

TP SASS

Main

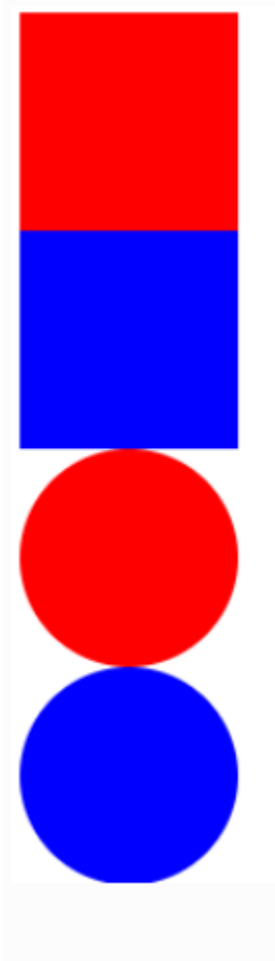
Créer un fichier index.html qui nous servira de base pour cet exercice.

```
<!DOCTYPE html>
<html lang="fr">
  <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" />
    <title>Sass</title>
    <link rel="stylesheet" href="style.css" />
  </head>
  <body>
    <div class="fig_1"></div>
    <div class="fig_2"></div>
    <div class="fig_3"></div>
    <div class="fig_4"></div>
  </body>
</html>
```

Puis, créer un “main.scss” qui servira à importer tous les autres fichiers .scss.

figures

Dans “main.scss”, modifier les classes fig_1, fig2, fig_3, fig_4 pour obtenir ceci :



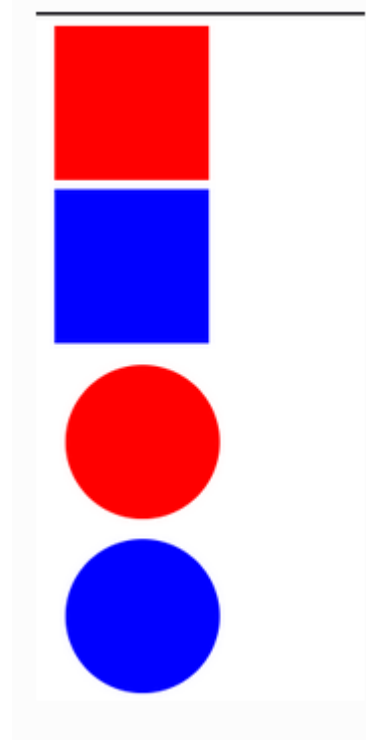
Vous devez lancer le "watcher" pour que le scss soit compiler.

Mixin

- Créer un fichier "mixin.scss", qui sera importé dans le "main.scss"
- Créer un "mixin" qui permet d'ajouter une marge en fonction d'un paramètre donné.

Par exemple, `@include marge(20)` donne une marge de 20 px tout autour de l'objet.

Pour tester votre "mixin", ajouter une marge de 20 px sur les cercles, et une marge de 10 px sur les carrés.



Lien vers les "mixins" : [Mixin](#)

Carre et cercle

- Réaliser un mixin "**square**", qui transforme une div en un carré, dont la couleur et la taille est fourni en paramètre.
- Réaliser un mixin "**circle**", qui transforme une div en un cercle, dont la couleur et la taille est fourni en paramètre.

Application, modifier les classes fig_1, fig_2, fig_3 et fig_4 en utilisant ces "mixins".

Optimisation

Créer une nouvelle mixin "**figure**" qui est capable de créer soit un cercle, soit un carré.

Ajouter un nouveau paramètre booléen, en utilisant l'algorithme suivant :

Paramètre : **flag_circle** est de type booléen

SI	flag_circle est vrai
ALORS	la div sera un cercle
SINON	ce sera un carré

Recherche : "if + sass".

Appliquer votre "mixin" dans votre css.

Variables

- Créez un fichier "variables.scss" à part
- Définissez deux variables de couleur : "**primary**" (bleu) et "**secondary**" (rouge)
- Importer le fichier "variable.scss" dans le fichier "main.scss"
- Appliquer les variables dans le code existant.
- Modifier les couleurs des variables dans variables.scss, toutes les couleurs doivent changer en même temps.

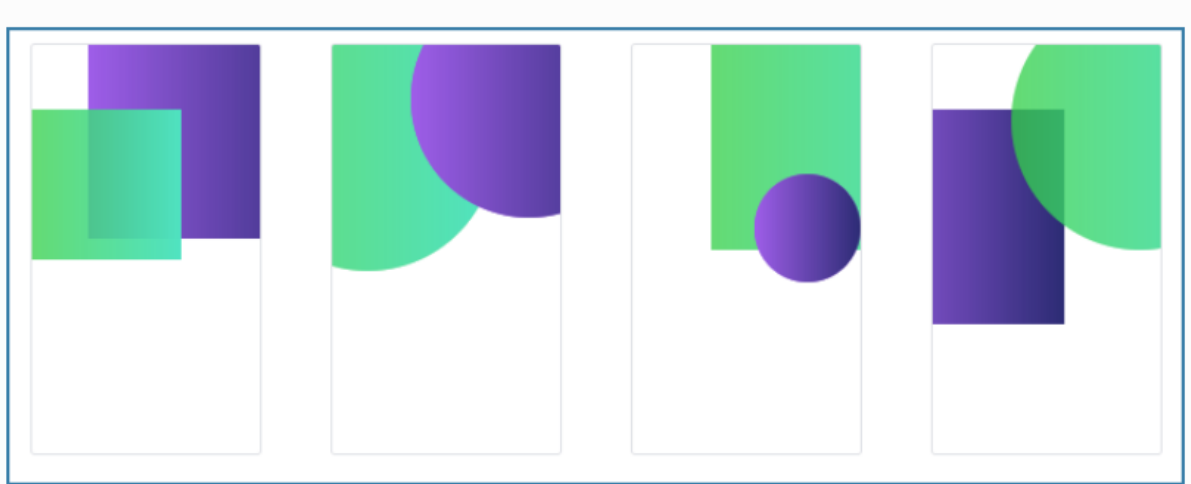
Exemple :



Cette méthode est pratique, quand le client change d'avis sur la charte graphique par exemple.

À développer

En utilisant ce que vous avez appris, et la documentation officielle, réalisez ceci :



Vous aurez besoin de créer :

- Une “mixin” capable de positionner une figure de façon absolue par rapport à son parent.
- Une “mixin” capable de créer un dégradé

code HTML

Chaque carte contient une figure majeure et une mineure.

Code HTML :

```
<section>

  <div class="card" id="card_1">

    <div class="card__maj"></div>

    <div class="card__min"></div>

  </div>

  <div class="card" id="card_2">

    <div class="card__maj"></div>

    <div class="card__min"></div>

  </div>

  <div class="card" id="card_3">

    <div class="card__maj"></div>

    <div class="card__min"></div>

  </div>

  <div class="card" id="card_4">

    <div class="card__maj"></div>

    <div class="card__min"></div>

  </div>

</section>
```

Bonus : Ceci est une version simplifiée de
<https://cssgradient.io/blog/css-gradient-text/>

Si vous êtes à l'aise avec le CSS, ajoutez les transitions, comme dans le site d'origine.