

font-family: \$font-highlight;

user-select: none;

color: \$color-btn-text;

SASS



.btn {

background-color: \$color-btn-default;

background: \$color-btn-default;

SOMMAIRE

- Qu'est-ce qu'une fonction ?
- A quoi servent les fonctions Sass ?
- Utilisation des fonctions Sass prédéfinies :
 - La fonction Sass random()
 - La fonction round()
 - La fonction adjust-color()
 - Les fonctions darken() et lighten()
 - La fonction « @import »
- Création de fonctions Sass

QU'EST-CE QU'UNE FONCTION ?

- Bloc de code contenant une suite d'instructions cohérentes et dont le rôle est d'effectuer une tâche précise.
- Effectuer des calculs ou opérations dans son corps et de renvoyer une valeur en résultat. (Les calculs vont être faits en fonction de certaines données qu'on va devoir passer à la fonction pour qu'elle fonctionne. On appelle ces données des arguments.)

A QUOI SERVENT LES FONCTIONS SASS ?

En programmation, les fonctions servent généralement deux objectifs :

- Gagner du temps de développement / simplifier nos scripts ;
- Masquer la complexité de certaines opérations.

UTILISATION DES FONCTIONS SASS PRÉDÉFINIES

- Une fonction prédéfini est une fonction dont le corps (le code) a déjà été **défini et est intégré au langage.**
- Pour **utiliser une fonction prédéfinie**, il suffit de l'appeler en **utilisant** son nom en lui passant éventuellement des données dont elle a besoin pour fonctionner et qu'on appelle arguments. La **syntaxe** générale pour appeler une fonction va être la suivante : **nom_de_fonction(argument1,argument2).**

LA FONCTION SASS RANDOM()

- La fonction Sass random() retourne un nombre aléatoire. Si on utilise cette fonction sans lui passer d'argument, la fonction retournera un nombre décimal aléatoire compris entre 0 et 1. Si on lui passe un argument plus grand que 1, la fonction random() retournera un nombre entier aléatoire compris entre 1 et le nombre passé.

1

```
<? index.html x  styles.scss  # styles.css
1  <!DOCTYPE html>
2  <html lang="en">
3    <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <meta http-equiv="X-UA-Compatible" content="ie=edge">
7      <title>Cours Sass</title>
8      <link rel="stylesheet" href="css/styles.css">
9    </head>
10   <body>
11     <h1>Cours Sass</h1>
12     <p>Un cours Sass complet</p>
13   </body>
14 </html>
```

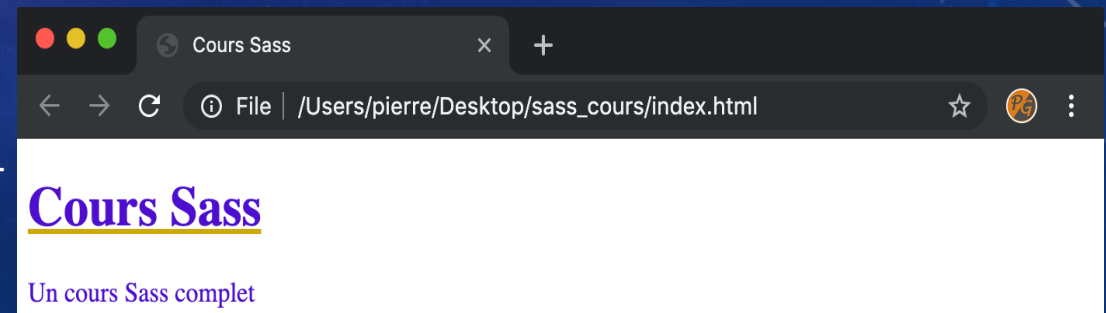
2

```
<? index.html  styles.scss x  # styles.css
1  $main-color: #005183;
2  $secondary-color: #CCAA11;
3  $random-color: rgb(random(255), random(255), random(255));
4
5  body, h1, h2, p, span, a, li {
6    color: $random-color;
7  }
8  h1 {
9    text-decoration-line: underline;
10   text-decoration-style: solid;
11   text-decoration-color: $secondary-color;
12 }
```

3

```
<? index.html  styles.scss  # styles.css x
1  body, h1, h2, p, span, a, li {
2    color: #4e0fcf;
3  }
4
5  h1 {
6    -webkit-text-decoration-line: underline;
7    text-decoration-line: underline;
8    -webkit-text-decoration-style: solid;
9    text-decoration-style: solid;
10   -webkit-text-decoration-color: #CCAA11;
11   text-decoration-color: #CCAA11;
12 }
```

4

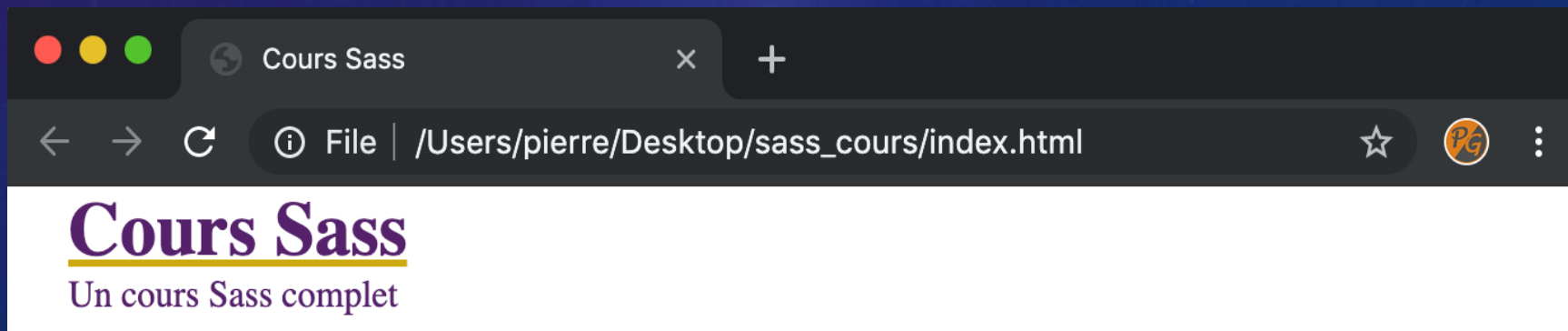


LA FONCTION ROUND()

- La fonction round() permet d'arrondir un nombre décimal à l'entier le plus proche. Cette fonction peut être très utile pour **arrondir un nombre** après un calcul dans le cas de **calculs de marges** par exemple.

```
<> index.html  styles.scss  # styles.css X
1  body, h1, h2, p, span, a, li {
2    color: #55216a;
3    margin: 0 14px;
4  }
5
6  h1 {
7    -webkit-text-decoration-line: underline;
8    text-decoration-line: underline;
9    -webkit-text-decoration-style: solid;
10   text-decoration-style: solid;
11   -webkit-text-decoration-color: #CCAA11;
12   text-decoration-color: #CCAA11;
13 }
```

```
<> index.html  styles.scss X  # styles.css
1  $main-color: #005183;
2  $secondary-color: #CCAA11;
3  $random-color: rgb(random(255), random(255), random(255));
4  $largeur: 1440;
5
6  body, h1, h2, p, span, a, li{
7    color: $random-color;
8    margin: 0 round($largeur/100)+px;
9  }
10
11 h1{
12   text-decoration-line: underline;
13   text-decoration-style: solid;
14   text-decoration-color: $secondary-color;
15 }
```



LA FONCTION ADJUST-COLOR()

- La fonction adjust-color() permet d'ajuster une couleur de base en modifiant certains de ses composants. Cette fonction va pouvoir être utile pour créer des variations d'une couleur principale par exemple.

Cette fonction peut prendre jusqu'à 5 arguments :

- soit une couleur puis trois nombres pour ajuster les canaux RGB de la couleur et enfin un niveau d'opacité ;
- soit une couleur puis trois nombres pour ajuster les canaux HSL de la couleur et enfin un niveau d'opacité.

- On va pouvoir utiliser cette fonction comme ceci (valeurs RGB) : `adjust-color(couleur, $red: nombre, $green: nombre, $blue: nombre, $alpha: nombre)`
- Ou comme cela (valeurs HSL) : `adjust-color(couleur, $hue: nombre, $saturation: nombre, $lightness: nombre, $alpha: nombre)`

```

<? index.html X  styles.scss  # styles.css
1  <!DOCTYPE html>
2  <html lang="en">
3    <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <meta http-equiv="X-UA-Compatible" content="ie=edge">
7      <title>Cours Sass</title>
8      <link rel="stylesheet" href="css/styles.css">
9    </head>
10   <body>
11     <h1>Cours Sass</h1>
12     <p>Un <a href="#">cours Sass</a> complet</p>
13   </body>
14 </html>

```

```

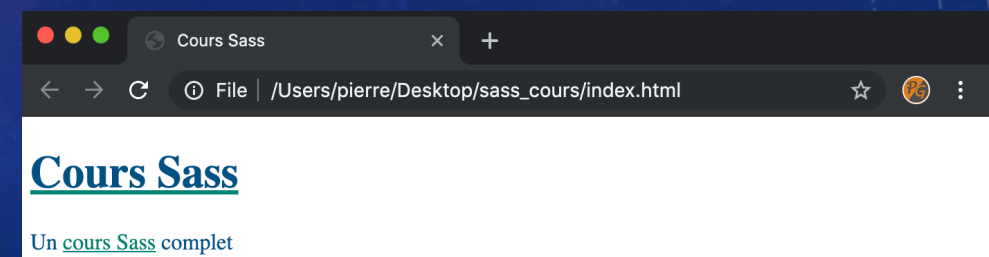
<? index.html  styles.scss X  # styles.css
1  $main-color: #005183;
2  $secondary-color: adjust-color($main-color, $green: 50, $blue: -15);
3
4  body, h1, h2, p, span, li {
5    color: $main-color;
6  }
7  a {
8    color: $secondary-color;
9  }
10 h1 {
11   text-decoration-line: underline;
12   text-decoration-style: solid;
13   text-decoration-color: $secondary-color;
14 }

```

```

<? index.html  styles.scss  # styles.css X
1  body, h1, h2, p, span, li {
2    color: #005183;
3  }
4
5  a {
6    color: #008374;
7  }
8
9  h1 {
10   -webkit-text-decoration-line: underline;
11   text-decoration-line: underline;
12   -webkit-text-decoration-style: solid;
13   text-decoration-style: solid;
14   -webkit-text-decoration-color: #008374;
15   text-decoration-color: #008374;
16 }
17 /** sourceMappingURL=styles.css.map */

```

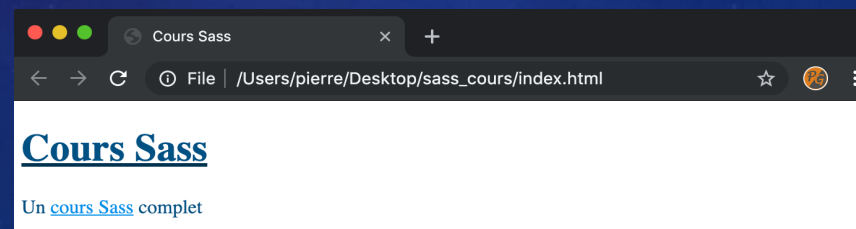


LES FONCTIONS DARKEN() ET LIGHTEN()

- Les fonctions **darken()** et **lighten()** permettent respectivement de rendre une **couleur plus foncée ou plus claire** d'un certain montant exprimé en **pourcentage**. On va passer deux arguments à ces fonctions : une **couleur de base à modifier** et un **montant compris entre 0% et 100%** représentant de combien on souhaite foncer ou éclaircir la couleur de départ.

```
<> index.html  styles.scss  # styles.css
1 $main-color: #005183;
2 $dark-main-color: darken(#005183,10%);
3 $light-main-color: lighten(#005183,20%);
4
5 body, h1, h2, p, span, li{
6   color: $main-color;
7 }
8 a{
9   color: $light-main-color;
10 }
11 h1{
12   text-decoration-line: underline;
13   text-decoration-style: solid;
14   text-decoration-color: $dark-main-color;
15 }
```

```
<> index.html  styles.scss  # styles.css
1 body, h1, h2, p, span, li {
2   color: #005183;
3 }
4
5 a {
6   color: #0090e9;
7 }
8
9 h1 {
10   -webkit-text-decoration-line: underline;
11   text-decoration-line: underline;
12   -webkit-text-decoration-style: solid;
13   text-decoration-style: solid;
14   -webkit-text-decoration-color: #003150;
15   text-decoration-color: #003150;
16 }
17 /*# sourceMappingURL=styles.css.map */
```



LA FONCTION « @IMPORT »

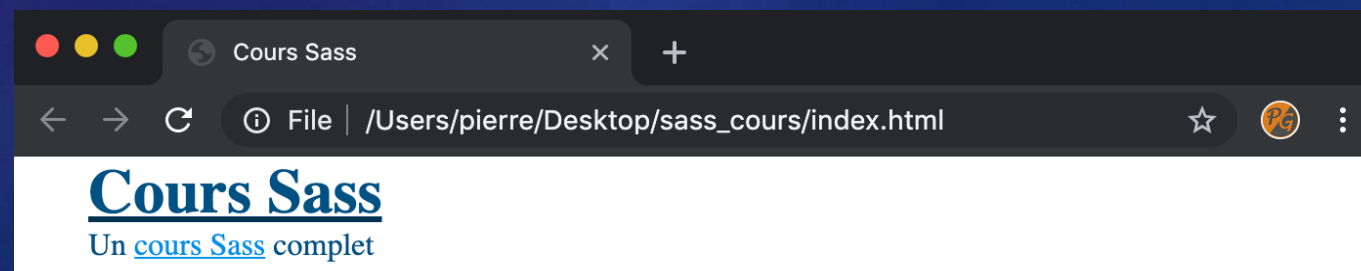
- Le CSS dispose d'une option d'importation du code à travers la règle **@import** qui nous permet d'importer des règles dans une feuille de style à partir d'autres fichiers CSS.
- Sass va récupérer le fichier qu'on souhaite importer et le combiner avec le fichier dans lequel on l'importe afin de n'envoyer au final qu'une seule feuille de style au navigateur.
- De plus, tous les mixins, fonctions ou variables définis dans la feuille qui importe et avant l'importation sont utilisables par la feuille de style importée.

CRÉATION DE FONCTIONS SASS

- On peut également créer nos propres fonctions Sass. Grâce à la règle **@function** suivi de la définition de notre fonction, c'est-à-dire de son nom, de la liste des arguments dont elle a besoin pour fonctionner et des instructions qu'elle doit exécuter.

```
index.html styles.scss x # styles.css
1 $main-color: #005183;
2 $dark-main-color: darken(#005183,10%);
3 $light-main-color: lighten(#005183,20%);
4
5 @function fois2($nb){
6   $resultat: $nb * 2;
7   @return $resultat;
8 }
9
10 body, h1, h2, p, span, li{
11   color: $main-color;
12   margin: 0 fois2(10)+px;
13 }
14 a{
15   color: $light-main-color;
16 }
17 h1{
18   text-decoration-line: underline;
19   text-decoration-style: solid;
20   text-decoration-color: $dark-main-color;
21 }
```

```
index.html styles.scss # styles.css x
1 body, h1, h2, p, span, li {
2   color: #005183;
3   margin: 0 20px;
4 }
5
6 a {
7   color: #0090e9;
8 }
9
10 h1 {
11   -webkit-text-decoration-line: underline;
12   text-decoration-line: underline;
13   -webkit-text-decoration-style: solid;
14   text-decoration-style: solid;
15   -webkit-text-decoration-color: #003150;
16   text-decoration-color: #003150;
17 }
```





SOURCE :

- <https://www.pierre-giraud.com/sass-apprendre-cours-complet/sass-fonction/>

FIN