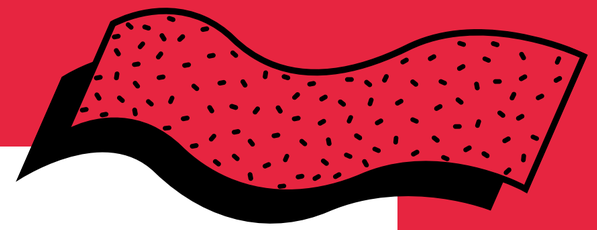

SASS

SYNTACTICALLY AWESOME STYLESHEETS

VERS LA DOC OFFICIELLE

QU'EST CE QUE LE SASS?

SASS EST UN PRÉPROCESSEUR CSS.

IL PROCÈDE À DES TRANSFORMATIONS SUR UN CODE SOURCE AVANT L'ÉTAPE DE TRADUCTION

C'EST UNE COMPILATION DANS LE CAS DE SASS OU INTERPRÉTATION POUR D'AUTRES TYPES DE PRÉPROCESSEUR

QU'EST CE QUE CA APPORTE?

Sass permet en quelques sortes “d’étendre” les possibilités du CSS classique

Permet d’’utiliser des variables, des règles imbriquées, des mixins, des fonctions, etc.

Créer un meilleur CSS, plus intuitif et mieux organisé.

La structure d’un fichier SCSS ressemble aux fichiers CSS classiques

Permet d’avoir un code plus clair et plus facilement maintenable

SASS supporte deux syntaxes différentes : SASS & SCSS

DIFFÉRENCE ENTRE SASS ET SCSS

SASS : “syntaxe indenté”, est la syntaxe originelle de Sass. Cette syntaxe utilise notamment l’indentation à la place des accolades pour spécifier la structure du document.

SCSS (Sassy CSS), est un sur-ensemble du CSS. Cela rend cette syntaxe plus simple d’utilisation que la précédente, et c’est donc sur celle-ci que ce cours sera basé.

SASS

```
$couleur: #363
.uneclasse
    color: $couleur
```

SCSS

```
$couleur: #363;
.uneclasse {
    color: $couleur;
}
```

CSS

```
.uneclasse {
    color: #363;
}
```

INSTALLATION

INSTALLER NODE JS

Sur visual studio code

CRÉER UN NOUVEAU PROJET

Dans l'explorateur de fichiers

LANCER LA COMMANDE DANS LE TERMINAL « NPM INSTALL G SASS »

Cette commande va installer Sass globalement.

ET C'EST TOUT!

Si vous préférez une installation locale, tapez simplement " npm install sass " à partir du répertoire dans lequel vous souhaitez installer Sass.

NOTE

LES NAVIGATEURS NE COMPRENNENT PAS LE CODE SASS.

NOUS ALLONS DEVOIR UTILISER UN COMPILATEUR DONT LE RÔLE VA ÊTRE DE TRADUIRE LES FICHIERS >>

.SASS OU .SCSS EN FICHIERS .CSS COMPRÉHENSIBLES PAR LE NAVIGATEUR.

COMPILER

sass --watch assets/css/style.scss:assets/css/style.css

Cette commande va servir a compiler Sass

MISE EN APPLICATION

LES VARIABLES

LES VARIABLES

- **LES VARIABLES SASS SONT TRÈS SIMPLE D'UTILISATION**

- **IL SUFFIT D'UTILISER UN SIGNE $ SUIVI D'UN NOM**

```
$primaire : blue;
$secondaire : lightblue;
$font-familly : Roboto, sans-serif,

h1{
 color : $primaire;
font-familly : $font-familly;
}
```

L'IMBRICATION & LE SÉLECTEUR PARENT

L'IMBRICATION & LE SÉLECTEUR PARENT

- SASS PROPOSE L'IMBRICATION ET DONC LA HIÉRARCHISATION DE LA STRUCTURE D'UNE PAGE
- PERMET UNE MEILLEURE ORGANISATION ET LISIBILITÉ DU CODE CSS DANS SON ENSEMBLE
- LE SÉLECTEUR PARENT & EST UN SÉLECTEUR QUI EST UTILISÉ DANS LES SÉLECTEURS IMBRIQUÉS POUR FAIRE RÉFÉRENCE AU SÉLECTEUR EXTÉRIEUR.
- LE SÉLECTEUR PARENT N'EST AUTORISÉ QU'AU DÉBUT DES SÉLECTEURS COMPOSÉS : ON PEUT ÉCRIRE & SPAN MAIS PAS SPAN &

L'IMBRICATION & LE SÉLECTEUR PARENT

```
nav{
    background-color: $primary-color;
    width: 50%;
    height: 100px;
    display: flex;
    align-items: center;
    justify-content: center;
    ul{
        display: flex;
        flex-direction: row;
        margin: 0px;
        li{
            list-style: none;
        }
    }
}
```

L'IMBRICATION & LE SÉLECTEUR PARENT

```
a{
    display: block;
    background-color: $secondary-color;
    padding : 10px;
    margin-right: 10px;
    color: $primary-color;
    text-decoration: none;
    &:hover{
        text-transform : uppercase;
        color: $alert;
        box-shadow : 1px 1px 10px $alert-color;
    }
}
```

LE @IMPORT

LE @IMPORT

- TOUT COMME CSS, SASS PREND ÉGALEMENT EN CHARGE LE @IMPORT

- IL EST POSSIBLE DE CRÉER DE PETITS FICHIERS AVEC DES EXTRAITS CSS À INCLURE DANS D'AUTRES FICHIERS SASS :
FICHIER DE RÉINITIALISATION, VARIABLES, COULEURS, POLICES, TAILLES DE POLICE, ETC.

```
@import "variables";
@import "colors";
@import "reset";
```

LES PARTIALS
@USE

LES PARTIALS @USE

- LES PARTIELS SASS/SCSS SONT LES FICHIERS CSS QUI CONTIENNENT UN PETIT CODE SPÉCIFIQUE À UN STYLE PARTICULIER SÉPARÉ DE LA FEUILLE DE STYLE PRINCIPALE
- FOURNIT UN EXCELLENT MOYEN D'ORGANISER LE CODE CSS DANS DIFFÉRENTS FICHIERS, PUIS DE LES UTILISER PARTOUT OÙ CELA EST NÉCESSAIRE
- FEUILLE DE STYLE PRINCIPALE MOINS VOLUMINEUSE
- LES PARTIELS NE SONT PAS COMPILÉS = ACCÉLÈRE LA COMPILATION GLOBALE DU CODE SASS EN CSS

LES PARTIALS @USE

```scss
/* _couleurs.scss */
$bleu: steelblue;
$rose: deeppink;
$vert: darkseagreen;
```

```scss
/* style.scss */

@use "couleurs";

.button {
   color : $bleu;
   background-color: $rose;
}
```

L'HÉRITAGE @EXTEND

L'HÉRITAGE @EXTEND

- PERMET DE PARTAGER UN ENSEMBLE DE PROPRIÉTÉS CSS D'UN SÉLECTEUR À UN AUTRE
- EST UTILE SI VOUS AVEZ DES ÉLÉMENTS DE STYLE PRESQUE IDENTIQUES QUI NE DIFFÈRENT QUE PAR QUELQUES PETITS DÉTAILS
- PAS BESOIN DE SPÉCIFIER PLUSIEURS CLASSES POUR UN ÉLÉMENT DANS VOTRE CODE HTML

L'HÉRITAGE @EXTEND

```
.button-basic  {
 border: none;
 padding: 15px 30px;
 text-align: center;
 font-size: 16px;
 cursor: pointer;
}

.button-report  {
 @extend .button-basic;
 background-color: red;
}

.button-submit  {
 @extend .button-basic;
 background-color: green;
 color: white;
}
```

```
%button-basic  {
 border: none;
 padding: 15px 30px;
 text-align: center;
 font-size: 16px;
 cursor: pointer;
}
```

LES MIXIN
@MIXIN

LES MIXIN @MIXIN

- UN MIXIN EST UN BLOC DE CODE QUI NOUS PERMET DE REGROUPER DES DÉCLARATIONS CSS QUE NOUS POUVONS RÉUTILISER SUR NOTRE SITE.
- COMME POUR UNE FONCTION, UNE MIXIN PEUT PRENDRE DES PARAMÈTRES.
- LES TRAITS D'UNION ET LES TRAITS DE SOULIGNEMENT SONT CONSIDÉRÉS COMME IDENTIQUES. CELA SIGNIFIE QUE @MIXIN IMPORTANT-TEXT { } ET @MIXIN IMPORTANT_TEXT { } SONT CONSIDÉRÉS COMME LE MÊME MIXIN
- MIXINS PARAISSENT TRÈS SIMILAIRES À L'HÉRITAGE SASS MAIS LES CAS D'USAGE SONT DIFFÉRENTS

LES MIXIN
@MIXIN

Sans paramètres

```
@mixin dimension-container{
    display: grid;
    margin : auto;
    width : 80%;
}

.container {
    @include dimension-container;
}
```

Avec paramètres

```
@mixin center-element ($margin-top : 0px,
$margin-bottom : 0px){
    margin-right : auto;
    margin-top : $margin-top;
    margin-bottom : $margin-bottom;
}
h1{
    @include center-element (25px, 25px);
}
```

LES FONCTIONS NATIVES DE SASS

LES FONCTIONS NATIVES SASS

LA SCSS NOUS FOURNIT DES FONCTIONS PRÊTES À L'EMPLOI. NOUS N'AVONS QU'À APPELER CELLES-CI POUR EFFECTUER LES OPÉRATIONS QU'ELLES CONTIENNENT !

UNE FONCTION PRÉDÉFINIE EST UNE FONCTION DONT LE CORPS A DÉJÀ ÉTÉ DÉFINI ET EST INTÉGRÉ AU LANGAGE.

POUR APPELER UNE FONCTION, ON ÉCRIT DONC SON NOM SUIVI D'UN COUPLE DE PARENTHÈSES DANS LESQUELLES ON PEUT PLACER DES DONNÉES

LA FONCTION RANDOM()

- LA FONCTION SASS RANDOM() RETOURNE UN NOMBRE ALÉATOIRE.
- SANS ARGUMENT, LA FONCTION RETOURNERA UN NOMBRE DÉCIMAL ALÉATOIRE COMPRIS ENTRE 0 ET 1
- AVEC ARGUMENT PLUS GRAND QUE 1, ELLE RETOURNE UN NOMBRE ENTIER ALÉATOIRE COMPRIS ENTRE 1 ET LE NOMBRE PASSÉ

```
$random-color : rgb(random(255),
random(255), random(255));

div{
 background-color : $random-color;
}
```

LA FONCTION ROUND()

- PERMET D'ARRONDIR UN NOMBRE DÉCIMAL À L'ENTIER LE PLUS PROCHE
- UTILE POUR ARRONDIR UN NOMBRE APRÈS UN CALCUL

```
$largeur : 1440;

div{
 margin : 0 round($largeur/100)+px;
}
```

LA FONCTION ADJUST-COLOR()

- PERMET D'AJUSTER UNE COULEUR DE BASE EN MODIFIANT CERTAINS DE SES COMPOSANTS
- SEUL LE PREMIER ARGUMENT RELATIF À LA COULEUR DE BASE EST OBLIGATOIRE AU FONCTIONNEMENT DE LA FONCTION
- ON UTILISE CETTE FONCTION COMME CECI (VALEURS RGB) :
  ADJUST-COLOR(COULEUR, $RED: NOMBRE, $GREEN: NOMBRE, $BLUE: NOMBRE, $ALPHA: NOMBRE)
- ON UTILISE CETTE FONCTION COMME CECI (VALEURS HSL) :
  ADJUST-COLOR(COULEUR, $HUE: NOMBRE, $SATURATION: NOMBRE, $LIGHTNESS: NOMBRE, $ALPHA: NOMBRE)

LA FONCTION ADJUST-COLOR()

```
$primaire: #e72540;
$secondaire : adjust-color(#e72540, $red:20,
$green:-62);

a{
 color : $primaire;
 text-decoration-color : $secondaire;
}
```

LES FONCTIONS DARKEN() & LIGHTEN()

- PERMETTENT RESPECTIVEMENT DE RENDRE UNE COULEUR PLUS FONCÉE OU PLUS CLAIRE D'UN CERTAIN MONTANT EXPRIMÉ EN POURCENTAGE

- 2 ARGUMENTS :
  UNE COULEUR DE BASE À MODIFIER ET UN MONTANT COMPRIS ENTRE 0% ET 100%

```
$couleur: #e72540;
$couleur-foncee : darken(#e72540, 15%);
$couleur-claire : lighten(#e72540, 30%);

button{color : $couleur;}
button:hover{color : $couleur-foncee;}
button:active{color : $couleur-claire;}
```

LA FONCTION INVERT()

LA FONCTION INVERT() RENVOIE L'INVERSE OU LE NÉGATIF D'UNE COULEUR DONNÉE

```
h1{
 color: invert(white);
 color: invert(#b37399);
 color: invert(#550e0c,
20%);
}
```

```
h1 {
color: black;
color: #4c8c66;
color: #663b3a;
}
```

LA FONCTION PERSO

- ON PEUT ÉGALEMENT CRÉER NOS PROPRES FONCTIONS SASS

- IL FAUT POUR CELA UTILISER LA RÈGLE @FUNCTION SUIVI DE LA DÉFINITION DE NOTRE FONCTION

LA FONCTION PERSO

```
$light-main-color : lighten(#0005183, 20%);

@function fois2($nb){
    $resultat: $nb * 2;
    @return $resultat;
}

body, h1, h2, p, span, li{
    color: $main-color;
    margin : 0 fois2(10)+px;
}
```

FONCTION NUMÉRIQUES

Les fonctions numériques sont utilisées pour manipuler des valeurs numériques.

FONCTIONS DE CHAINES

Les fonctions de chaîne sont utilisées pour manipuler et obtenir des informations sur les chaînes. Les chaînes Sass sont basées sur 1. Le premier caractère d'une chaîne est à l'index 1, pas 0.

FONCTIONS DE LISTE

Les fonctions de liste sont utilisées pour accéder aux valeurs d'une liste, combiner des listes et ajouter des éléments aux listes. Les listes sont immuables et le premier élément de liste est à l'index 1.

FONCTION CARTE

Dans Sass, le type de données de carte représente une ou plusieurs paires clé/valeur. Il est également possible d'utiliser les fonctions Liste avec des cartes. La carte sera traitée comme une liste avec deux éléments.

FONCTIONS DE SELECTEURS

Les fonctions de sélecteur sont utilisées pour vérifier et manipuler les sélecteurs.

FONCTIONS D' INSTROSPECTION

Les fonctions d'introspection sont rarement utilisées lors de la construction d'une feuille de style. Cependant, ils sont utiles si quelque chose ne fonctionne pas correctement : comme les fonctions de débogage.

FONCTION COULEURS

les fonctions de couleur dans Sass se répartissent en trois parties : Définir les fonctions de couleur, Obtenir les fonctions de couleur et Manipuler les fonctions de couleur

POUR EN SAVOIR PLUS ...

CONDITIONS ET BOUCLES SASS

CONDITIONS ET BOUCLES SASS

- SASS MET À NOTRE DISPOSITION QUATRE RÈGLES @ NOUS PERMETTANT DE GÉNÉRER DES STRUCTURES DE CONTRÔLE QUI VONT SE MANIFESTER SOIT SOUS FORME DE CONDITIONS :

- @IF ET @ELSE PERMETTENT D'EXÉCUTER UN BLOC DE CODE SI UNE CONDITION EST REMPLIE OU D'EXÉCUTER UN AUTRE CODE SINON

- @EACH ÉVALUE UN BLOC POUR CHAQUE ÉLÉMENT D'UNE LISTE OU CHAQUE PAIRE D'UNE MAP

- @FOR ÉVALUE UN BLOC UN NOMBRE DE FOIS PRÉCISÉ LORS DE LA CRÉATION DE LA RÈGLE

- @WHILE ÉVALUE UN BLOC JUSQU'À CE QU'UNE CERTAINE CONDITION DE SORTIE SOIT REMPLIE

CONDITIONS SASS

```scss
$light: #eee;
$dark: #333 ;
@mixin theme($light-theme : true){
    @if($light-theme){
        color : $dark;
        background-color: $light;
    }@else{
        color: $light;
        background-color: $dark;
    }
}

body{
    @include theme($light-theme : true);

    span{
        @include theme($light-theme: false);
    }
}
```

BOUCLES SASS

```
$width : 1000 ;
$margin : 0 ;

@while($margin < $whidth/50){
    $margin : $margin + 1;
}

body{
    margin : 0 $margin+px;
}
```