

# 4 : LES ARRIÈRES-PLANS

## Explications :

En CSS on va avoir la possibilité de gérer le fond grâce à différentes propriétés que ce soit pour le fond de toute la page qu'on pourrait alors par exemple appliquer à la balise body ou alors le fond d'une balise en particulier.

## Exemple dans body :

**style.css**

```
body{
  background-color: #bababa; //change la couleur du fond de toute
  la page car l'attribut est présent dans la balise body
  font-family: Asar, sans-serif;
  text-align: center; //texte aligné sur au centre
  color: red;
}

.taille-police{
  font-size: 18px;
  font-weight: 900;
  font-style: italic;
  text-decoration: underline;
}

.autre-taille{
  font-size: 1.2em;
  color: #3271a8;
}
```

## Résultat :



Grâce à **background-color: #5e5e5e;** (code hexadécimal de ma couleur), on peut voir que le fond de mon exemple précédent a changé.

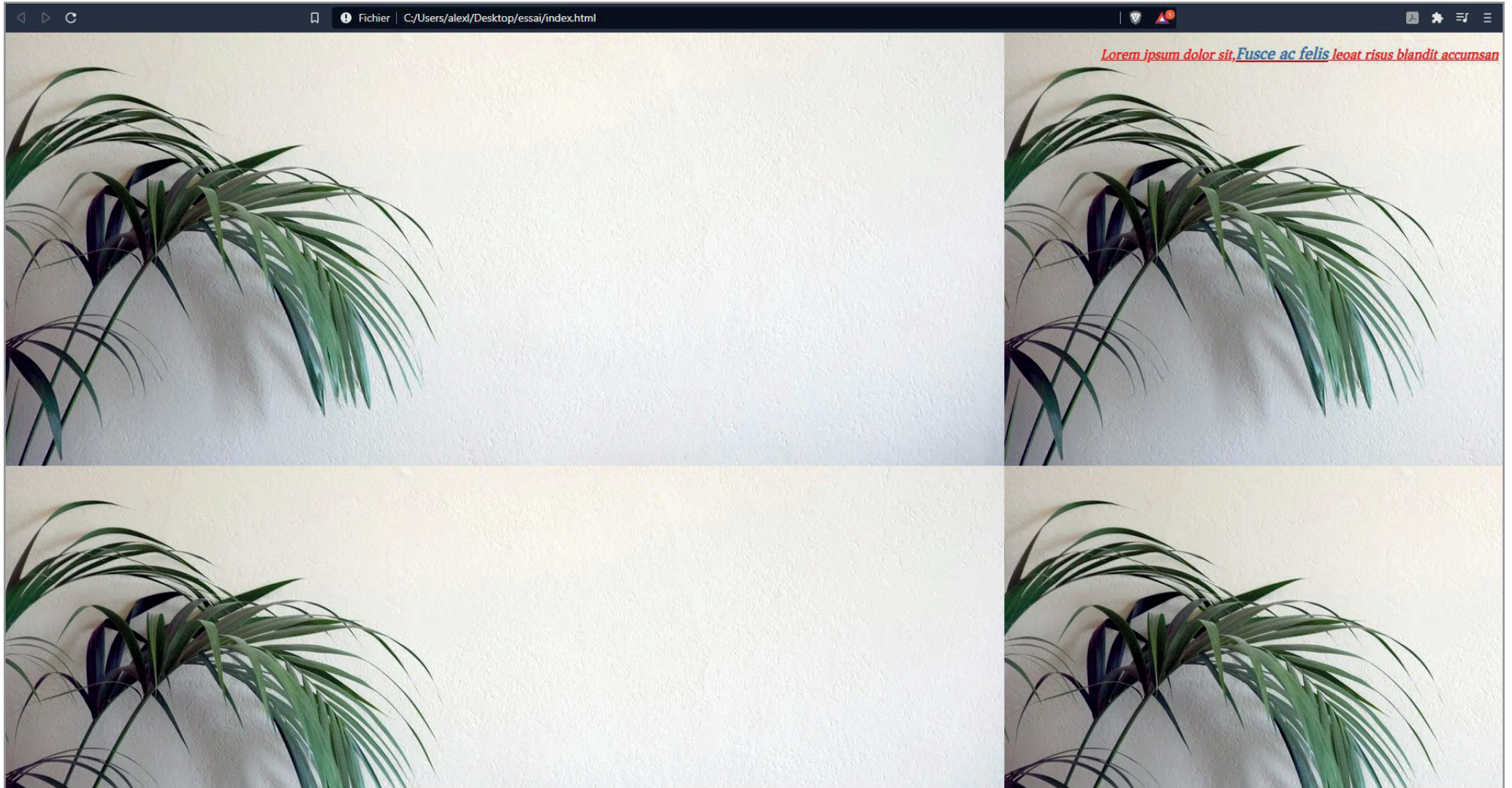
Maintenant, à défaut d'avoir une simple couleur de fond, on pourrait très bien souhaiter mettre une **image de fond**. Pour ça on peut prendre par exemple une image de fond que l'on trouve sur internet pour pouvoir copier le lien de l'image (sur google images par exemple, sélectionner une image-> clique droit dessus -> copier l'adresse de l'image

**style.css**

```
body{
  background-color: #bababa;
  background-image: url(https://cdn.pixabay.com/photo/2017/08/28/19/15/background-2691002_1280.jpg); //URL de mon image que
  je veux mettre en fond sur toute ma page
}
```

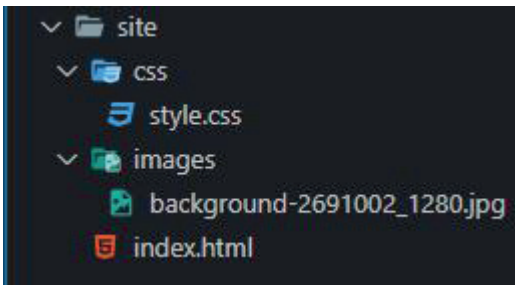
background-image à une syntaxe un petit peu particulière, en effet il ne faut pas oublier de mettre url suivi d'une parenthèse comme dans l'exemple précédent. Si l'image ne se charge pas chez l'utilisateur, il aura à la place l'attribut du background-color. Si aucun attribut background-color n'est instancié, le fond de base est blanc.

## Résultat :



On voit bien que le fond vient de s'appliquer. Vous voyez qu'il se répète plusieurs fois mais nous allons voir qu'il existe plusieurs propriétés qui vont nous permettre de gérer le fond comme on le souhaite.

À noter que dans l'exemple on peut tout à fait utiliser un fond de manière externe via une url, mais la plupart du temps on va avoir tendance à utiliser un fond qu'on va avoir directement dans nos fichiers. Pour cela on va utiliser le chemin relatif de notre fond :



Voici notre structure, nous allons donc demander à background-image d'aller chercher le fichier background-259[...].jpg qui est situé dans notre dossier images.

**style.css**

```
body{
  background-color: #bababa;
  background-image: url(../images/background-2691002_1280.jpg);
  //Chemin vers le dossier images et vers mon image
}
```

Je vais ici ne pas vous montrer le résultat car c'est le même que le résultat précédent, j'ai enregistré l'image de plante dans mon dossier images.

On va maintenant voir comment éviter que le fond se répète comme c'est le cas actuellement.

Pour ça on va utiliser encore le paramètre background car c'est à l'arrière plan qu'on s'intéresse.

On peut utiliser :

background-repeat : repeat; (va se répéter comme actuellement).

background-repeat : repeat-x;(va se répéter sur l'axe horizontal).

background-repeat : repeat-y;(va se répéter sur l'axe vertical).

background-repeat : no-repeat;(ne va pas se répéter, lorsqu'on arrive à la limite de la taille de l'image, le background-color prend le relais).

à la suite de **background-repeat**, on peut utiliser background-attachment pour faire en sorte que notre image se comporte comme on le souhaite. Je m'explique, par exemple si on écrit

**background-attachment: fixed;** l'image va rester fixe quand on va scroller dans notre page.

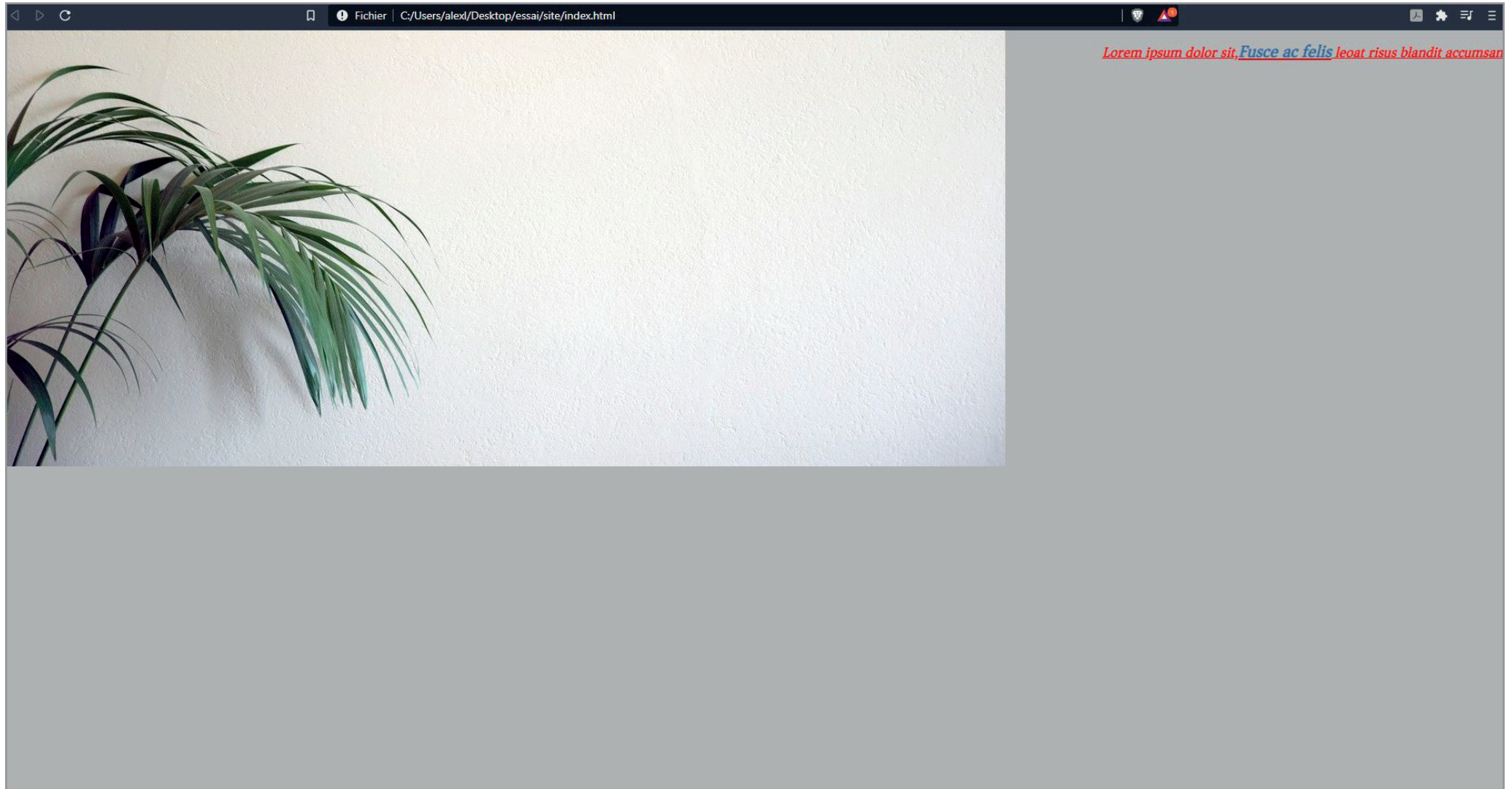
**style.css**

```
body{
  background-color: #bababa;
  background-image: url(../images/background-2691002_1280.jpg);
  background-repeat: no-repeat;
  background-attachment: fixed;
}
```



## Résultat :

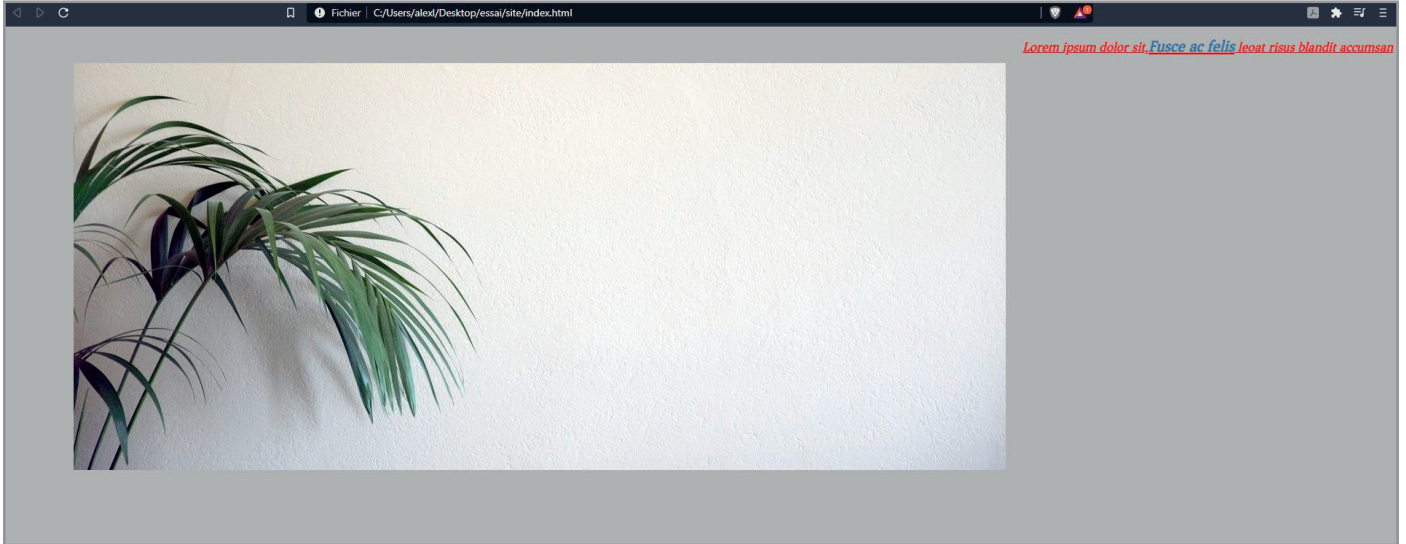
L'image ne se répète plus et si je scroll elle descendra avec moi, chose impossible à vous montrer sur une capture d'écran. Si nous avions mis `background-attachment: scroll`; l'image ne suivrait pas notre scroll et nous cesserions de la voir à force de scroller car elle restera dans le cas présent en haut de la page.



Par défaut, un fond est positionné en haut à gauche et nous allons maintenant voir comment le positionner dans notre page. La propriété **background-position** va nous permettre de jouer avec la position du background.

On va devoir lui préciser la valeur en x (la valeur horizontale) et en y (la valeur verticale). On va alors avoir plusieurs possibilités qui s'offre à nous, on peut écrire les valeurs en pixel, ce qui donnerait *par exemple* : `background-position: 100px 50px;`

Si on suit cet exemple, notre image de fond sera maintenant décalée de 100px sur l'axe horizontal et de 50px sur l'axe vertical.



À défaut de lui donner des valeurs en pixel, nous allons pouvoir également lui donner des indications en anglais pour gérer la position de notre image de fond.

Si j'écris `background-position: center center;` (2 fois center pour qu'il soit centré sur le plan vertical et horizontal) l'image sera au centre de notre page.

Si j'écris `background-position: right top;` l'image sera en haut à droite de notre page.

Et ainsi de suite si vous souhaitez d'autres positions. (ex : `left top`, `left bottom`...)

Il existe maintenant une autre propriété qui s'appelle maintenant uniquement `background` qui va nous permettre de prendre toutes les propriétés précédentes et de les mettre en une seule, cela donnerait :

**style.css**

```
body{
  /*background-color: #bababa;
  background-image: url(../images/background-2691002_1280.jpg);
  background-repeat: no-repeat;
  background-attachment: fixed;
  background-position: 100px 50px;*/

  background:#bababa url(../images/background-2691002_1280.jpg)
  no-repeat fixed 100px 50px;
}
```

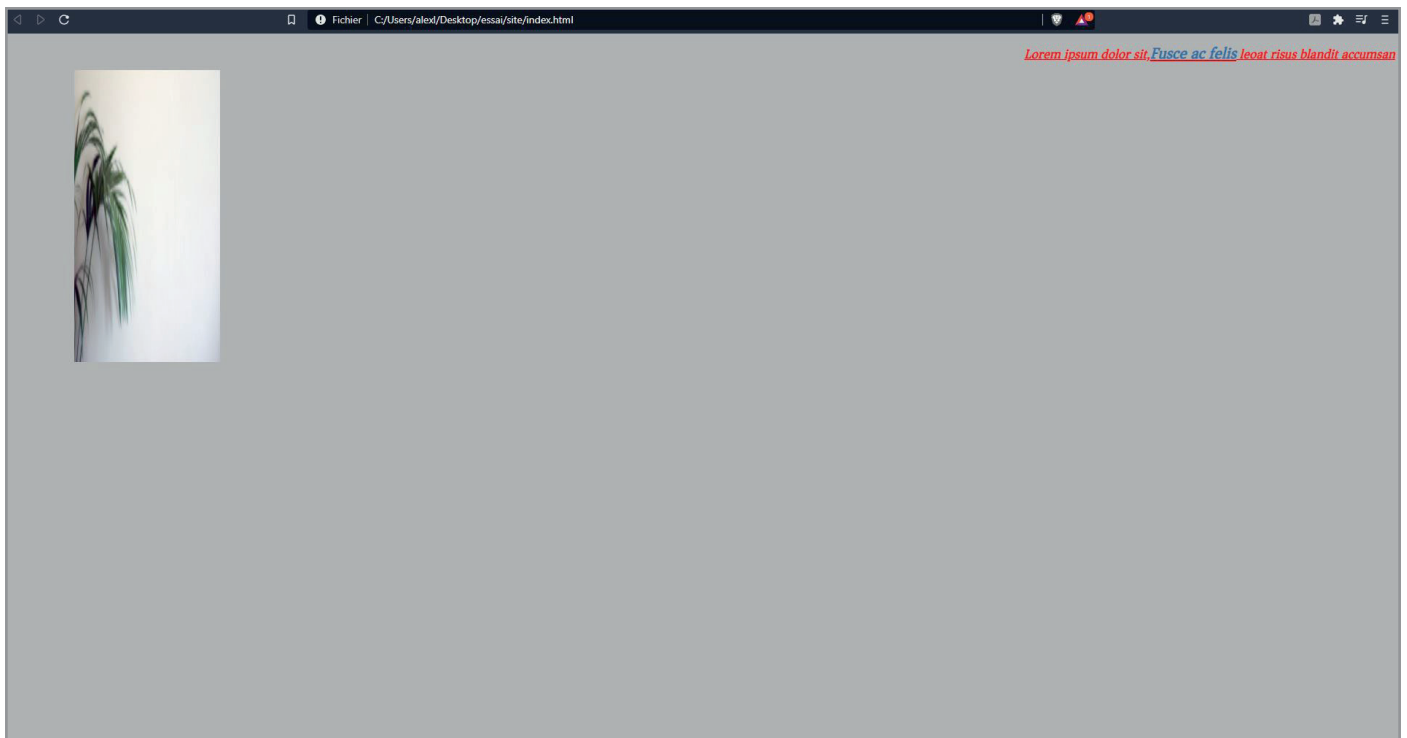
Si vous oubliez une propriété, CSS va mettre celle par défaut dans son langage, par exemple si vous enlevez le `no-repeat` dans le code rose, CSS mettra par défaut `repeat`.

Une dernière propriété intéressante est la propriété **background-size** que nous ne pouvons pas intégrer directement dans `background:#code`; on la mettra donc en dessous si on fait le choix d'écrire tout les paramètres dans `background:#code`;

**background-size** va permettre de définir la taille de notre image sur le plan horizontal et sur le plan vertical :

**style.css**

```
body{  
    background:#bababa url(../images/background-2691002_1280.jpg)  
    no-repeat fixed 100px 50px;  
    background-size: 200px 400px;  
}
```



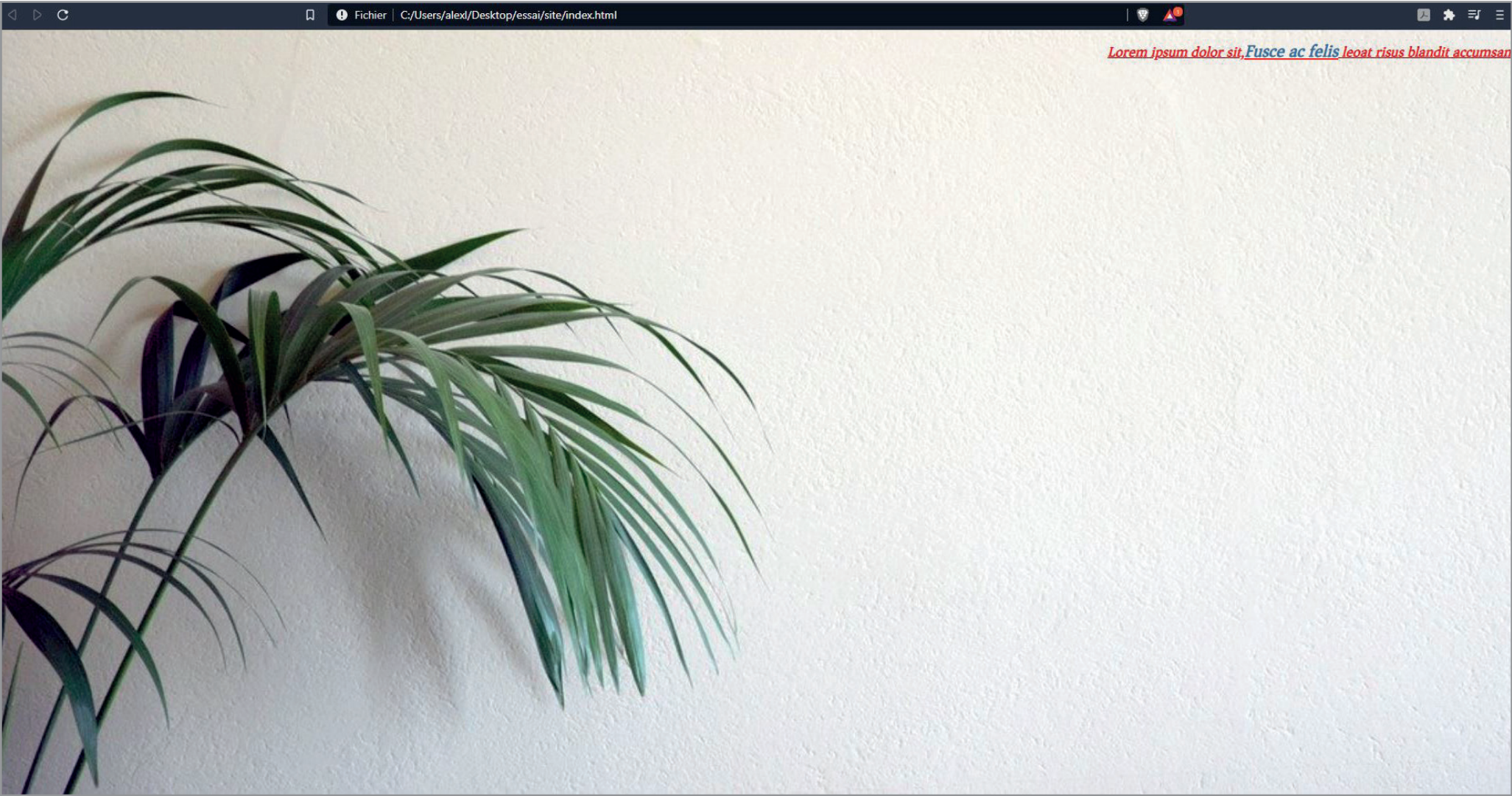
On voit bien ici que nous avons une image de fond qui à 200px de large par 400px de haut. On voit bien que le ratio de l'image n'est pas bon et on peut donc utiliser d'autre attributs tel que **background-size:cover**; qui va permettre que l'image prenne la plus grande taille qu'elle puisse prendre.

À défaut d'utiliser cover on peut avoir une autre propriété à mettre dans background-size qui est **contain**. La petite différence c'est que cover va vraiment faire en sorte que le fond prenne le plus de place possible pour combler le maximum d'espace alors que **contain** va faire en sorte que le fond prenne le plus de place possible et que l'image soit affichée dans son intégralité, ce qui n'est pas le cas de cover qui va étirer l'image si elle est trop petite.

Les 2 pages suivantes seront le résultat de ces deux attributs de background-size en enlevant `background-position:100px 50px`; pour que l'image soit positionné de base par CSS en haut à droite.



Résultat avec cover :





Résultat avec contain(on voit donc mon background-color) :

