

Grille et sass

Développer votre grille

L'objectif est de construire une mini-grille css générique.

Définition d'une grille

Nous partons, pour nos calculs, d'une grille avec gouttières extérieures.

(vous pouvez adapter ensuite cette formule à votre guise en fonction de vos besoins)

Nous pouvons définir une grille à partir de :

- nombre de colonnes (par exemple 12)
- largeur des gouttières (par exemple 2,7%)

Le reste se déduit :

- Les autres variables de la grille :
 - nombre de gouttières = nombre de colonnes + 1
 - largeur d'une colonne = $(100\% - ((\text{nombre de gouttières}) * (\text{largeur des gouttières}))) / (\text{nombre de colonnes})$
- Les paramètres des boîtes à placer sur la grille :
 - Un décalage (offset) de n colonnes entraîne un décalage (`margin-left`) de :
 $n * (\text{largeur d'une colonne}) + (n+1) * (\text{largeur des gouttières})$
 - Un élément de taille x colonnes aura une largeur (`width` ou `flex-basis`) de
 $x * (\text{largeur d'une colonne}) + (x-1) * (\text{largeur des gouttières})$

Une grille non sémantique - génération des classes

Cette grille sera réalisée sous la forme d'un ensemble de classes css, et sera donc utilisée pour réaliser le placement d'un document html en ajoutant des attributs `class` dans les balises.

À l'aide de la directive [@for](#) de sass, et des outils de calcul proposés, générer les classes de grille, basée sur `flex`, reproduisant les classes principales de la grille `Materialize`.

Le nombre de colonnes et la largeur des gouttières doivent être déclarées sous forme de variables.

Pourquoi ?

Une première version sémantique - réutilisation des classes générées

En reprenant le fichier grille.html, et sans toucher au html (sauf pour y inclure votre feuille de styles) :

- Transformer votre grille `@for` en `partial`.
- Reconstruire votre rendu en utilisant votre grille `@for` et en réutilisant les classes souhaitées, par `@extend` à l'aide de sélecteurs.

Une version mixins sass pour html sémantique

La version 3 est une version sass de la grille, fournie principalement sous forme de mixins, permettant toujours de ne plus renseigner les éléments de grilles dans le fichier source html. Le nombre de colonnes, la largeur d'une gouttière, la taille du décalage (offset) et la largeur d'une cellule (ces deux derniers définis en nombre de colonnes) seront reçus comme des paramètres de mixins.

Ces fonctions permettent aux développeurs de bénéficier des calculs sans utiliser la grille.

Par exemple l'appel à

```
.exemple { @include grille(12, 2%, 3, 4); }
```

devra compléter les propriétés de l'élément de classe "exemple" pour lui donner une largeur et une marge gauche pour qu'il occupe 3 cases, et soit décalé de 4 cases par rapport à l'élément précédent, dans une grille de 12 colonnes avec des gouttières de 2%.