

02  
**Vídeo 2**

## Transcrição

Analizando o menu principal da página do Apeperia, vemos que ele recebeu o nome de `menu-opcoes`:

The screenshot shows a browser window displaying the Apeperia website. The navigation bar at the top has four items: 'sobre', 'planos', 'blog', and 'contato'. Below the navigation bar, there is a large image of a coffee cup. The developer tools are open, specifically the 'Elements' tab. In the HTML pane, the navigation bar is represented by the following code:

```
<nav class="menu-opcoes">
  <ul>
    <li><a href="#">sobre</a></li>
    <li><a href="#">planos</a></li>
    <li><a href="#">blog</a></li>
    <li><a href="#">contato</a></li>
  </ul>
</nav>
```

A red box highlights the `nav` element with the class `menu-opcoes`. In the CSS pane, the following styles are defined:

```
.menu-opcoes a {
  color: white;
  text-decoration: none;
}
```

Acho que o nome não é muito apropriado, por isso, quero substitui-lo para `menu-principal`. Mudaremos isso no arquivo `index.html`:

```
<nav class="menu-principal">
  <ul>
    <li><a href="#">sobre</a></li>
    <li><a href="#">planos</a></li>
    <li><a href="#">blog</a></li>
    <li><a href="#">contato</a></li>
  </ul>
</nav>
```

No entanto, se visualizarmos a página no navegador, veremos que estragamos o menu:



Falta atualizarmos o CSS com as alterações. Certamente no arquivo `estilos.less`, a classe recebeu o nome de `.menu-opcoes`:

```
.menu-opcoes {
  position: absolute;
  right: 0;
  top: -.5em;
  font-size: 1.2em;
  font-weight: lighter;
}
```

Teremos que alterar o nome em várias partes no código. Se fizermos tudo corretamente, o menu voltará a ser o que era antes da mudança no nome. Mas deve ter uma maneira menos trabalhosa de fazer isso. Vamos descobrir como:

Se analisarmos o código, veremos que a `ul` do menu está dentro da classe `.menu-principal`, assim como a `li`, `a` e o `a:hover`:

```
.menu-principal {
  position: absolute;
  right: 0;
  top: -.5em;
  font-size: 1.2em;
  font-weight: lighter;
}

.menu-principal ul {
  padding-left: 0;
}

.menu-principal li {
  display: inline-block;
  width: 5em;
}

.menu-principal a {
```

```

color: white;
text-decoration: none;
}

.menu-principal a:hover {
  text-decoration: underline;
}

```

Mas queremos apenas indicar que a `ul`, por exemplo, é filha do `menu-principal`:

```

.menu-principal {
  position: absolute;
  right: 0;
  top: -.5em;
  font-size: 1.2em;
  font-weight: lighter;
}

ul filha do cara de cima{
  padding-left: 0;
}

```

No HTML, resolveríamos isso facilmente colocando a `ul` dentro das tags de abertura e fechamento, como vemos abaixo um trecho do `index.html`:

```

<nav class="menu-principal">
  <ul>
    <li><a href="#">sobre</a></li>
    <li><a href="#">planos</a></li>
    <li><a href="#">blog</a></li>
    <li><a href="#">contato</a></li>
  </ul>
</nav>

```

Porém, no CSS não conseguimos **aninhar** setores dentro de outros. A boa notícia é que com o LESS conseguimos!

Vamos adiciona a `ul` dentro do colchete do `.menu.principal`:

```

.menu-principal {
  position: absolute;
  right: 0;
  top: -.5em;
  font-size: 1.2em;
  font-weight: lighter;

  ul {
    padding-left: 0;
  }
}

```

No Terminal, não teremos problema com a compilação. No CSS, o trecho do código aparecerá repetido, mas tudo funciona corretamente.

```
.menu-principal {
  position: absolute;
  right: 0;
  top: -.5em;
  font-size: 1.2em;
  font-weight: lighter;
}

.menu-principal ul {
  padding-left: 0;
}

.menu-principal ul {
  padding-left: 0;
}
```

Então, conseguimos fazer aninhamento (em inglês, usamos o termo *nesting*) dos seletores no CSS. Vamos colocar dentro do `.menu-principal` os seguintes trechos que também são filhos da classe:

```
.menu-principal {
  position: absolute;
  right: 0;
  top: -.5em;
  font-size: 1.2em;
  font-weight: lighter;
}

ul {
  padding-left: 0;
}
```

Indicamos que a `ul`, por exemplo, é filha do `menu-principal`. Vamos colocar os outros elementos dentro da classe:

```
.menu-principal {
  position: absolute;
  right: 0;
  top: -.5em;
  font-size: 1.2em;
  font-weight: lighter;

  ul {
    padding-left: 0;
  }

  li {
    display: inline-block;
    width: 5em;
  }

  a {
    color: white;
    text-decoration: none;
```

```

        }
    }

    a:hover {
        text-decoration: underline;
    }
}

```

Se algum dia precisar alterar o nome `.menu-principal`, só precisaremos mudar no início. Por exemplo, se quisermos chamá-la de `menu-legal`, no LESS faremos assim:

```

.menu-legal {
    position: absolute;
    right: 0;
    top: -.5em;
    font-size: 1.2em;
    font-weight: lighter;

    ul {
        padding-left: 0;
    }
/* ... */
}

```

Já o CSS, ficará assim:

```

.menu-legal {
    position: absolute;
    right: 0;
    top: -.5em;
    font-size: 1.2em;
    font-weight: lighter;
}

.menu-legal ul {
    padding-left: 0;
}
/* ... */
}

```

Mas deixaremos o nome da classe como `menu-principal`.

Porém, observe que temos o `a` e o `a:hover`. Para que não haja a repetição do `a` dentro da classe, podemos colocar o `:hover` como um filho do `a`. Para isto, iremos usar o `&:hover`.

```

a {
    color: white;
    text-decoration: none;

    &:hover {
        text-decoration: underline;
    }
}

```

Após salvarmos as alterações, iremos visualizar o trecho no código da seguinte forma:

```
.menu-principal a {  
    color: white;  
    text-decoration: none;  
}  
  
.menu-principal a:hover {  
    text-decoration: underline;  
}
```

Então, o nosso arquivo `estilos.less` ficou assim:

```
.menu-principal {  
    position: absolute;  
    right: 0;  
    top: -.5em;  
    font-size: 1.2em;  
    font-weight: lighter;  
  
    ul {  
        padding-left: 0;  
    }  
  
    li {  
        display: inline-block;  
        width: 5em;  
    }  
  
    a {  
        color: white;  
        text-decoration: none;  
  
        &:hover {  
            text-decoration: underline;  
        }  
    }  
}
```

O `nesting` é um recurso bastante útil nos pré-processadores e está disponível no LESS também. Ele facilita bastante para evitar repetições no nosso código.