# **MEMO SASS en SCSS**

### Installation dans le projet en cours

|  |  |
| --- | --- |
| Placer le dossier dart-sass dans le projet | Dispo sur le github de sass |
| En ligne de commande tapper  ./dart-sass/sass.bat style.scss style.css -–watch | ./dart-sass/sass.bat appel la commande  Le premier lien est le fichier à surveiller  Le deuxième lien est le fichier css a créer |

NB : le fichier .map généré sert à savoir ou est définie dans le scss une regle css (enlever pour la production)

### Principe d’imbrication

|  |  |
| --- | --- |
| **SANS imbrication = CSS** | **AVEC imbrication = SCSS** |
| .parent {  color: red;  }  .parent .child {  color: blue;  }  .parent:hover {  color: green;  } | SCSS.parent {  color: red;  .child {  color: blue;  }  & :hover{ // ne pas oublier de mettre un &  Color :green ;  }  } |

### Variable et héritage

|  |  |  |
| --- | --- | --- |
| Import de fichier | @import «style2.css» ; | Permet de séparer nos fichier scss et les regrouper pour le css |
| **Héritage** | %btn{ proriété1 : val1 ; }  .btn-fat{ @extend %btn ;} | Créer une classe générique %btn  Qu’ont integre dans la classe que l’on souhaite |
| **Variable**  **Dans sélecteur**  **Opération**  **fonction** | $maVariable : #fff5 ;  .btn{color = $maVariable}  #{$maVar}  $maVar / 2 ;  Darken($var,10) | Créer une variable $maVariable avec la valeur #fff5  Et l’attribut a la class btn  Pour utiliser une variable dans un selecteur il faut faire #{$var}  On peut faire des opération sur les variable  Il existe des fonction sass, ici darken assombris de « 10 » $var |

### Les mixins : comme un @extend mais avec des propriétés qui peuvent varier

|  |  |  |
| --- | --- | --- |
| **Déclaration** | @mixin rotate($rotation : 10deg){  Transform : rotate([$rotation) ;  } | Déclare un mixin avec une valeur par defaut de 10deg  Definie les regles |
| **Utilisation** | .maclass{  Include rotate(50deg) ;  } | Sur .maclass  On utilise la mixin rotate avec 50deg en parametre |

### Les fonctions

|  |  |  |
| --- | --- | --- |
| **déclaration** | @function rem($size, $base : 16){  @return 1rem \* $size / $base ;  } | On declare une fonction nommée rem  Qui retournera… |
| **utilisation** | .maclass{  Padding : rem(5, 16) rem(12) ;  } | Sur .maclass  On utilise la fonction rem() avec les valeur 5 et 16 |

### Les conditions, boucles

|  |  |  |  |
| --- | --- | --- | --- |
| **condition** | @if (condition){  … }@else{...} | SI (condition) { si true} sinon {si false} | |
| **Boucle for** | @for $i from 1 through 4{  .m#{$i}{instruction} etc… } | Pour i de 1 à 4  Pour chaque .m$i (m1,m2,m3 et m4) repete les instructions | |
| **Boucle each** | $animals : bear, lion, camel ;  @each $animal in $animals{.#{$animal}{…}} | | Pour chaques nom de $animals (bear, lion et camel)  Execute le code |