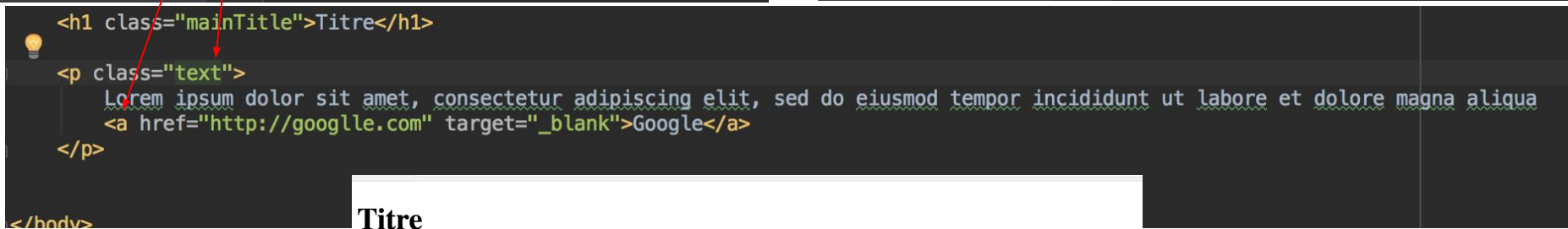
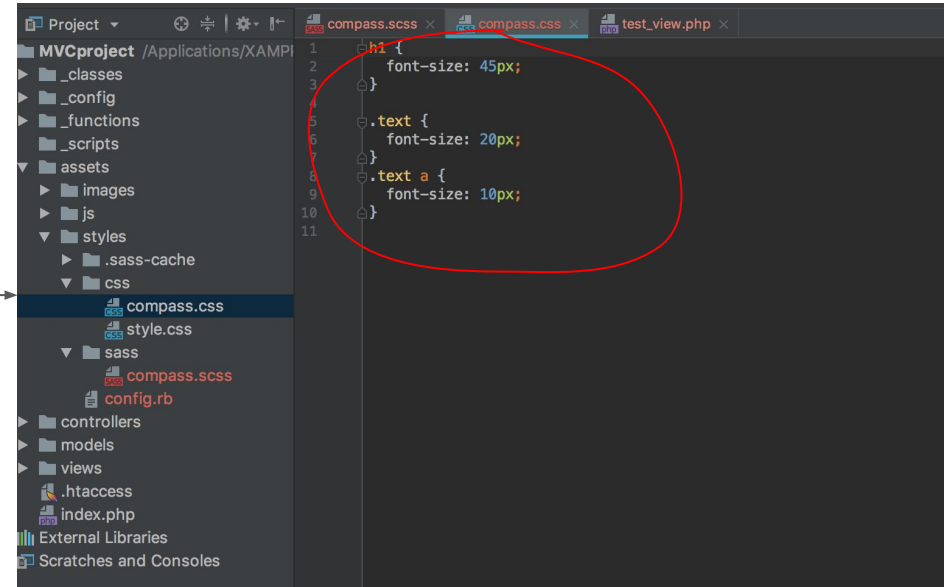
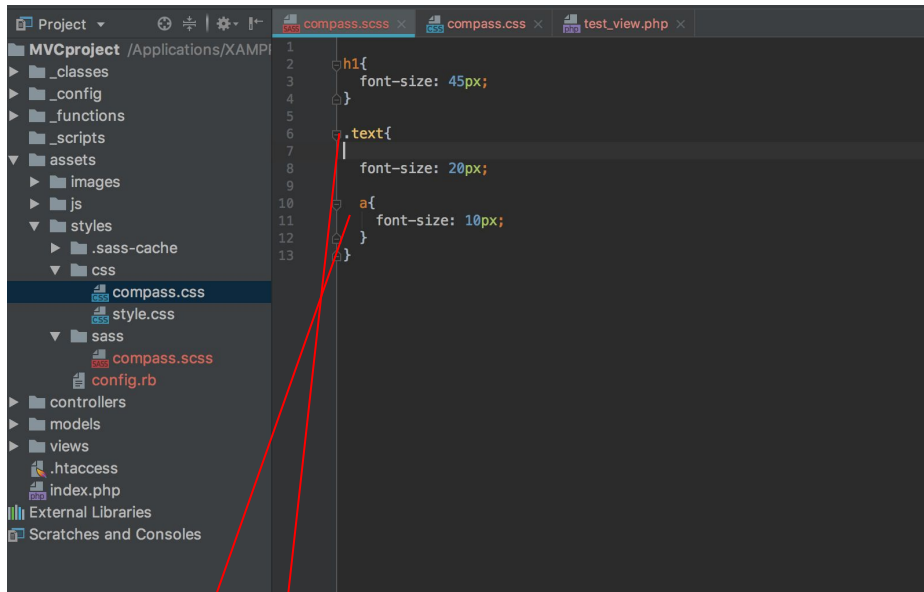


Le framework permet de comprendre et d'aller chercher les balises imbriquée tout seul



Titre

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua [Google](http://google.com)

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Test</title>
5   <meta charset="UTF-8"/>
6   <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
7   <link rel="stylesheet" href="assets/styles/css/compass.css">
8 </head>
9 <body>
10
11   <h1 class="mainTitle">Titre</h1>
12
13   <p class="text">
14     Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua
15     <a href="http://googlle.com" target="_blank">Google</a>
16   </p>
17
18   <p class="text red">
19     Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua
20     <a href="http://googlle.com" target="_blank">Google</a>
21   </p>
22
23
24
25 </body>
26 </html>
27

```

```

1 h1{
2   font-size: 45px;
3 }
4
5
6 .text{
7
8   font-size: 20px;
9
10  &.red{
11
12    color: red;
13  }
14
15  a{
16    font-size: 10px;
17  }
18 }

```

Titre

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua [Google](#)

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua [Google](#)

```
compass.scss x compass.css x test_view.php x
1 $mainColor : green;
2
3
4 h1{
5   font-size: 45px;
6 }
7
8 .text{
9
10  font-size: 20px;
11
12  &.red{
13
14    color: $mainColor;
15  }
16
17  a{
18    font-size: 10px;
19  }
20 }
```

Il y a possibilité de créer des variables qui peuvent être des hauteurs, couleurs etc...

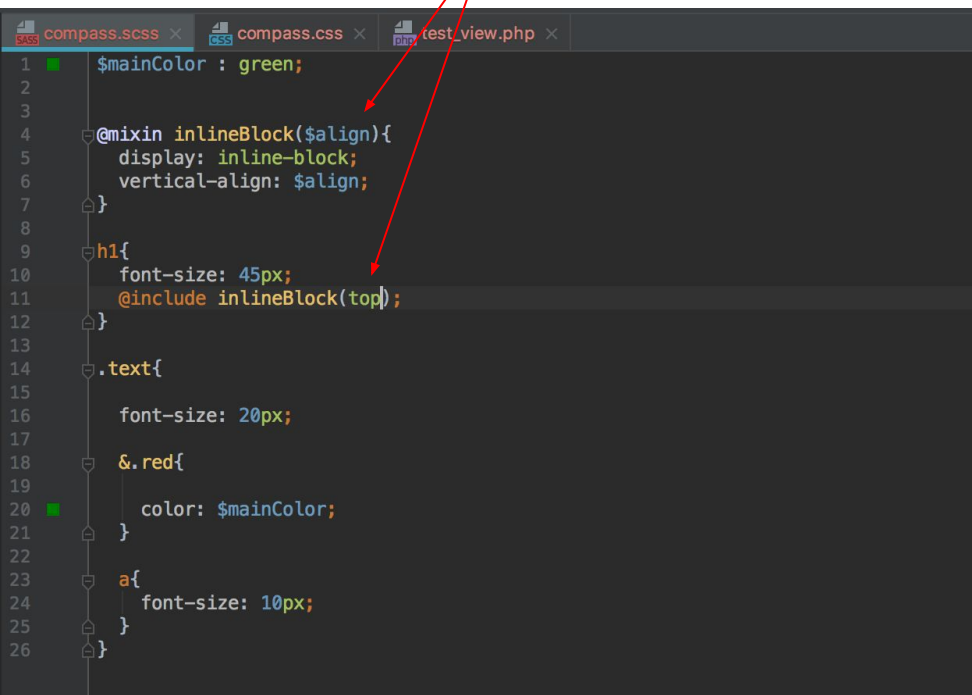


On peut créer des @mixin que l'on peut appeler ensuite dans des class et id que l'on veut.

```
compass.scss x  compass.css x  test_view.php x
1  $mainColor : green;
2
3
4  @mixin inlineBlock(){
5      display: inline-block;
6      vertical-align: top;
7  }
8
9  h1{
10     font-size: 45px;
11     @include inlineBlock();
12 }
13
14 .text{
15
16     font-size: 20px;
17
18     &.red{
19
20         color: $mainColor;
21     }
22
23     a{
24         font-size: 10px;
25     }
26 }
```

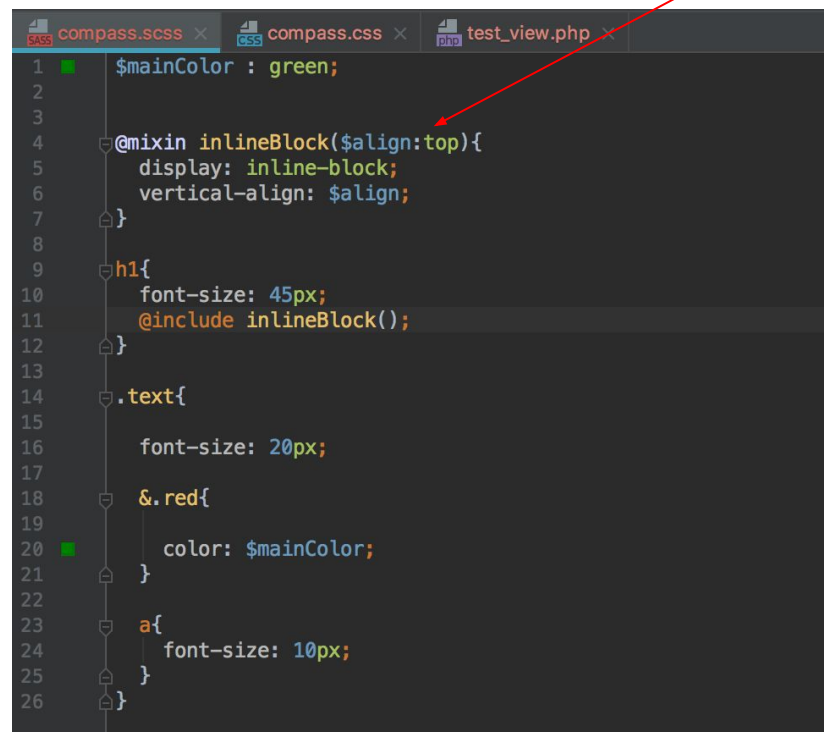
```
element.style to c
}
h1 {
    font-size: 45px;
    display: inline-block;
    vertical-align: top;
}
h1 {
    user agent s
```

On peut donner des paramètres à nos @mixin

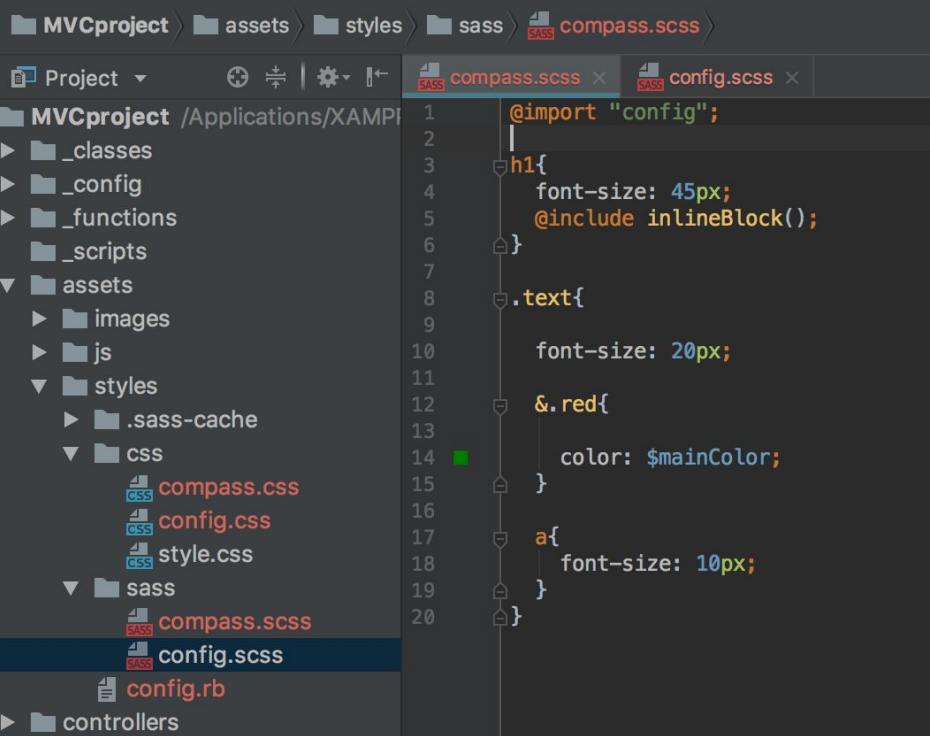


```
1 $mainColor : green;
2
3
4 @mixin inlineBlock($align){
5   display: inline-block;
6   vertical-align: $align;
7 }
8
9 h1{
10   font-size: 45px;
11   @include inlineBlock(top);
12 }
13
14 .text{
15   font-size: 20px;
16
17   &.red{
18     color: $mainColor;
19   }
20
21   a{
22     font-size: 10px;
23   }
24 }
25
26
```

On peut également donner des paramètres par défaut



```
1 $mainColor : green;
2
3
4 @mixin inlineBlock($align:top){
5   display: inline-block;
6   vertical-align: $align;
7 }
8
9 h1{
10   font-size: 45px;
11   @include inlineBlock();
12 }
13
14 .text{
15   font-size: 20px;
16
17   &.red{
18     color: $mainColor;
19   }
20
21   a{
22     font-size: 10px;
23   }
24 }
25
26
```



Il y a possibilité de mettre les @mixin et variable dans un fichier à part