

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<link rel="stylesheet" href="sass/css/nesting.css">
<title>nesting</title>
</head>
<body>

<!--
```

Nesting

O recurso “nesting” ou “aninhamento” do Sass, é um recurso que tem como objetivo deixar a vida do desenvolvedor mais fácil, pois ele organiza melhor a escrita de estilos CSS e evita a repetição desnecessária de escrita de seletores. Com o “nesting” é possível escrever blocos de estilo um dentro do outro deixando o CSS mais próximo da estrutura de estilo em árvore do HTML (DOM).

Perigo do Nesting!

A medida que você se acostuma a usar o “nesting”, é comum o desejo de querer usar o “nesting” em todo e qualquer caso. Porém, isso poderá se tornar um problema quando o arquivo CSS for gerado a partir de um arquivo Sass, pois, cada aninhamento gerado pelo “nesting” produz um seletor específico, ou vários seletores, que por sua vez irão gerar arquivos bem maiores do que o esperado.

Isso acabará dificultando a nossa vida na hora de dar manutenção no código, poderemos criar um seletor CSS tão específico e complexo, que se torna impossível dar manutenção nele, e até pode atrapalhar outras funcionalidades do nosso estilo. Além do fato de que, quanto maior o arquivo CSS gerado, mais ele vai precisar de banda de conexão para ser baixado no navegador.

Podemos usar: Lista de Seletores!

Com o “nesting” é possível fazer aninhamento de lista de seletores separados por vírgula que compartilham de um mesmo

bloco de execução, como por exemplo:

```
body, div, article {  
    background-color: red;  
}
```

Quando o Sass encontra esse tipo de aninhamento, ele é inteligente o suficiente para criar uma lista de seletores também no CSS final gerando um código menos inchado.

Podemos usar: Combinadores de Seletor!

Assim como no CSS usamos símbolos para gerar combinação de seletores entre um elemento Descendente, Filho Direto, Irmão Geral, Irmão Adjacente e etc, podemos usar esses mesmos combinadores com “nesting”. Podemos fazer isso da seguinte maneira:

- Colocando o seletor de combinação na frente do elemento filho:

```
body {  
    background-color: red;  
    > div {  
        border: 1px solid blue;  
    }  
    img {  
        width: 100px;  
    }  
}
```

...Somente a div terá um fundo vermelho.

- Colocando o seletor de combinação depois do elemento pai:

```
body > {  
    background-color: red;  
    div {  
        border: 1px solid blue;  
    }  
}
```

```
    }  
    img {  
        width: 100px;  
    }  
}
```

...Todos os elementos filho de body terão um fundo vermelho.

Podemos usar: Agrupamento de seletores!

Digamos que queremos que somente um determinado grupo de seletores recebam formatação, e eles estão bem no meio de um grupo de seletores específicos, mas queremos que os demais não peguem a formatação, somente o grupo de seletores específico que escolhemos. Neste caso podemos agrupá-los usando brackets, da seguinte maneira:

```
body {  
    background-color: red;  
    {  
        background-color: black;  
        div {  
            border: 1px solid blue;  
        }  
        img {  
            width: 100px;  
        }  
    }  
}
```

...perceba que tanto a div quando a imagem terão por padrão um fundo preto.

-->

```
<ul id="chat">
```

```
<li>
```

```
<div class="avatar">
  
</div>
<div class="message">
  <strong>John Smith</strong>
  <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Ea nemo eveniet necessitatibus, deleniti illo
    corporis enim in fugiat, modi amet maxime totam est praesentium voluptates maiores aliquid fuga, id
    tempore?
    Lorem, ipsum dolor sit amet consectetur adipisicing elit. Delectus iusto saepe exercitationem
    quibusdam dignissimos assumenda quis, aut beatae vero repellat dolor quaerat incidunt harum amet
    ratione enim vitae molestiae accusamus!
  </p>
  <time>10 minutos atrás</time>
</div>
<div class="menu">
  <button type="button">
    
  </button>
</div>

</li>

<li>

  <div class="avatar">
    
  </div>
  <div class="message">
    <strong>John Smith</strong>
    <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Asperiores voluptas quam enim neque ipsam
      fugit.</p>
    <time>10 minutos atrás</time>
  </div>
  <div class="menu">
    <button type="button">
```

```
        
      </button>
    </div>

  </li>

</ul>

</body>
</html>
```

ARQUIVO EM SCSS...

```
*{
  box-sizing: border-box;
}

body {
  margin: 0;
  font-family: sans-serif;
  color: #7f808c;
}

#chat {
  display: flex;
  flex-direction: column;
  margin: 20px;
  padding: 0;
  list-style: none;
  li {
    display: flex;
    margin-bottom: 32px;
    .avatar {
```

```
padding: 0 16px;
display: flex;
align-items: flex-end;
img{
    border-radius: 50%;
    width: 48px;
}
}
.message {
    flex: 1;
    background-color: #f5f6fa;
    padding: 16px;
    border-radius: 8px;
    border-bottom-left-radius: 0;
    strong {
        color: #242939;
    }
    p {
        font-size: 14px;
    }
    time {
        font-size: 12px;
        letter-spacing: 1px;
        opacity: .65;
    }
}
}
.menu {
    display: flex;
    align-items: center;
    button {
        border: none;
        background: none;
        outline: none;
        img {
            width: 16px;
            opacity: .5;
        }
    }
}
```

```
        cursor: pointer;
    }
}
}
```

ARQUIVO EM CSS...

```
* {
  box-sizing: border-box;
}

body {
  margin: 0;
  font-family: sans-serif;
  color: #7f808c;
}

#chat {
  display: flex;
  flex-direction: column;
  margin: 20px;
  padding: 0;
  list-style: none;
}

#chat li {
  display: flex;
  margin-bottom: 32px;
}

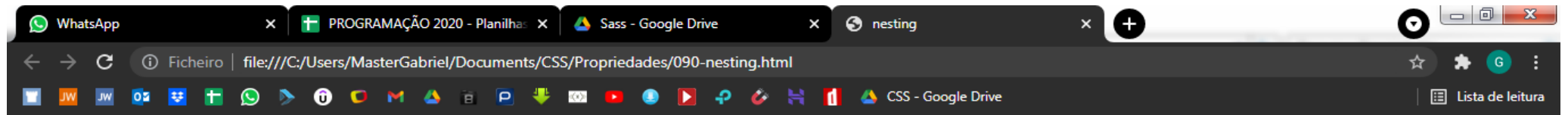
#chat li .avatar {
  padding: 0 16px;
  display: flex;
  align-items: flex-end;
```

```
}  
#chat li .avatar img {  
  border-radius: 50%;  
  width: 48px;  
}  
#chat li .message {  
  flex: 1;  
  background-color: #f5f6fa;  
  padding: 16px;  
  border-radius: 8px;  
  border-bottom-left-radius: 0;  
}  
#chat li .message strong {  
  color: #242939;  
}  
#chat li .message p {  
  font-size: 14px;  
}  
#chat li .message time {  
  font-size: 12px;  
  letter-spacing: 1px;  
  opacity: 0.65;  
}  
#chat li .menu {  
  display: flex;  
  align-items: center;  
}  
#chat li .menu button {  
  border: none;  
  background: none;  
  outline: none;  
}  
#chat li .menu button img {  
  width: 16px;  
  opacity: 0.5;  
  cursor: pointer;
```



```
}  
  
/*# sourceMappingURL=nesting.css.map */
```

RESULTADO NO NAVEGADOR...



John Smith

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ea nemo eveniet necessitatibus, deleniti illo corporis enim in fugiat, modi amet maxime totam est praesentium voluptates maiores aliquid fuga, id tempore? Lorem, ipsum dolor sit amet consectetur adipiscing elit. Delectus iusto saepe exercitationem quibusdam dignissimos assumenda quis, aut beatae vero repellat dolor quaerat incidunt harum amet ratione enim vitae molestiae accusamus!

10 minutos atrás

John Smith

Lorem ipsum dolor sit amet consectetur adipiscing elit. Asperiores voluptas quam enim neque ipsam fugit.

10 minutos atrás

