

The Sass logo is written in a thick, pink, cursive script. The letter 'S' is large and loops around the 'a', while 'ss' follows in a fluid, connected style.

Sass



CSS with
superpowers



Sass is the most mature, stable, and powerful professional grade CSS extension language in the world.

Sass

Welcome back to the fifties



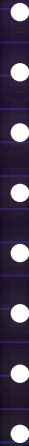
Sommaire

01

Qu'est-ce que SASS ?

A. Définition

B. Différence SASS et scss



02

Pourquoi l'utiliser ?

A. Limite du CSS

B. Avantages du SASS

C. Les options SASS

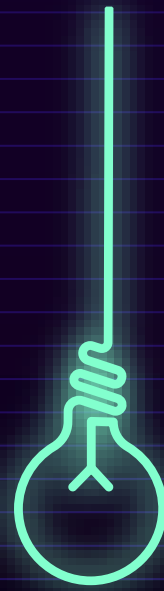
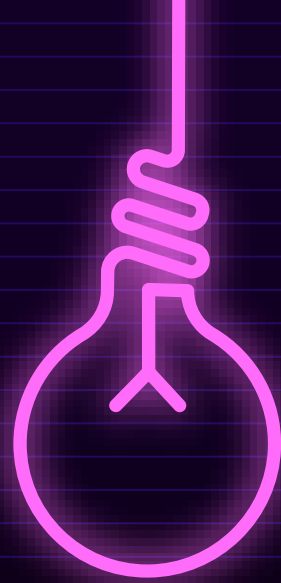
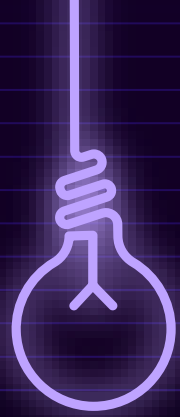
03

Comment l'utiliser?

Application VS ligne de commande

04

Conclusion

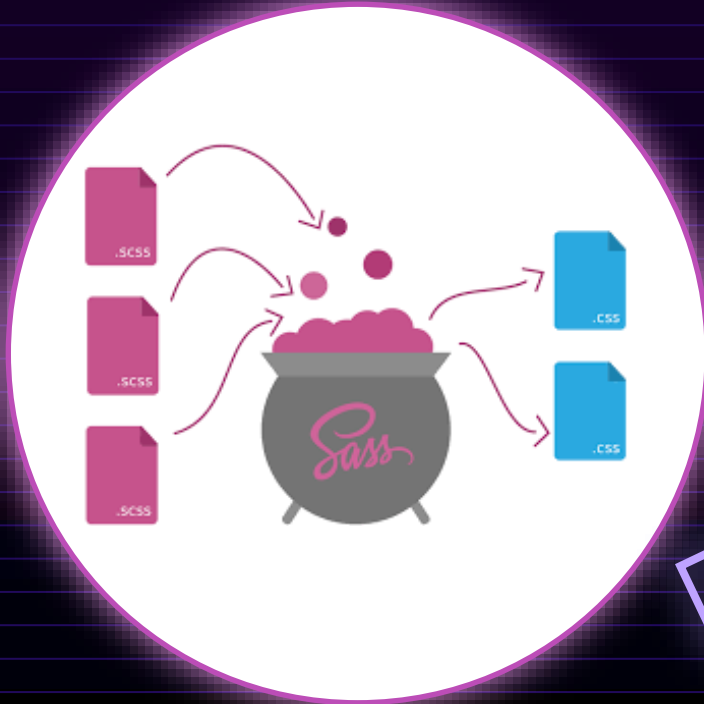


*Sass*ert à quoi?




01. Qu'est-ce que le SASS

A. Définition



SASS = Syntactically Awesome Style Sheets

PRÉPROCESSEUR =
programme qui procède à des transformations
sur un code source, avant l'étape de traduction



Le préprocesseur lit et compile le
code SASS en un fichier CSS.

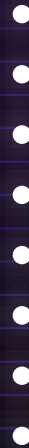
01. Qu'est-ce que le SASS

B. Différence SASS et scss

SASS

= technologie qui est utilisée
comme préprocesseur pour le css

Elle peut utiliser 2 langages
>> SassScript et le scss



SCSS

langage amélioré du
SassScript

01. Qu'est-ce que le SASS

B. Différence SASS et scss

```
1 .navigation
2   color : #333333
3   ul
4     margin : 5px
5     padding : 5px
6     li
7       list-style : none
8       a
9         color : #444444
10        text-decoration : none
11
12 .banner
13   height : 500px
14   img
15     display : block
16     width : 100%
17
```

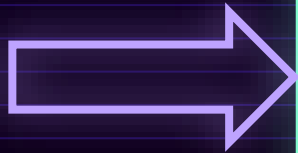
SassScript

```
1 .navigation {
2   color : □#333333;
3   ul {
4     margin : 5px;
5     padding : 5px;
6     li {
7       list-style : none;
8       a {
9         color : □#444444;
10        text-decoration : none;
11      }
12    }
13  }
14 }
15
16 .banner {
17   height : 500px;
18   img {
19     display : block;
20     width : 100%;
21   }
22 }
```

scss

02. Pourquoi l'utiliser ?

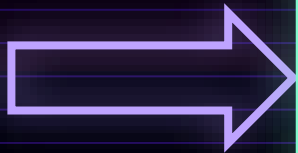
A. Les limites du css



- Sa simplicité



- La complexité des designs



- Un manque de fonctionnalités



**CSS
IS
AWESOME**



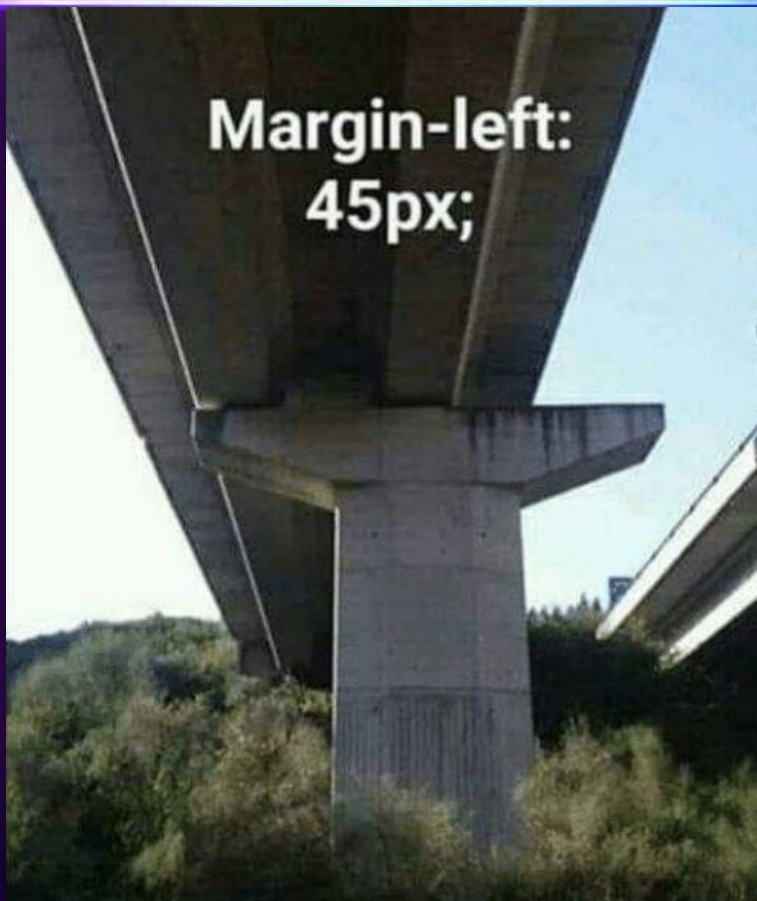
Me: Time to go back and review
Flexbox Froggy--again.



hotel-room.css ×

```
1 .air-conditioner {  
2     margin-left: -25px;  
3 }
```

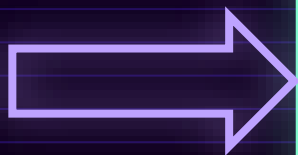
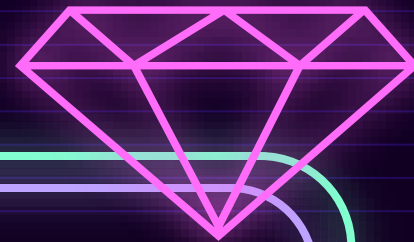
**Margin-left:
45px;**



```
8940 h3 {
8941   orphans: 3;
8942   widows: 3;
8943 }
8944 h2,
8945 h3 {
8946   page-break-after: avoid;
8947 }
8948 @page {
8949   size: a3;
8950 }
8951 body {
8952   min-width: 992px !important;
8953 }
8954 .container {
8955   min-width: 992px !important;
8956 }
8957 .navbar {
8958   display: none;
8959 }
8960 .badge {
8961   border: 1px solid #000;
8962 }
8963 .table {
8964   border-collapse: collapse !important;
8965 }
8966 .table td,
8967 .table th {
8968   background-color: #fff !important;
8969 }
```

02. Pourquoi l'utiliser ?

B. Les avantages du SASS

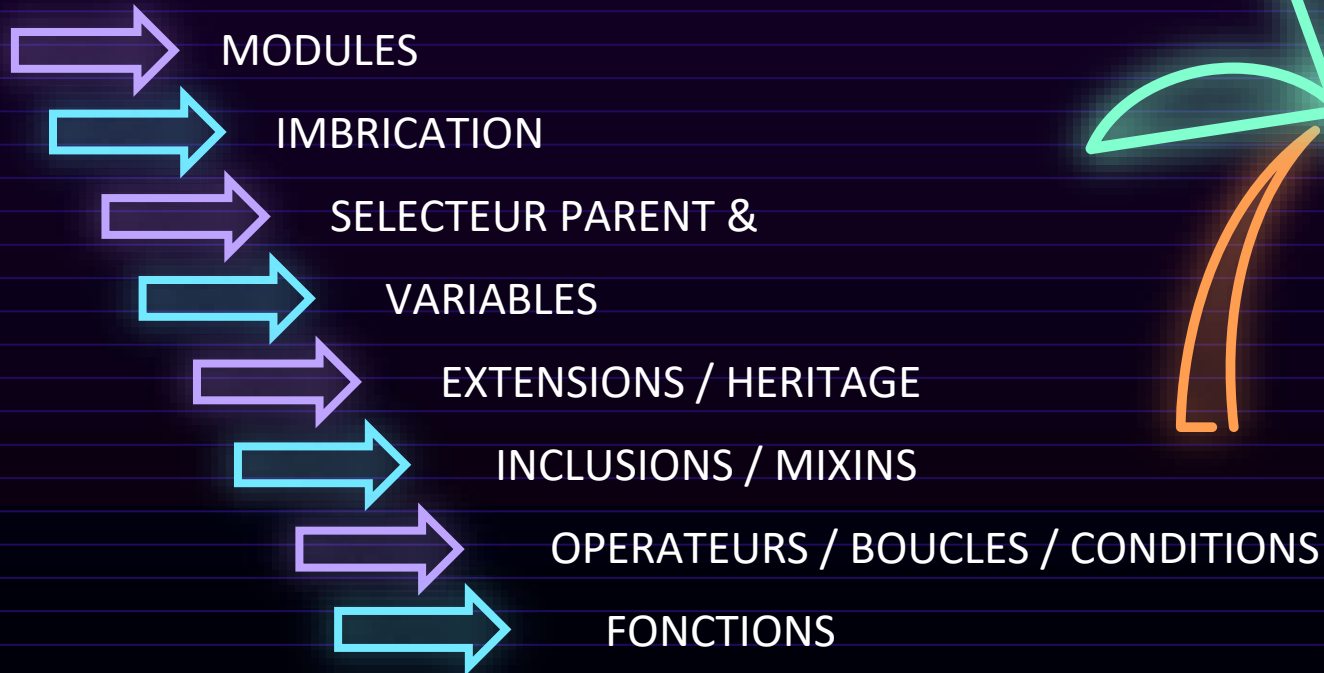


PLUS SOPHISTIQUÉ,
SIMPLIFIE ÉNORMÉMENT LE TRAVAIL DE CRÉATION DU DESIGN,
CODE MIEUX STRUCTURÉ,
MOINS D'ÉCRITURE (philosophie DRY : Don't Repeat Yourself),
ACCÉLÈRE LE DÉVELOPPEMENT...

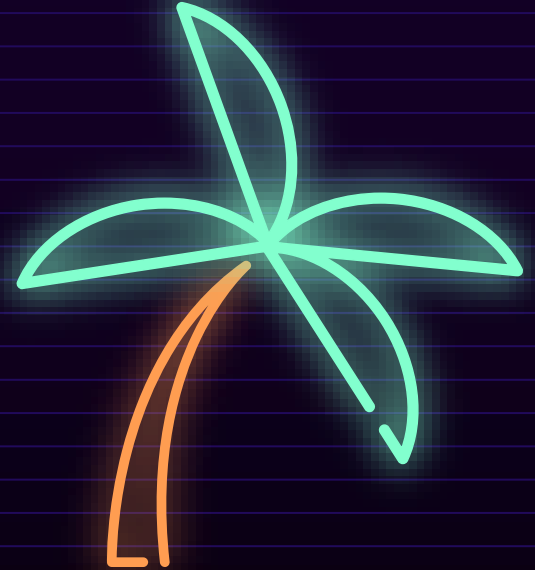
...en offrant des options supplémentaires à CSS...

02. Pourquoi l'utiliser ?

C. Les options SASS



...



02. Pourquoi l'utiliser ?

C. Les options SASS

MODULES

Création de modules qui vont contenir leur propres règles CSS pour les inclure dans d'autres

```
.container {  
  width: 80%;  
  max-width: 80%;  
}
```

module.scss

```
@use 'module';
```

```
body {  
  color: #222;  
  font-family: 'segoe UI', 'roboto', helvetica, verdana,  
  sans-serif;  
  font-size: 16px;  
}
```

style.scss



```
.container {  
  width: 80%;  
  max-width: 80%;  
}
```

style.css

```
body {  
  color: #222;  
  font-family: "segoe UI", "roboto", helvetica, verdana,  
  sans-serif;  
  font-size: 16px;  
}
```


02. Pourquoi l'utiliser ?

C. Les options SASS

IMBRICATION

si une règle est contenue dans une autre, elle sera placée sur un niveau inférieur. Elle est **encapsulée dans sa règle parente**

```
header {  
  height: 60px;  
}  
header .logo {  
  float: left;  
  width: 80px;  
  height: 100%;  
}
```

.CSS



```
header {  
  height: 60px;  
  
  .logo {  
    float: left;  
    width: 80px;  
    height: 100%;  
  }  
}
```

.SCSS

02. Pourquoi l'utiliser ?

C. Les options SASS

IMBRICATION

si une règle est contenue dans une autre, elle sera placée sur un niveau inférieur. Elle est **encapsulée dans sa règle parente**

```
.alert {  
  background-color: green;  
  color: white;  
  padding: 10px;  
}  
  
nav ul {  
  list-style: none;  
  display: flex;  
  gap: 30px;  
}  
  
nav ul li a {  
  text-decoration: none;  
  color: gray;  
}
```

.CSS



```
.alert {  
  background-color: green;  
  color: white;  
  padding: 10px;  
}  
  
nav {  
  ul {  
    list-style: none;  
    display: flex;  
    gap: 30px;  
  
    li {  
      a {  
        text-decoration: none;  
        color: gray;  
      }  
    }  
  }  
}
```

.SCSS

02. Pourquoi l'utiliser ?

C. Les options SASS

SÉLECTEUR PARENT &

Le sélecteur parent & permet d'ajouter une pseudo-classe comme un hover à son parent

```
nav ul li a {  
  text-decoration: none;  
  color: gray;  
}  
  
nav ul li a:hover {  
  color: red;  
}
```

.CSS



```
nav {  
  ul {  
    list-style: none;  
    display: flex;  
    gap: 30px;  
  
    li {  
      a {  
        text-decoration: none;  
        color: gray;  
  
        &:hover {  
          color: red;  
        }  
      }  
    }  
  }  
}
```

.SCSS

02. Pourquoi l'utiliser ?

C. Les options SASS

VARIABLES

permet de stocker des informations réutilisables à tout moment. Il suffit de la déclarer avec le signe \$

```
$background-color: #ddd;      .SCSS
$container-width: 80%;

.container {
  background-color: $background-color;
  width: $container-width;
  max-width: $container-width;
}
```



```
.container {                  .CSS
  background-color: #ddd;
  width: 80%;
  max-width: 80%;
}
```

02. Pourquoi l'utiliser ?

C. Les options SASS

EXTENSIONS / HÉRITAGE

Création de blocs de propriété pour les inclure dans d'autres règles grâce à l'`@extend`. Moins de répétitions et de recopiage de règles

```
%important {  
  font-size: 18px;  
  font-weight: bold;  
  text-decoration: underline;  
}  
  
.success {  
  @extend %important;  
  color: green;  
}  
  
.warning {  
  @extend %important;  
  color: goldenrod;  
}  
  
.error {  
  @extend %important;  
  color: darkred;  
}
```

.SCSS



```
.error, .warning, .success {  
  font-size: 18px;  
  font-weight: bold;  
  text-decoration: underline;  
}  
  
.success {  
  color: green;  
}  
  
.warning {  
  color: goldenrod;  
}  
  
.error {  
  color: darkred;  
}
```

.CSS

02. Pourquoi l'utiliser ?

C. Les options SASS

INCLUSIONS / MIXINS

Permet de créer des groupes de propriétés CSS comme pour les extensions mais de manière plus poussée (à utiliser avec parcimonie car peut alourdir le css)

```
/* Si aucune valeur n'est précisée alors $height vaut 0.8em */      .SCSS
@mixin style($style, $height : 0.8em) {
    list-style: $style;
    line-height: $height;
}

.menuList {
    @include style('none');
}
```

02. Pourquoi l'utiliser ?

C. Les options SASS

INCLUSIONS / MIXINS

Permet de créer des groupes de propriétés CSS comme pour les extensions mais de manière plus poussée

```
@mixin reset-list {  
  margin: 0;  
  padding: 0;  
  list-style: none;  
}  
  
@mixin horizontal-list {  
  @include reset-list;  
  
  li {  
    display: inline-block;  
    margin: {  
      left: -2px;  
      right: 2em;  
    }  
  }  
}  
  
nav ul {  
  @include horizontal-list;  
}
```

.SCSS



```
nav ul {  
  margin: 0;  
  padding: 0;  
  list-style: none;  
}  
nav ul li {  
  display: inline-block;  
  margin-left: -2px;  
  margin-right: 2em;  
}
```

.CSS

03. Comment l'utiliser ?

Application

"Live sass compiler" sur VScode



Ligne de cmd

Installation de SASS
avec node.js

A vertical spiral binding on the left side of a dark blue notebook cover.

04

CONCLUSION

04. Conclusion

SASS est un préprocesseur servant à lire et compiler le scss pour le rendre lisible par les navigateurs

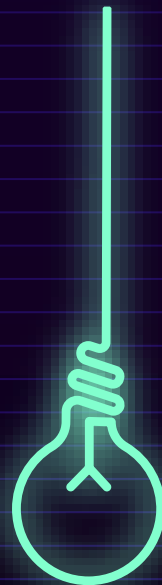
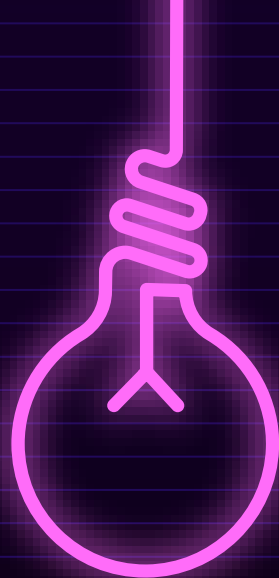
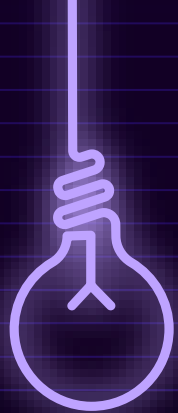
Il simplifie le travail

Le code est mieux structuré, organisé, clair, optimisé

On évite les répétitions

Les options (variables, imbrications, fonctions, conditions...) permettent de sophistication le design.





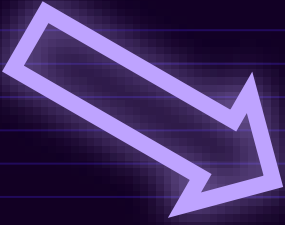
Sass 'est bien !

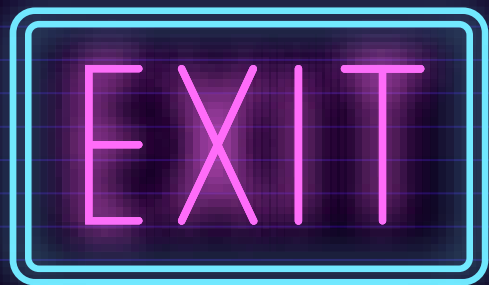


Ressources

- <https://sass-lang.com/install>
- <https://pf.vos-competences.com>
- <https://writecode.fr/>
- <https://itexpert.fr/blog/sass/>
- <https://fr.wikipedia.org/wiki/Pr%C3%A9processeur>

Merci pour
votre écoute





FOOD

