KNACSS v7 documentation

introduction

KNACSS, c'est un peu comme une feuille de style CSS « reset » sur-vitaminée qui permet de commencer un projet à partir de zéro tout en tenant compte de bonnes pratiques générales (accessibilité, performance, responsive webdesign).
KNACSS prend en charge les styles de base, mais également la typographie, les modèles de boîte, les alignements et positionnements d'éléments, les grilles de mise en page, dans l'esprit d'être adapté à toutes les tailles d'écran (responsive). Le tout automatiquement, en l'appliquant simplement sur votre structure HTML.

KNACSS est distribué librement sur Github sous licence WTFPL, toutes contributions sont bienvenues.

nouvelle version v7

La version 7 de KNACSS est sorti le **11 décembre 2017**. Les modifications sont consignées au fur et à mesure dans un fichier *"changelog"* .
Parmi les nouveautés les plus significatives :

- **Grillade**, le système de grille principal de KNACSS a été entièrement repensé pour les navigateurs modernes. Il est bâti sur Grid Layout et non plus Flexbox. Il n'est pas prévu d'alternatives pour les anciens navigateurs, mais rien ne vous empêche de continuer à utiliser l'ancienne version de Grillade ou un autre framework de grilles.
- **L'architecture** de KNACSS a été refondue pour être plus simple et plus intuitive. Les variables également.
- Le fichier de "reset" Normalize.css a été remplacé par celui de Bootstrap : **Reboot.css**.
- La plupart des **composants** du quotidien disposent à présent de styles de base : alertes, boutons, badges, checkbox, radio, tabs, flèches, il existe même un style prévu pour le bouton "burger" de navigation.

Découvrez ici-même les nouveautés de KNACSS v7 sous forme d'une documentation-styleguide.
Note : les textes de remplissage en alsacien sont issus du générateur de templates Schnapsum .

compatibilité navigateurs

KNACSS version 7 est - dans sa grande majorité - entièrement compatible avec l'ensemble des navigateurs desktop et mobiles à partir d'IE10 inclus.
Seule exception notable, le système de grilles de KNACSS est conçu à partir de la spécification CSS Grid Layout, reconnue correctement chez Microsoft

qu'à partir de sa version Edge 16 (sorti en fin 2017). Pour plus de détails, reportez-vous à la section Grilles.

installation

Il y a deux moyens d'installer KNACSS : la version "prêt à consommer" et la version "à compiler". Seule la deuxième est configurable.

version "prêt à consommer"

Si vous êtes débutant ou pressé, cette méthode est la plus simple à installer puisqu'il… n'y a rien à installer. Elle implique cependant que vous utilisiez KNACSS tel quel, sans pouvoir le configurer à votre goût et particularités.

KNACSS n'est constitué que d'un seul fichier CSS :

- en version CSS classique et lisible
- ou en version minifiée

Si vous ne comptez pas décortiquer le fichier CSS, il est préférable d'opter pour la version minifiée, plus légère (9ko gzippé). Il vous suffit ensuite de l'insérer dans votre page HTML, avant votre propre feuille de style, bien entendu.

Pour information, Autoprefixer est activé sur cette version de KNACSS et la liste des navigateurs supportés (Browserslist) est

```
[
      '> 1%',
      'last 2 versions',
      'IE >= 10', 'Edge >= 16',
      'Chrome >= 60',
      'Firefox >= 50', 'Firefox ESR',
      'Safari >= 10',
      'ios_saf >= 10',
      'Android >= 5'];
```

version "à compiler"

Pour tirer le meilleur de KNACSS, il est préférable d'opter pour la version Sass (Scss) à compiler. Elle offre la possibilité d'adapter l'outil à son projet grâce à l'apport de variables, de fonctions et d'imbrications. Vous pouvez également inclure ou non les fichiers partiels qui vous intéressent (tableaux, formulaires, grilles, composants, classes utilitaires, etc.).

Vous pouvez télécharger KNACSS complet, avec l'ensemble des fichiers et architecture, mais il est généralement plus simple de recourir à NPM ou Yarn

pour l'installation :

```
npm install knacss
```

Le fichier source principal est `/sass/knacss.scss`, c'est lui qui se charge de l'import des variables, mixins et tous les fichiers de styles du projet.
C'est ce fichier qu'il faut compiler.
Nous recommandons de compiler à l'aide de Gulp (les fichiers `gulpfile.js` et `package.json` sont fournis), mais rien ne vous empêche d'utiliser d'autres workflows tels que Grunt, CSSnext, prepros, koala, etc.

> Les fichiers de configuration *_config/_variables.scss* et *_config/_mixins.scss* devront être copiés dans votre propre dossier de travail (et modifiés avec vos propres valeurs) afin de ne pas être écrasés lors de chaque mise à jour de KNACSS via NPM.
> Au sein de l'agence Alsacréations, nous avons conçu un environnement de Workflow nommé Bretzel adapté à KNACSS, Gulp et Sass. N'hésitez pas à y piocher des idées pour vos propres projets !

Sachez qu'un **Pense-bête en PDF** est disponible pour vous souvenir des classes utiles de KNACSS.

Préfixes navigateurs

Certaines fonctionnalités avancées de KNACSS nécessitent d'employer toute une panoplie de préfixes CSS (`-webkit-`, `-moz-`, `-ms-`,…) pour être certain que les propriétés CSS fonctionneront partout.
Dans la version Sass de KNACSS, **les préfixes n'apparaissent pas** pour ne pas polluer la lecture du fichier de travail. **Il vous sera donc nécessaire de les ajouter**, de préférence automatiquement grâce à un plugin ou à l'excellent outil qu'est autoprefixer.

Projet et compilation

KNACSS se contente de gérer les styles CSS de vos pages, son compilateur Gulp intégré ne se charge que de transformer du SCSS en CSS.
Au sein d'un véritable projet, vous devrez réaliser bien plus de tâches : compilations, optimisation d'images, concaténations et minifications de fichiers, etc.
Sachez que chez nous, à l'agence Alsacréations nous nous servons d'un outil de workflow Gulp libre, gratuit, que nous avons conçu et qui se nomme **Bretzel** .

conventions

Voici quelques conventions de nommage et bonnes pratiques de KNACSS, cette liste est non exhaustive :

- **Priorité aux classes :** Privilégiez au maximum l'usage de classes plutôt que des sélecteurs basés sur les noms des balises ou leur `id`
- **Nommage des classes :** Choisissez des noms de composants fonctionnels réutilisables (ex. `alert`), des noms de sous-éléments préfixés par leur parent (ex. `alert-title`) et des variantes facilement distinguables (ex. `alert-title--alternate`)
- **Pas de mélange :** Séparez la structure de l'apparence (une règle CSS ne doit pas comporter à la fois `padding` et `background` par exemple). Prévoyez des styles de base réutilisables, puis des classes de variantes graphiques
- **Autonomie des composants :** Séparez le conteneur du contenu (un composant ne doit jamais être ciblé par un sélecteur qui tient compte de son parent) Par exemple, n'écrivez pas `.sidebar .button` mais `.button-primary`
- **Variables :** les variables de KNACSS sont rédigées en minuscule, en anglais et les mots composés sont séparés d'un trait d'union. De préférence, le nom du composant apparaît en premier dans le nom d'une variable (ex. `$checkbox-size` plutôt que `$size-checkbox`), à l'exception des couleurs globales de texte ou de fond (`$color-primary` , `$background-base` , etc.)
- **Couleurs :** Employez systématiquement une variable pour désigner vos couleurs au sein des projets.
- **Points de rupture :** optez pour la méthodologie "Mobile First" et appliquez de préférence des media queries de ce type : `@media (min-width: $breakpoint)` . Si vous deviez choisir un intervalle maximum, optez pour `@media (max-width: ($breakpoint - 1))` pour éviter les chevauchements
- **Classes utilitaires :** KNACSS propose quelques classes utilitaires telles que `.mt0` , `.txtcenter` , `.fl` , etc. mais il est préférable de ne pas en abuser. Évitez d'accumuler les classes sur un même élément
- **Réutilisez :** Utilisez au maximum les modèles et composants réutilisables tels que les objet "media" et "autogrid." (voir fichiers dans `/sass/components/`)

couleurs

Ci-dessous, la palette de couleurs employée de base sur KNACSS, ainsi que les noms de variables Scss associées.

color names

#fff	#f8f9fa	#e7e9ed	#dee2e6	#ced4da	#acb3c2	#727e96
$white	**$gray-100**	**$gray-200**	**$gray-300**	**$gray-400**	**$gray-500**	**$gray-600**

#454d5d	#333	#212529	#000
$gray-700	**$gray-800**	**$gray-900**	**$black**

#5BC0DE	#0275D8	#5CB85C	#F0AD4E	#D9534F
$blue-300	**$blue-500**	**$green-500**	**$orange-500**	**$red-500**

semantic colors

$green-500	$blue-500	$green-500	$blue-300	$orange-500	$red-500	$gray-800
$color-brand	**$color-primary**	**$color-success**	**$color-info**	**$color-warning**	**$color-danger**	**$color-inverse**

transparent	$gray-200
$color-ghost	**$color-muted**

$gray-900	$white	$gray-800	darken($link-color, 15%)	$gray-800
$color-base	**$background-base**	**$link-color**	**$link-color-hover**	**$forms-color**

fontes

Trois variables sont prévues pour les familles de police. La première, `$font-stack-common` , est appliquée par défaut sur le contenu. Il s'agit de ce que l'on appelle la System Font Stack, déjà adoptée par des sites tels que Medium, Booking, WordPress, Alsacréations, etc.
Les voici en oeuvre :

> Ici un texte de contenu en $font-stack-common. Lorem Elsass Ipsum mitt picon bière munster du ftomi! Ponchour bisame. Bibbeleskaas jetz here's some code rossbolla sech choucroute un schwanz geburtstàg

```
$font-stack-common: -apple-system,BlinkMacSystemFont,"Segoe UI",Roboto,Oxygen-
Sans,Ubuntu,Cantarell,"Helvetica Neue",sans-serif;
```

> Ici un texte de contenu en $font-stack-headings. Lorem Elsass Ipsum mitt picon bière munster du ftomi! Ponchour bisame. Bibbeleskaas jetz here's some code rossbolla sech choucroute un schwanz geburtstàg

```
$font-stack-headings: sans-serif;
```

> Ici un texte de contenu en $font-stack-monospace. Lorem Elsass Ipsum mitt picon bière munster du ftomi! Ponchour bisame. Bibbeleskaas jetz here's some code rossbolla sech choucroute un schwanz geburtstàg

```
$font-stack-monospace: consolas, courier, monospace;
```

titres

Les six niveaux de titre sont stylés avec leurs éléments respectifs : `h1` à `h6`, ou (recommandé) les classes `.h1-like` à `.h6-like`.

titre niveau 1

titre niveau 2

titre niveau 3

titre niveau 4

titre niveau 5

titre niveau 6

paragraphe stylé comme un titre 1

```
<h1>titre niveau 1</h1>
<h2>titre niveau 2</h2>
<h3>titre niveau 3</h3>
<h4>titre niveau 4</h4>
<h5>titre niveau 5</h5>
```

```
<h6>titre niveau 6</h6>
<p class="h1-like">paragraphe stylé comme un titre 1</p>
```

paragraphes et textes

Lorem Elsass Ipsum mitt picon bière munster du ftomi! Je suis un lien Ponchour bisame. Bibbeleskaas je suis marqué jetz rossbolla sech *je suis un * choucroute un schwanz geburtstàg, Je suis une touche `du clavier` Chinette dû, ìch bier deppfele schiesser. Flammekueche de knèkes Seppele gal! a hopla geburtstàg, **je suis un ** alles fraü Chulia Roberts `je suis du code` oder knäckes dûû blottkopf.

```
<p>Lorem Elsass Ipsum mitt picon bière munster du ftomi! <a href="#">Je suis un lien</a> Ponchour bisame.
Bibbeleskaas <mark>je suis marqué</mark> jetz rossbolla sech <em>je suis un <em></em> choucroute un schwanz
geburtstàg, Je suis une touche <kbd>du clavier</kbd> Chinette dû, ìch bier deppfele schiesser. Flammekueche
de knèkes Seppele gal! a hopla geburtstàg, <strong>je suis un <strong></strong> alles fraü Chulia Roberts
<code>je suis du code</code> oder knäckes dûû blottkopf.</p>
```

listes

- Liste non ordonnée
- Salade
- Tomate
- Oignon
- Choucroute

1. Liste ordonnée
2. Salade
3. Tomate
4. Oignon
5. Choucroute

Liste `.unstyled` ou `.is-unstyled`

Salade

Tomate

Oignon

Choucroute

```
<ul>
    <li>Liste non ordonnée</li>
    <li>Salade</li>
    <li>Tomate</li>
    <li>Oignon</li>
    <li>Choucroute</li>
</ul>

<ol>
    <li>Liste ordonnée</li>
    <li>Salade</li>
    <li>Tomate</li>
    <li>Oignon</li>
    <li>Choucroute</li>
</ol>

<ul class="unstyled">
    <li>Liste <code>.unstyled</code></li>
    <li>Salade</li>
    <li>Tomate</li>
    <li>Oignon</li>
    <li>Choucroute</li>
</ul>
```

tableaux

Les tableaux de données disposent de styles de base. Ceux munis d'une classe `.table` bénéficient en sus de propriétés graphiques définies selon des

variables de projet dédiées : couleur de fond, des titres, etc. La classe additionnelle `.table--zebra` colorie une rangée sur deux.

Fruit	Note	Prix
Kiwi	★★★	13.37€
Banane	★★	0.42€

.table

Voici la structure HTML permettant d'obtenir ce résultat :

```
<table class="table" summary="">
  <caption>.table</caption>
  <thead>
    <tr>
      <th scope="col">Fruit</th>
      <th scope="col">Note</th>
      <th scope="col">Prix</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Kiwi</td>
      <td>★★★</td>
      <td>13.37€</td>
    </tr>
    <tr>
      <td>Banane</td>
      <td>★★</td>
      <td>0.42€</td>
    </tr>
  </tbody>
</table>
```

citations

> Lorem Elsass Ipsum mitt picon bière munster du ftomi! Ponchour bisame. Bibbeleskaas jetz here's some code rossbolla sech choucroute un schwanz geburtstàg

```
<blockquote>
   <p>Lorem Elsass Ipsum mitt picon bière munster du ftomi! Ponchour bisame. Bibbeleskaas jetz here's some code rossbolla sech choucroute un schwanz geburtstàg</p>
</blockquote>
```

alertes

Une alerte est une boîte d'information disposant de la classe `.alert` (pour les styles par défaut) ou une classe commençant par `.alert--` pour les différentes variantes `.alert--primary`, `.alert--success`, `.alert--warning`, `.alert--danger`, `.alert--info`, `.alert--inverse`, `.alert--ghost`.
Les variantes sont gérées par un ensemble (thème) de variables comprenant : couleur de fond, couleur de texte et bordure optionnelle.

Alert Default

Primary

Success

Warning

Danger

Info

Inverse

Ghost

```
<div class="alert">Alert Default</div>
<div class="alert--primary">Primary</div>
<div class="alert--success">Success</div>
<div class="alert--warning">Warning</div>
<div class="alert--danger">Danger</div>
<div class="alert--info">Info</div>
<div class="alert--inverse">Inverse</div>
<div class="alert--ghost">Ghost</div>
```

inputs

text search number email ●●●●●●●●

```
<input type="text" placeholder="text">
<input type="search" placeholder="search">
<input type="number" placeholder="number">
<input type="email" placeholder="email">
<input type="password" value="password">
<progress></progress>
```

checkboxes

KNACSS prend en compte les styles de base des cases à cocher, il suffit d'appliquer la classe `.checkbox` sur l'élément `input` afin de le voir en action.

Voici le code HTML recommandé : `<input type="checkbox" class="checkbox" id="c1"><label for="c1">click here</label>`

- Salade
- Tomate
- Oignon

```
<form action="#">
  <ul class="is-unstyled">
    <li>
      <input type="checkbox" class="checkbox" id="c1">
      <label for="c1">Salade</label>
    </li>
    <li>
      <input type="checkbox" class="checkbox" id="c2" checked="checked">
      <label for="c2">Tomate</label>
    </li>
    <li>
      <input type="checkbox" class="checkbox" id="c3" checked="checked" disabled="disabled">
      <label for="c3">Oignon</label>
    </li>
    <li>
      <input type="checkbox" class="checkbox" id="c4" disabled="disabled">
      <label for="c4">Choucroute</label>
    </li>
  </ul>
</form>
```

radios

KNACSS prend en compte les styles de base des boutons radio, il suffit d'appliquer la classe `.radio` sur l'élément `input` afin de le voir en action.

Voici le code HTML recommandé : `<input type="radio" class="radio" name="radio" id="r1"><label for="r1">Click here</label>`

- Salade
- Tomate
- Oignon

```
<form action="#">
  <ul class="is-unstyled">
    <li>
      <input type="radio" class="radio" name="radio" id="r1">
      <label for="r1">Salade</label>
    </li>
    <li>
      <input type="radio" class="radio" name="radio" id="r2" checked="checked">
      <label for="r2">Tomate</label>
    </li>
    <li>
      <input type="radio" class="radio" name="radio" id="r3" checked="checked" disabled="disabled">
      <label for="r3">Oignon</label>
    </li>
    <li>
      <input type="radio" class="radio" name="radio" id="r4" disabled="disabled">
      <label for="r4">Choucroute</label>
    </li>
  </ul>
</form>
```

switchs

Les boutons de switch (ou toggle) sont gérés à l'aide de la classe `.switch` sur l'élément `input checkbox`.

Voici le code HTML recommandé : `<input type="checkbox" class="switch" id="switch">`
`<label for="switch" class="label">slide to unlock</label>`

× slide to unlock

```
<input type="checkbox" class="switch" id="switch">
<label for="switch" class="label">slide to unlock</label>
```

selects

Les éléments `<select>` sont automatiquement stylés dans KNACSS. Vous pouvez adapter les styles de base en modifiant les variables du projet.

Salade

```
<select>
  <option value="valeur1">Salade</option>
  <option value="valeur2">Tomate</option>
  <option value="valeur3">Oignon</option>
</select>
```

flèches / arrows

Quatre classes sont prévues pour afficher des flèches décoratives à droite de n'importe quel élément. Il s'agit de `icon-arrow--down` , `icon-arrow--up` , `icon-arrow--right` et `icon-arrow--left` . Si vous souhaitez que la flèche apparaisse à gauche, appliquez la classe `.flex-row-reverse` sur le parent.

Attention, les flèches "arrows" se basent sur les masques CSS, actuellement incompatibles avec IE / Edge.

arrow down

arrow up

arrow right

arrow left

arrow left

```
<p>icon-arrow down <i class="icon-arrow--down"></i></p>
<p>icon-arrow up <i class="icon-arrow--up"></i></p>
<p>icon-arrow right <i class="icon-arrow--right"></i></p>
<p>icon-arrow left <i class="icon-arrow--left"></i></p>
<p class="flex-row-reverse">icon-arrow left <i class="icon-arrow--left"></i></p>
```

boutons

Les boutons, comme les alertes disposent d'un ensemble de thème de couleurs (couleur de fond, couleur de texte et bordure optionnelle).
Il est recommandé d'utiliser l'élément HTML `<button>` , mais rien n'empêche d'opter pour d'autres types d'éléments, tant qu'ils sont associés à l'attribut ARIA `role="button"`
Un bouton dispose de la classe `.btn` ou `.button` (pour les styles par défaut) ou une classe commençant par `.btn--` ou `.button--` pour les différentes variantes, par exemple `.btn--primary` , `.btn--success` , `.btn--warning` , `.btn--danger` , `.btn--info` , `.btn--inverse` , `.btn--ghost` .

Button Default Primary Success Warning Danger Info Inverse

Ghost

```
<button class="btn">Button Default</button>
<button class="btn--primary">Primary</button>
<button class="btn--success">Success</button>
<input  class="btn--warning" type="button" role="button" value="Warning">
<span class="btn--danger" role="button">Danger</span>
<span class="btn--info" role="button">Info</span>
<span class="btn--inverse" role="button">Inverse</span>
```

```
<button class="btn--ghost">Ghost</button>
```

burger button

Un élément possédant la classe `.nav-button` et contenant un élément vide devient un bouton de navigation stylé et prêt à l'action (avec un peu de renfort de JavaScript ou CSS pour déclencher l'événement bien sûr).
Pour des raisons d'accessibilité, il est fortement recommandé d'utiliser la structure HTML ci-dessous pour votre bouton.

```
<button class="nav-button" type="button" role="button" aria-label="open/close navigation"><i></i></button>
```

Voici le bout de code JavaScript employé pour activer ce bouton

```
<script>
/**!
Navigation Button Toggle class
*/
(function() {

// old browser or not ?
if ( !('querySelector' in document && 'addEventListener' in window) ) {
return;
}
window.document.documentElement.className += ' js-enabled';

function toggleNav() {

// Define targets by their class or id
var button = document.querySelector('.nav-button');
var target = document.querySelector('body > nav');
```

```
// click-touch event
if ( button ) {
  button.addEventListener('click',
  function (e) {
      button.classList.toggle('is-active');
    target.classList.toggle('is-opened');
    e.preventDefault();
  }, false );
}
} // end toggleNav()

toggleNav();
}());
</script>
```

tags (étiquette)

Un tag dispose de la classe `.tag` (pour les styles par défaut) ou une classe commençant par `.tag--` pour les différentes variantes, par exemple `.tag--primary`, `.tag--success`, `.tag--warning`, `.tag--danger`, `.tag--info`, `.tag--inverse`, `.tag--ghost`.

Tag Default Primary Success Warning Danger Info Inverse

```
<button class="tag">Badge Default</button>
<button class="tag--primary">Primary</button>
<button class="tag--success">Success</button>
<span class="tag--warning">Warning</span>
<span class="tag--danger">Danger</span>
<span class="tag--info">Info</span>
<span class="tag--inverse">Inverse</span>
```

badges (arrondis)

Un badge est systématiquement rond et s'adapte à son contenu. Il dispose de la classe `.badge` (pour les styles par défaut) ou une classe commençant par `.badge--` pour les différentes variantes, par exemple `.badge--primary`, `.badge--success`, `.badge--warning`, `.badge--danger`, `.badge--info`, `.badge--inverse`, `.badge--ghost`.

Badge Default Primary Success Warning Danger Info Inverse -20%

```
<button class="badge">Badge Default</button>
<button class="badge--primary">Primary</button>
<button class="badge--success">Success</button>
<span class="badge--warning">Warning</span>
<span class="badge--danger">Danger</span>
<span class="badge--info">Info</span>
<span class="badge--inverse">Inverse</span>
```

tabs / onglets

KNACSS prévoit des styles pour une navigation à base d'onglets accessibles (fonctionnels aux flèches du clavier et à la souris). Le principe général est de placer la classe `.js-tabs` sur le conteneur global.
Les instructions complètes, le code HTML et JavaScript se trouvent sur le projet d'Alsacréations *Pepin*.
Attention : pour fonctionner, les onglets nécessitent un code JavaScript (à récupérer sur Pepin).

Salade Tomate Oignons

Contenu 1.

Lorem Elsass Ipsum mitt picon bière munster du ftomi! Ponchour bisame. Bibbeleskaas jetz rossbolla sech choucroute un schwanz geburtstàg, Chinette dû, ìch bier deppfele schiesser.

```
<div class="tabs js-tabs">
  <nav class="tabs-menu">
    <a href="#tab1" class="tabs-menu-link is-active">Salade</a>
    <a href="#tab2" class="tabs-menu-link">Tomate</a>
    <a href="#tab3" class="tabs-menu-link">Oignons</a>
  </nav>

  <div class="tabs-content">
    <div id="tab1" class="tabs-content-item">Contenu 1. <br>Lorem Elsass Ipsum mitt picon bière munster du
ftomi! Ponchour bisame. Bibbeleskaas jetz rossbolla sech choucroute un schwanz geburtstàg, Chinette dû, ìch
bier deppfele schiesser.</div>
    <div id="tab2" class="tabs-content-item">Contenu 2. <br>Flammekueche de knèkes Seppele gal! a hopla
geburtstàg, alles fraü Chulia Roberts oder knäckes dûû blottkopf. Noch bredele schissabibala, yeuh e
schmutz.</div>
    <div id="tab3" class="tabs-content-item">Contenu 3. <br>E gewurtztraminer doch Carola schneck, schmutz
a riesling de chambon eme rucksack Roger dû hopla geiss, jetz Chorchette de Scharrarbergheim.</div>
  </div>
</div>
```

alignements

span link

```
<div class="txtleft">
    <span>span</span>
    <a href="#">link</a>
</div>
```

span link

```
<div class="txtcenter">
    <span>span</span>
    <a href="#">link</a>
</div>
```

span link

```
<div class="txtright">
    <span>span</span>
    <a href="#">link</a>
</div>
```

div.left

```
<div style="width: 30%" class="left">div.left</div>
```

div.center

```
<div style="width: 30%" class="center">div.center</div>
```

div.right

```
<div style="width: 30%" class="right">div.right</div>
```

flottants

div.fl div.fr

```
<div class="clearfix">
        <div style="width: 20%" class="fl">div.fl</div>
        <div style="width: 20%" class="fr">div.fr</div>
    </div>
```

flexbox

div span ...

```
<section class="flex-container">
  <div>div</div>
  <span>span</span>
  <em>…</em>
</section>
```

div

span

...

```
<div class="flex-container--column">
<div>div</div>
<span>span</span>
<em>…</em>
</div>
```

.w33 .w150p .item-fluid

```
<div class="flex-container">
  <div class="w33">.w33</div>
  <span class="w150p">.w150p</span>
  <span class="item-fluid">.item-fluid</span>
```

```
</div>
```

.item-first item item

```
<div class="flex-container">
  <div>item</div>
  <span>item</span>
  <span class="item-first">.item-first</span>
</div>
```

item item .item-last

```
<div class="flex-container">
  <div class="item-last">.item-last</div>
  <span>item</span>
  <span>item</span>
</div>
```

.item-center

```
<div class="flex-container" style="height: 200px;">
  <span class="item-center">.item-center</span>
</div>
```

Grillade

Grillade, le système de grille principal de KNACSS a été entièrement repensé pour les navigateurs modernes. Il est dorénavant bâti sur Grid Layout et non plus Flexbox. Grid Layout n'est pleinement compatible qu'à partir de Edge 16, même si Autoprefixer est activé et *devrait* rendre la grille compatible sur IE10.

Par défaut, le nouveau système de grilles est automatiquement importé au sein de KNACSS, **cependant les deux versions (nouvelle et ancienne en Flexbox) co-existent toujours pour des raisons de compatibilité :**

- `grillade-grid.scss` (nouvelle version en Grid Layout)
- `grillade-flex.scss` (ancienne version en Flexbox)

Ainsi, il est possible, en Sass, de remplacer le fichier importé "grillade-grid.scss" par l'ancienne version "grillade-flex.scss" pour assurer un support jusqu'à IE10 minimum. Les fichiers CSS compilés peuvent également être utilisés de manière autonome :

- `grillade-grid.css` (3ko)
- `grillade-flex.css` (11ko)

Grillade n'est pas un "framework", il s'agit simplement de fonctionnalités basiques pour réaliser des grilles simples et courantes (et responsive).
Si vous avez besoin de grilles complexes :

- Utilisez la spécification CSS Grid Layout (c'est fait pour ça)
- Utilisez la grille de Bootstrap (bonne chance)

Les grilles ne s'activent qu'à partir d'un breakpoint minimum (`$tiny` = 480px par défaut). En deça, les éléments s'affichent en blocs, les uns sous les autres.

	Grillade v6	**Grillade v7**
Techno CSS employée	Flexbox	Grid Layout
Poids fichier CSS	11 Ko	3 Ko
Compatibilité	IE10+	Edge 16+ sûr IE10 à tester
Fonctionnalités	Mobile first Largeurs fluides Gouttières Fusion de colonnes ~~Fusion de rangées~~ Gestion des écrans intermédiaires ~~Tailles d'enfants responsive~~ Ordonnancement des enfants	Mobile first Largeurs fluides Gouttières Fusion de colonnes Fusion de rangées Gestion des écrans intermédiaires Tailles d'enfants responsive Ordonnancement des enfants

comparaison entre Grillade v6 et Grillade v7

Voici quelques exemples de grilles réalisées avec KNACSS v7…

.autogrid (ou .grid)

Une "autogrid" (classe `.autogrid` ou `.grid`) est une micro-grille où les enfants demeurent toujours sur une seule ligne et se répartissent de façon équitable quel que soit leur nombre. Il est possible d'appliquer une gouttière avec les classes `.has-gutter` , `.has-gutter-l` , `.has-gutter-xl` .

1 2 3 4 5

```
<section class="autogrid">
    <div>1</div>
    <div>2</div>
    <div>3</div>
    <div>4</div>
    <div>5</div>
</section>
```

.grid-6

Une grille de 6 colonnes, avec retour à la ligne possible, et sans gouttière

1 2 3 4 5 6

7 8

```
<section class="grid-6">
    <div>1</div>
    <div>2</div>
    <div>3</div>
    <div>4</div>
    <div>5</div>
    <div>6</div>
    <div>7</div>
    <div>8</div>
</section>
```

.grid-6 .has-gutter

Une grille de 6 colonnes avec une gouttière. Les classes prévues sont `.has-gutter`, `.has-gutter-l`, `.has-gutter-xl`

1 2 3 4 5 6

7 8

```
<section class="grid-6 has-gutter">
    <div>1</div>
    <div>2</div>
    <div>3</div>
    <div>4</div>
    <div>5</div>
    <div>6</div>
    <div>7</div>
    <div>8</div>
</section>
```

.grid-4 .has-gutter

Une grille de 4 colonnes avec une gouttière. Certains enfants s'étendent sur plusieurs colonnes ou rangées.

1 .col-2 .row-2

4 5 6

```
<section class="grid-4 has-gutter">
    <div>1</div>
    <div class="col-2">.col-2</div>
    <div class="row-2">.row-2</div>
    <div>4</div>
    <div>5</div>
    <div>6</div>
</section>
```

.grid-6-small-3

Une grille de 6 colonnes sur grand écran, et 3 colonnes sur écran moyen.

1 2 3 4 5 6

7 8

```
<section class="grid-6-small-3">
<div>1</div>
<div>2</div>
<div>3</div>
<div>4</div>
<div>5</div>
<div>6</div>
<div>7</div>
<div>8</div>
</section>
```

.grid-7-small-4

Une grille de 7 colonnes sur grand écran, et 4 colonnes sur écran moyen. Quelques enfants changent de taille sur écran moyen

1 2 .col-2-small-1 4 5 .col-1-small-2

7 .col-2-small-all

```
<section class="grid-7-small-4">
    <div>1</div>
    <div>2</div>
    <div class="col-2-small-1">.col-2-small-1</div>
    <div>4</div>
```

```
    <div>5</div>
    <div class="col-1-small-2">.col-1-small-2</div>
    <div>7</div>
    <div class="col-2-small-all">.col-2-small-all</div>
</section>
```

.grid-6-small-3 .has-gutter

Panachage des possibilités de grilles

(3) item-first	1	2	4	5	6
7	col-3-small-2			col-2-small-1	
col-2-small-1		col-4-small-2			
row-2	13	14	15	16	18
	col-3			19	
col-all					

```
<section class="grid-6-small-3 has-gutter">
  <div>1</div>
  <div>2</div>
  <div class="item-first">(3) item-first</div>
  <div>4</div>
  <div>5</div>
  <div>6</div>
  <div>7</div>
```

```
  <div class="col-3-small-2">col-3-small-2</div>
  <div class="col-2-small-1">col-2-small-1</div>
  <div class="col-2-small-1">col-2-small-1</div>
  <div class="col-4-small-2">col-4-small-2</div>
  <div class="row-2">row-2</div>
  <div>13</div>
  <div>14</div>
  <div>15</div>
  <div>16</div>
  <div class="col-3">col-3</div>
  <div>18</div>
  <div>19</div>
  <div class="col-all">col-all</div>
</section>
```

mixins

Le fichier regroupant les mixins et fonctions Sass se nomme `sass/_config/_mixins.scss` . Il contient des raccourcis Sass qui facilitent la vie de l'intégrateur.

mixin "respond-to()"

Ce mixin permet de faciliter et d'harmoniser les points de ruptures. Écrivez ceci dans votre fichier Sass :

```
p {
  color: blue;
  @include respond-to("small-up") {
    color: hotpink;
  }
}
```

Le résultat après compilation sera :

```
p {
```

```
  color: blue;
}
@media (min-width: 576px) {
  p {
    color: hotpink;
  }
}
```

Les paramètres et les intervales correspondants sont :

```
$bp-aliases: (
  'tiny'   : (max-width: #{$tiny - 1}),
  'small'  : (max-width: #{$small - 1}),
  'medium' : (max-width: #{$medium - 1}),
  'large' : (max-width: #{$large - 1}),
  'extra-large'  : (max-width: #{$extra-large - 1}),
  'tiny-up'   : (min-width: #{$tiny}),
  'small-up'  : (min-width: #{$small}),
  'medium-up' : (min-width: #{$medium}),
  'large-up' : (min-width: #{$large}),
  'extra-large-up' : (min-width: #{$extra-large}),
  'retina' : (min-resolution: 2dppx)
);
```

mixin "grid()"

Ce mixin de trois paramètres optionnels permet de générer une grille simple et rapide. Écrivez ceci dans votre fichier Sass :

```
.ingrid {
  @include grid(4, 1rem, 640px);
}
```

Le résultat après compilation sera :

```
@media (min-width: 640px) {
```

```
  .ingrid {
    display: grid;
    grid-template-columns: repeat(4, 1fr);
    grid-gap: 1rem;
  }
}
```

mixin "font-size()"

Ce mixin permet de gérer les tailles d'éléments autant mobiles que desktop. Écrivez ceci dans votre fichier Sass :

```
h2 {
  @include font-size(h2);
}
```

Le résultat après compilation sera :

```
h2 {
  font-size: 2.4rem;
}
@media (min-width: 576px) {
  h2 {
    font-size: 2.8rem;
  }
}
```

Les éléments prévus pour être ciblés par ce mixin sont : base (police générale), h1, h2, h3, h4, h5 et h6. Les valeurs de polices de tous ces éléments sont indiquées dans le fichier de variables du projet.

variables

Le fichier permettant de configurer l'ensemble des variables du projet se nomme `sass/_config/_variables.scss` . Il est fortement conseillé de modifier ce fichier avec vos propres valeurs, plutôt que de les écraser par la suite.

```
// Config file and project variables

// ----------------
// Breakpoints zone
// ----------------

// Warning: you should use your own values, regardless of the devices
// Best practise is Mobile First: (min-width: $breakpoint)
$tiny        : 480px !default; // or 'em' if you prefer, of course
$small       : 576px !default;
$medium      : 768px !default;
$large       : 992px !default;
$extra-large : 1200px !default;

// ----------
// Fonts zone
// ----------

// Font families
$font-family-base      : -apple-system, BlinkMacSystemFont, "Segoe UI", Roboto, Oxygen-Sans, Ubuntu,
Cantarell, "Helvetica Neue", sans-serif !default; // system font stack
$font-family-headings  : sans-serif !default; // font for h1, h2.. h6
$font-family-monospace : consolas, courier, monospace !default; // font for code and samples

// Font sizes (1.6rem value is "16px" equivalent)
$font-size-base : 1.6rem !default;

$font-sizes: (
  base: (
    mobile  : 1.4rem,
    desktop : $font-size-base
  ),
  h1: (
    mobile  : 2.8rem,
    desktop : 3.2rem
```

```
    ),
    h2: (
      mobile  : 2.4rem,
      desktop : 2.8rem
    ),
    h3: (
      mobile  : 2.0rem,
      desktop : 2.4rem
    ),
    h4: (
      mobile  : 1.8rem,
      desktop : 2.0rem
    ),
    h5: (
      mobile  : 1.6rem,
      desktop : 1.8rem
    ),
    h6: (
      mobile  : 1.4rem,
      desktop : 1.6rem
    )
  ) !default;

  // Line heights
  $line-height-s    : 1.3 !default;
  $line-height-base : 1.5 !default;
  $line-height-l    : 1.7 !default;

  // Default margin-bottom
  $margin-bottom-base     : 1rem !default;
  $headings-margin-bottom : $margin-bottom-base /2 !default;
  $paragraph-margin-bottom: $margin-bottom-base !default;

  // Font weights
  $weight-light   : 200 !default;
  $weight-book    : 300 !default;
```

```
$weight-regular : 400 !default;
$weight-medium  : 500 !default;
$weight-bold    : 700 !default;

// ------------
// Spacing zone
// ------------

// Grid gutters (for .has-gutter-* classes)
$grid-gutters: (
  '': 1rem,
  '-l': 2rem,
  '-xl': 4rem
) !default;

// Spacings
$spacer-tiny             : .5rem  !default;
$spacer-tiny-plus        : .7rem !default;
$spacer-small            : 1rem !default;
$spacer-small-plus       : 1.5rem !default;
$spacer-medium           : 2rem !default;
$spacer-medium-plus      : 3rem !default;
$spacer-large            : 4rem !default;
$spacer-large-plus       : 6rem !default;
$spacer-extra-large      : 8rem !default;
$spacer-extra-large-plus : 12rem !default;
$spacer-ultra-large      : 16rem !default;
$spacer-ultra-large-plus : 20rem !default;

// z-indexes
$zindex-navigation   : 1000 !default;
$zindex-dropdown     : 2000 !default;
$zindex-popover      : 3000 !default;
$zindex-tooltip      : 4000 !default;
$zindex-modal        : 5000 !default;
$zindex-notification : 6000 !default;
```

```
$zindex-debug        : 7000 !default;

// ----------
// Color zone
// ----------

// Color names
$white    : #fff !default;
$gray-100 : #f8f9fa !default;
$gray-200 : #e7e9ed !default;
$gray-300 : #dee2e6 !default;
$gray-400 : #ced4da !default;
$gray-500 : #acb3c2 !default;
$gray-600 : #727e96 !default;
$gray-700 : #454d5d !default;
$gray-800 : #333 !default;
$gray-900 : #212529 !default;
$black    : #000 !default;

$blue-300   : #5BC0DE !default;
$blue-500   : #0275D8 !default;
$green-500  : #5CB85C !default;
$orange-500 : #F0AD4E !default;
$red-500    : #D9534F !default;

// Semantic colors
$color-brand    : $green-500;
$color-primary  : $blue-500;
$color-success  : $green-500;
$color-info     : $blue-300;
$color-warning  : $orange-500;
$color-danger   : $red-500;
$color-inverse  : $gray-800;
$color-ghost    : transparent;
$color-muted    : $gray-200;
```

```
$color-base      : $gray-900;
$background-base : $white;

$link-color            : $gray-800;
$link-color-hover      : darken($link-color, 15%);
$link-decoration       : none;
$link-decoration-hover : underline;

$forms-color     : $gray-800;

// ---------------
// Components zone
// ---------------

// Global border-radius
$border-radius: 0 !default;

// Component: quotes
$quote-color     : $gray-200;

// Component: arrows
$arrow-color     : $black;

// Components: checkboxes, radios, switches
$checkbox-color         : $gray-800;
$checkbox-size          : 2rem;
$checkbox-border-radius : 4px;
$switch-color           : $gray-800;
$switch-size            : 2rem;
$switch-border-radius   : 3em;

// Component: tables
$table-border               : $gray-500;
$table-caption-color        : $gray-800;
$table-background           : transparent;
$table-head-color           : $color-base;
```

```
$table-head-background      : transparent;
$table-footer-color         : $color-base;
$table-footer-background    : transparent;

// Components: buttons, badges, alerts color variants list
// Convention is: name - background-color - color - border
$variants-list: (
  (primary,    $color-primary,    $white,    none),
  (success,    $color-success,    $white,    none),
  (info,       $color-info,       $white,    none),
  (warning,    $color-warning,    $white,    none),
  (danger,     $color-danger,     $white,    none),
  (inverse,    $color-inverse,    $white,    none),
  (ghost,      $color-ghost,      $white,    0 0 0 1px $white inset)
) !default;

// Component: tabs
$tabs-border            : $gray-200;
$tabs-active-border     : $gray-800;
$tabs-color             : $color-base;
$tabs-active-color      : $gray-800;
$tabs-background        : transparent;
$tabs-active-background : transparent;
$tabs-border-radius     : 0;

// Component: nav burger button
$burger-color             : $gray-800;
$burger-background        : transparent;
$burger-hover-background  : transparent;
$burger-size              : 2.6rem;
$burger-weight            : 5px; // size of stripes
$burger-padding           : 0;
```

à propos

KNACSS (prononcez "Knèckess") est un outil construit avec amour et du CSS par Raphaël Goetter et Alsacréations.
Alsacréations est une innovante agence web alsacienne créée en 2006 qui répond à tous types de projets (développement, intégration, accessibilité, mobilité).
Un grand merci à Matthieu Bousendorfer pour le design du site web, à Stéphanie Walter pour la conception du logo, ainsi qu'à Hugo Giraudel, Philippe Vayssière, Xavier Zawala, Nicolas Hoffmann pour leurs conseils et astuces pour faire évoluer l'outil.