

**SUSY3**

*If you're interested in reading Sass poetry,  
be sure to look at Susy's source code!*

— Hugo Giraudel, SitePoint

*Your design, our math*

# **SUSY, LA MATHÉMATICIENNE**

Un outil «design-agnostique»!

# SYNTAXE INSPIRÉE DE CSS GRID.

`grid-template-columns` → `columns`

Tout comme `grid-template-columns`, le paramètre `columns` exige une liste de dimensions de colonnes:

## Susy 2

```
$susy: (  
  columns: 4,  
  container: 1140px  
);
```

## Susy 3

```
$susy: (  
  'columns': susy-repeat(4)  
);
```

# EXEMPLES

## GRILLE SYMÉTRIQUE

Les 2 syntaxes ont le même résultat

```
$grid-template-columns: repeat(12, 120px);  
$columns: susy-repeat(12, 120px);
```

# GRILLE ASYMÉTRIQUE

Avec unités de mesure différentes

```
$grid-template-columns: 200px repeat(4, 1fr) 30em;  
$columns: 200px susy-repeat(4) 30em;
```

**200 px**

**susy-repeat(4)**

**width: span(3)**  
**main.content**

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Mollitia recusandae sapiente voluptatum. Adipisci blanditiis dolorem error et eum explicabo pariatur vitae voluptate? Aliquid error in omnis perspiciatis voluptates. Accusamus, recusandae. Error iste laborum maxime modi nemo non nostrum odio officiis porro quis. Ab asperiores aut deserunt, distinctio, dolores eius fugiat fugit hic ipsam itaque minima neque nisi quis totam voluptatem. Deleniti dignissimos illo ipsum ullam voluptatibus. Ad amet aut cum doloremque dolores eos excepturi in inventore, repudiandae rerum! Aliquam blanditiis delectus ea explicabo, impedit nihil possimus quam sit velit voluptatibus.

**30em**

**width: span(1 at last)**  
**aside.sidebar**



# 12 COLONNES BASÉES SUR LES EM

```
$columns: susy-repeat(12, 5em);  
$columns: 5em 5em 5em 5em 5em 5em 5em 5em 5em 5em 5em 5em;
```

# LA GRILLE DITE DU SAINT GRAAL

```
$columns: 12em 1 200px;
```

## main.content

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Mollitia recusandae sapiente voluptatum. Adipisci blanditiis dolorem error et eum explicabo pariatur vitae voluptate? Aliquid error in omnis perspiciatis voluptates. Accusamus, recusandae. Error iste laborum maxime modi nemo non nostrum odio officiis porro quis. Ab asperiores aut deserunt, distinctio, dolores eius fugiat fugit hic ipsam itaque minima neque nisi quis totam voluptatem. Deleniti dignissimos illo ipsum ullam voluptatibus. Ad amet aut cum doloremque dolores eos excepturi in inventore, repudiandae rerum! Aliquam blanditiis delectus ea explicabo, impedit nihil possimus quam sit velit voluptatibus.

## aside.sidebar

# + DE COLONNES DANS LA PARTIE FLUIDE DU SAINT GRAAL

```
$columns: 12em susy-repeat(4) 200px;  
$columns: 12em 1 1 1 1 200px;
```

# RÉPÉTITION D'UNE ALTERNANCE DE COLONNES

```
$columns: susy-repeat(3, 8em 200px);  
$columns: 8em 200px 8em 200px 8em 200px;
```

# SUSY UTILISE LA FONCTION CALC()

Par exemple:

```
[Réglage] 200px susy-repeat(4) 30em  
[SCSS] width : span(3);  
[CSS] width : calc(200px + ((100% - (200px + 30em)) / 5.25 * 2.5))
```

# SUSY 3, C'EST:

- **SU**
  - LE NOYAU, LE MODULE DE MATHS
- **SUSY**
  - L'API POUR UTILISER SU

**SPAN ( ) ET GUTTER ( )**

sont les 2 principales fonctions de l'API



# PLUS QUE 4 RÉGLAGES POUR SUSY 3

```
$susy: (); // donne les valeurs par défaut :  
$susy: (  
    'columns': susy-repeat(4),  
    'gutters': 0.25,  
    'spread': 'narrow',  
    'container-spread': 'narrow',  
);
```

# ' GUTTER '

Le paramètre 'gutter' n'a pas changé  
Il est semblable à la propriété `grid-column-gap`  
Si la valeur ne précise pas d'unité de mesure,  
elle agit comme une fraction

```
// gouttières fluides  
$gutters: 0.25;  
// gouttières statiques  
$gutters: 10px;
```

# BOX-SIZING

Définir le modèle de boîte  
avant de débiter les mises en page

```
* { box-sizing: border-box; }
```

# SUSY SANS SASS?

Si les gouttières sont remplacées par du padding

Pas besoin de Susy...

De simples variables CSS feront l'affaire

[codepen/mirisuzanne/](https://codepen.io/mirisuzanne/)

# IL N'Y A PLUS DE MIXINS DANS SUSY...

```
@include container();  
@include span();
```

# PLUSIEURS STRATÉGIES POSSIBLES

Sans mixins, c'est à vous de choisir une stratégie:

- des flottants
- des flexbox
- ou des tableaux css  
(`display:table`)

# AVEC DES FLOTTANTS

```
.contenu {  
    @include span(4) ;  
    width: span(4 at 2);  
    float:left;  
    margin-left: span(1);  
}
```

# AVEC DES FLEXBOX

Remplacer le flex-basis par la fonction span()

```
.flexConteneurBloc__item {  
    flex: 1 1 span(3 of 12);  
}
```



# LES FONCTIONS SPAN() ET GUTTER()

peuvent être utilisées n'importe où:

```
.contenu {  
  width: span(3);  
  float:left;  
  margin-right: gutter();  
}  
.flexConteneurBloc__item {  
  flex: 1 1 span(3);  
  padding: 0 gutter()/2;  
}  
.repousserDe3col {  
  margin-left: span(3 wide);  
}
```

# MIXIN SPAN ?

Si l'intégration devient trop répétitive,  
on peut réintroduire le mixin span()

```
@mixin span($span,$config: $susy) {  
  width: span($span, $config);  
  @if index($span, 'last') {  
    float: right;  
  } @else {  
    float: left;  
    margin-right: gutter();  
  }  
}
```

# LE CONCEPT DU SPREAD

```
$susy: (  
  'columns': susy-repeat(4),  
  'gutters': 0.25,  
  'spread': 'narrow',  
  'container-spread': 'narrow',  
);
```

# SPREAD

Comment se combine colonnes et gouttières

**narrow**

une gouttière de moins que le nombre de colonnes

**wide**

un nombre égal de gouttières et de colonnes

**wider**

une gouttière de plus que le nombre de colonnes

Susy a besoin de savoir

# COMMENT S'ÉTENDENT (SPREAD)

les éléments et leurs conteneurs.

La notion de `container` avec son `max-width` a disparu.

Chaque élément participant à la grille de mise en page est un conteneur potentiel.

Un conteneur est le contexte-parent d'un élément.

# POSITIONNEMENT DES GOUTTIÈRES

Les réglages de spread et container-spread varient selon le positionnement des gouttières exigé par la mise en page

```
$susy: (          // gutter-position: before | after (margins)
  'spread': 'narrow',
  'container-spread': 'narrow',
);

$susy: (          // gutter-position: split (margins)
  'spread': 'narrow',
  'container-spread': 'wide',
);

$susy: (          // gutter-position: inside (padding)
  'spread': 'wide',
  'container-spread': 'wide',
);
```

Il est possible de redéfinir localement ces options de position des gouttières



# EN REPOUSSANT (PUSHING)

```
.push-3 {  
  margin-left: span(3 wide);  
}
```

# EN PLACANT EN RETRAIT (PULLING)

```
.pull-3 {  
  margin-left: 0 - span(3 wide);  
}
```

# EN AJOUTANT DU PADDING

```
.pad-left-3 {  
  padding-left: span(3 wide);  
}
```

# EN FAISANT DÉBORDER (BLEEDING)

```
.bleed-left-3 {  
  margin-left: 0 - span(3 wide);  
  padding-left: span(3 wide);  
}
```

# SYNTAXE RACCOURCIE DES FONCTIONS

## PREMIÈRE PARTIE :LE SPAN

```
// <width> <location> <spread>  
$span: 2;  
$spread: 3 wide;
```

```
// <location> est requis seulement pour les grilles asymétriques  
$location: 3 at 2 narrow;
```

# SYNTAXE RACCOURCIE DES FONCTIONS

## DEUXIÈME PARTIE :LE CONTEXTE

```
// of <columns> <container-spread> <gutters>  
$columns: of susy-repeat(6);  
$spread: of (120px 1 1 14em) wide;
```

Le paramètre <columns> de la syntaxe raccourcie  
est semblable au réglage

MAIS si sa valeur est un **nombre sans unité**,  
il est traité comme une **tranche** du réglage

```
// RÉGLAGE columns: susy-repeat(12, 120px)  
$shorthand: of 4;  
$meaning: of susy-repeat(4, 120px);
```

# LA FONCTION SPAN()

Requiert seulement une valeur de largeur  
mais accepte tous les paramètres de la syntaxe  
raccourcie

```
// Usage habituel  
$width: span(3);  
// Usage de complexité maximale  
$width: span(first 3 wide of (1 1 2 3 5 8) wide set-gutters 20px)
```



# LA FONCTION GUTTER()

- retourne la largeur d'une seule gouttière
- n'accepte que la seconde partie de la syntaxe raccourcie, soit, le contexte

```
// Usage habituel
$padding: gutter();
// Avec le contexte
$padding: gutter(of 4);
$same-meaning: gutter(4);
```

# PLUGIN SVG-GRID

Pour générer le visuel de la grille de mise en page:  
la fonction `susy-svg-grid( )`  
retourne un svg en ligne  
utilisé comme image d'arrière-plan

# À utiliser dans la classe du conteneur principal

```
.conteneurPrincipale,  
.wrap {  
  max-width : 1140px;  
  margin : 0 auto;  
  background : susy-svg-grid() no-repeat scroll;  
  @include clearfix;  
}
```

# RÉFÉRENCES

<http://oddbird.net/2017/06/28/susy3/>  
<http://oddbird.net/susy/docs/>