ou relativo. Tamanho absoluto:

- Define o texto para um tamanho especificado
- 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 em relação aos elementos circundantes
- Permite que um usuário altere o tamanho do texto em navegadores

Mudando o tamanho da fonte com pixels

Definir o tamanho do texto com pixels oferece controle total sobre o tamanho do texto:

```
1  h1 {  
2    font-family: "Times New Roman", Times, serif;  
3    font-style: normal;  
4    font-size: 40px;  
5  }  
6  
7  h2 {  
8    font-style: italic;  
9    font-size: 30px;  
10 }  
11  
12 h3 {  
13   font-style: oblique;  
14   font-size: 14px;  
15 }
```

Mudando o tamanho da fonte com em

Para permitir que os usuários redimensionem o texto (no menu do navegador), muitos desenvolvedores usam em em vez de pixels. A unidade de tamanho é recomendada pelo W3C. 1em é igual ao tamanho da fonte atual. O tamanho do texto padrão nos navegadores é de 16px. Então, o tamanho padrão de 1em é 16px. Tamanho pode ser calculado de pixels para em usando esta fórmula: $pixels / 16 = em$

```
1  h1 {  
2    font-family: "Times New Roman", Times, serif;  
3    font-style: normal;  
4    font-size: 2.5em; /* 40px/16=2.5em */  
5  }  
6  
7  h2 {  
8    font-style: italic;  
9    font-size: 1.875em; /* 30px/16=1.875em */  
10 }  
11  
12 h3 {  
13   font-style: oblique;  
14   font-size: 0.875em; /* 14px/16=0.875em */  
15 }
```

Font Weight

A propriedade **font-weight** especifica a carga de uma fonte, deixando-a normal ou em negrito

```

1  h1 {
2      font-family: "Times New Roman", Times, serif;
3      font-style: normal;
4      font-size: 2.5em; /* 40px/16=2.5em */
5      font-weight: normal;
6  }
7
8  h2 {
9      font-style: italic;
10     font-size: 1.875em; /* 30px/16=1.875em */
11     font-weight: bold;
12 }

```

Font Variant

A propriedade font-variant especifica se um texto deve ou não ser exibido em uma fonte 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.

```

14 h3 {
15     font-style: oblique;
16     font-size: 0.875em; /* 14px/16=0.875em */
17     font-variant: normal;
18 }
19
20 h4 {
21     font-variant: small-caps;
22 }

```

Abaixo temos uma tabela com algumas propriedades das fontes

Propriedade	Descrição
font	Define todas as propriedades da fonte em uma declaração
font-family	Especifica a família de fontes para texto
font-size	Especifica o tamanho da fonte do texto
font-size	Especifica o estilo da fonte para o texto
font-variant	Especifica se um texto deve ou não ser exibido em uma fonte small-caps
font-weight	Especifica a carga de uma fonte