**Annick Lacroix**

**BOOTSTRAP**

**HTML5 et CSS3**

**Afpa 2018-2019**

# Introduction

Démarrez avec Bootstrap, le framework le plus populaire au monde pour la création de sites réactifs destinés aux mobiles, avec BootstrapCDN et une page de démarrage de modèle.

## **Démarrage rapide**

Vous souhaitez ajouter rapidement Bootstrap à votre projet? Utilisez BootstrapCDN, fourni gratuitement par les employés de StackPath. Vous utilisez un gestionnaire de paquets ou avez besoin de télécharger les fichiers source?

<https://getbootstrap.com/docs/4.2/getting-started/download/>

## **CSS**

Copiez-collez la feuille de style <link>dans votre <head>avant toutes les autres feuilles de style pour charger notre CSS.

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.2.1/css/bootstrap.min.css" integrity="sha384-GJzZqFGwb1QTTN6wy59ffF1BuGJpLSa9DkKMp0DgiMDm4iYMj70gZWKYbI706tWS" crossorigin="anonymous">

## **JS**

Beaucoup de nos composants nécessitent l'utilisation de JavaScript pour fonctionner. Plus précisément, ils requièrent [jQuery](https://jquery.com/) , [Popper.js](https://popper.js.org/) et nos propres plugins JavaScript. Placez-les <script>près de la fin de vos pages, juste avant la </body>balise de fermeture , pour les activer. jQuery doit venir en premier, puis Popper.js, puis nos plugins JavaScript.

Nous utilisons [la version slim de jQuery](https://blog.jquery.com/2016/06/09/jquery-3-0-final-released/) , mais la version complète est également prise en charge.

<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo" crossorigin="anonymous"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.6/umd/popper.min.js" integrity="sha384-wHAiFfRlMFy6i5SRaxvfOCifBUQy1xHdJ/yoi7FRNXMRBu5WHdZYu1hA6ZOblgut" crossorigin="anonymous"></script>

<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.2.1/js/bootstrap.min.js" integrity="sha384-B0UglyR+jN6CkvvICOB2joaf5I4l3gm9GU6Hc1og6Ls7i6U/mkkaduKaBhlAXv9k" crossorigin="anonymous"></script>

Vous voulez savoir quels composants requièrent explicitement jQuery, notre JS et Popper.js? Cliquez sur le lien Afficher les composants ci-dessous. Si vous n'êtes pas sûr de la structure générale de la page, continuez à lire pour un exemple de modèle de page.

Notre bootstrap.bundle.jset bootstrap.bundle.min.jsinclure [Popper](https://popper.js.org/) , mais pas [jQuery](https://jquery.com/) . Pour plus d'informations sur ce qui est inclus dans Bootstrap, veuillez consulter notre section [Contenu](https://getbootstrap.com/docs/4.2/getting-started/contents/#precompiled-bootstrap) .

### Afficher les composants nécessitant JavaScript

* Alertes pour congédiement
* Boutons pour basculer d'états et fonctionnalité de case à cocher / radio
* Carrousel pour tous les comportements, contrôles et indicateurs de diapositive
* Réduire pour basculer la visibilité du contenu
* Dropdowns pour l'affichage et le positionnement (nécessite également [Popper.js](https://popper.js.org/) )
* Modaux pour l'affichage, le positionnement et le comportement de défilement
* Navbar pour l'extension de notre plugin Collapse afin d'implémenter un comportement réactif
* Infobulles et popovers pour l'affichage et le positionnement (nécessite également [Popper.js](https://popper.js.org/) )
* Scrollspy pour le comportement de défilement et les mises à jour de navigation

## **Modèle de départ**

Assurez-vous que vos pages sont configurées avec les normes de conception et de développement les plus récentes. Cela signifie qu’il est nécessaire d’utiliser un doctype HTML5 et d’inclure une balise méta viewport pour des comportements réactifs appropriés. Rassemblez tout et vos pages devraient ressembler à ceci:

<!doctype html>

<html lang="en">

<head>

<!-- Required meta tags -->

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

<!-- Bootstrap CSS -->

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.2.1/css/bootstrap.min.css" integrity="sha384-GJzZqFGwb1QTTN6wy59ffF1BuGJpLSa9DkKMp0DgiMDm4iYMj70gZWKYbI706tWS" crossorigin="anonymous">

<title>Hello, world!</title>

</head>

<body>

<h1>Hello, world!</h1>

<!-- Optional JavaScript -->

<!-- jQuery first, then Popper.js, then Bootstrap JS -->

<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo" crossorigin="anonymous"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.6/umd/popper.min.js" integrity="sha384-wHAiFfRlMFy6i5SRaxvfOCifBUQy1xHdJ/yoi7FRNXMRBu5WHdZYu1hA6ZOblgut" crossorigin="anonymous"></script>

<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.2.1/js/bootstrap.min.js" integrity="sha384-B0UglyR+jN6CkvvICOB2joaf5I4l3gm9GU6Hc1og6Ls7i6U/mkkaduKaBhlAXv9k" crossorigin="anonymous"></script>

</body>

</html>

C'est tout ce dont vous avez besoin pour connaître l'ensemble des besoins en pages. Consultez la [documentation de présentation](https://getbootstrap.com/docs/4.2/layout/overview/) ou [nos exemples officiels](https://getbootstrap.com/docs/4.2/examples/) pour commencer à présenter le contenu et les composants de votre site.

## **Globaux importants**

Bootstrap utilise une poignée de styles et de paramètres globaux importants dont vous devrez tenir compte lors de son utilisation, qui sont tous presque exclusivement orientés vers la normalisation des styles de navigateur croisés. Plongeons dedans.

### **Doctype HTML5**

Bootstrap nécessite l'utilisation du doctype HTML5. Sans cela, vous verrez un style funky incomplet, mais l'inclure ne devrait pas causer de hoquet considérable.

<!doctype html>

<html lang="en">

...

</html>

### **Balise méta sensible**

Bootstrap est d'abord développé sur mobile , stratégie dans laquelle nous optimisons d'abord le code pour les appareils mobiles, puis mettons à l'échelle les composants à l'aide des requêtes de support CSS. Pour que le rendu et le zoom tactile soient corrects pour tous les périphériques, **ajoutez la balise méta responsive de la fenêtre d'affichage** à votre <head>.

<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

Vous pouvez voir un exemple de cela en action dans le [modèle de départ](https://getbootstrap.com/docs/4.2/getting-started/introduction/#starter-template) .

### **Taille de la boîte**

Pour un dimensionnement plus simple en CSS, nous passons de la   
box-sizingvaleur globale de content-boxà border-box. Cela garantit paddingque la largeur finale calculée d'un élément n'est pas affectée, mais cela peut entraîner des problèmes avec certains logiciels tiers tels que Google Maps et le moteur de recherche personnalisé Google.

Dans les rares cas où vous devez le remplacer, utilisez l'une des méthodes suivantes:

.selector-for-some-widget {

box-sizing: content-box;

}

Avec l'extrait ci - dessus, y compris les éléments imbriqués-contenu généré par ::beforeet ::after-Will tous héritent des spécifiés box-sizingpour que .selector-for-some-widget.

En savoir plus sur le [modèle et le dimensionnement des boîtes à l'aide de CSS Tricks](https://css-tricks.com/box-sizing/) .

### **Redémarrer**

Pour améliorer le rendu inter-navigateurs, nous utilisons [Reboot](https://getbootstrap.com/docs/4.2/content/reboot/) pour corriger les incohérences entre les navigateurs et les périphériques tout en effectuant des réinitialisations légèrement plus nuancées sur des éléments HTML courants.

### **Box-sizing**

La [box-sizing](https://css-tricks.com/almanac/properties/b/box-sizing/)propriété peut rendre la construction de mises en page CSS plus facile et beaucoup plus intuitive. C'est une aubaine pour les développeurs qu'ici, chez CSS-Tricks, nous [célébrons la Journée internationale de sensibilisation au dimensionnement des boîtes](https://css-tricks.com/international-box-sizing-awareness-day/) en février.

Mais, comment est-il si utile et bien-aimé qu’il mérite ses propres vacances sur Internet? Temps pour un peu d'histoire CSS.

### **Histoire du modèle de boîte**

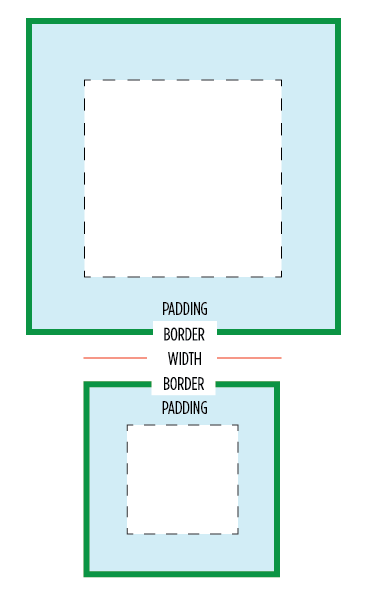
Depuis le début de CSS, le [modèle de boîte](https://css-tricks.com/the-css-box-model/) a fonctionné comme ceci par défaut:

width + padding + border = largeur visible / rendue réelle de la boîte d'un élément

height + padding + border = hauteur visible / rendue réelle de la boîte d'un élément

Cela peut être un peu contre-intuitif, car la largeur et la hauteur que vous définissez pour un élément sortent de la fenêtre dès que vous commencez à ajouter du remplissage et des bordures à l'élément.

À l'époque de la conception Web, les premières versions d'Internet Explorer (<= IE6) géraient le modèle de boîte différemment lorsqu'il était en "mode quirks". Le modèle de boîte "quirks" fonctionnait comme suit: width = largeur visible / rendue réelle de la hauteur de la boîte d'un élément = hauteur visible / rendue réelle de la boîte d'un élément Les valeurs de bordure et de remplissage ont été déplacées à l'intérieur de la boîte de l'élément, en coupant dans la largeur / hauteur de la boîte plutôt que de l'élargir.

****

La boîte en haut montre le modèle de boîte par défaut. La boîte en bas montre ce qui était autrefois l'interprétation du "mode quirks" du modèle de boîte.

Certaines personnes ont préféré cette interprétation "décalée" du modèle de boîte et l'ont considérée plus intuitive. C'est un point valable. Avoir la largeur visible réelle d'une boîte s'avère être différent de ce que vous avez déclaré dans le CSS, c'est un peu compliqué.

Mais à l’époque de la conception à largeur fixe, il n’était pas particulièrement compliqué de travailler avec le modèle par défaut une fois que vous l’aviez compris. Vous pouvez faire un peu d'arithmétique pour déterminer le nombre de pixels dont vous avez besoin pour réduire la largeur ou la hauteur déclarée d'un élément afin de l'adapter à son remplissage et à sa bordure. Le problème pour les développeurs actuels est que ces longueurs absolues de pixels ne se traduisent pas par une conception réactive, de sorte que les mêmes calculs ne s'appliquent plus.

Dès que la conception réactive (ou, comme on l'appelait autrefois, la présentation «fluide» ou «liquide») commençait à gagner en popularité, les développeurs et les concepteurs souhaitaient une mise à jour du modèle de boîte. Le grand designer [Jon Hicks](http://www.hicksdesign.co.uk/) , connu pour ses excellentes conceptions en largeur fluide, a dit ceci à ce sujet dans la [liste de souhaits CSS que](https://css-tricks.com/css-wishlist/) nous avons créée en 2008:

J'aimerais un modèle de boîte différent! Je trouve étrange que le rembourrage et la bordure ajoutent la largeur d'un objet et j'aimerais pouvoir donner quelque chose comme un textarea 100% width et un rembourrage 3px sans se soucier de ce que la disposition va faire. Peut-être que quelque chose comme padding-inside en tant que nouveau sélecteur?

Dans ce sens, j'aimerais aussi pouvoir spécifier une largeur de 100% pour un élément, moins une largeur fixe. Encore une fois, très utile pour créer des conceptions fluides avec des éléments de formulaire!

### **Présent-jourbox-sizing**

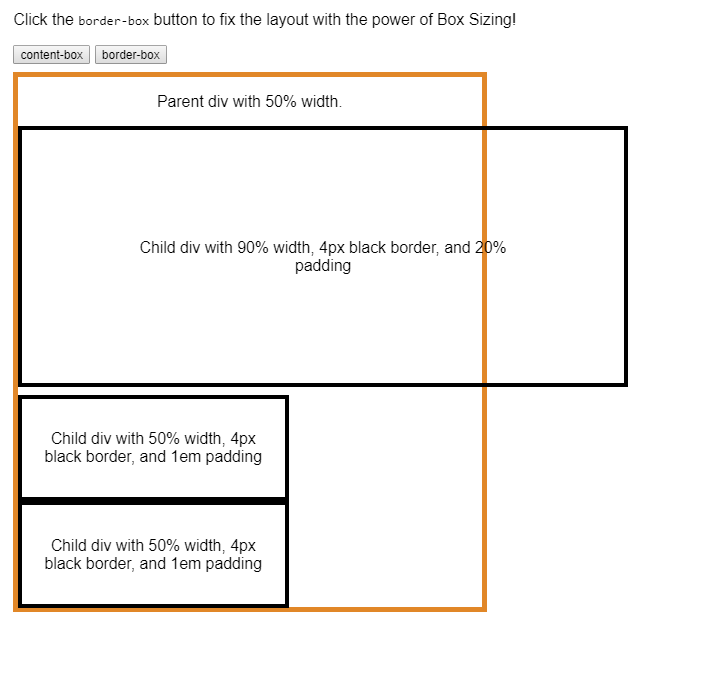
Ces souhaits ont été exaucés lorsque la box-sizingpropriété a été introduite dans CSS3. Bien que box-sizingtrois valeurs ( content-box, padding-boxet border-box), la valeur la plus populaire est border-box.

Aujourd'hui, les versions actuelles de tous les navigateurs utilisent le modèle de boîte d'origine "largeur ou hauteur + remplissage + bordure = largeur ou hauteur réelle". Avec box-sizing: border-box;, nous pouvons changer le modèle de boîte à ce qui était autrefois la manière "bizarre", où la largeur et la hauteur spécifiées d'un élément ne sont pas affectées par le remplissage ou les bordures. Cela s’est avéré si utile dans le design réactif qu’il a trouvé sa place dans les styles de réinitialisation.

À ce stade, vous vous demandez peut-être: "Est-il possible que Old IE ait fait quelque chose de bien?" [Beaucoup de gens le pensent](http://www.jefftk.com/p/the-revenge-of-the-ie-box-model) .

### **Démo**

Cette démo montre comment border-boxrendre les mises en page réactives plus faciles à gérer. La divlargeur du parent est de 50% et il a 3 enfants avec des largeurs, des rembourrages et des marges différentes. Cliquez sur le border-boxbouton pour que tous les enfants se trouvent au bon endroit dans le parent.



### **Bonnes, meilleures et (probablement) meilleures box-sizing méthodes de réinitialisation**

#### **Le "vieux" border-boxReset**

La première box-sizing: border-box;réinitialisation ressemblait à ceci:

\* {

box-sizing: border-box;

}

Cela fonctionne assez bien, mais il ne contient pas de pseudo-éléments, ce qui peut entraîner des résultats inattendus. Une réinitialisation révisée couvrant les pseudo-éléments est rapidement apparue:

#### Dimensionnement de la boîte universelle

\*, \*:before, \*:after {

box-sizing: border-box;

}

Cette méthode a également sélectionné des pseudo-éléments, améliorant ainsi l'effet de normalisation de border-box. Mais, le \*sélecteur rend difficile l’utilisation par les développeurs content-boxou padding-boxailleurs dans le CSS. Ce qui nous amène au favori actuel pour les meilleures pratiques:

#### **Dimensionnement de boîte universelle avec héritage**

html {

box-sizing: border-box;

}

\*, \*:before, \*:after {

box-sizing: inherit;

}

Cette réinitialisation vous donne plus de flexibilité que ses prédécesseurs - vous pouvez utiliser content-boxou padding-box(le cas échéant) à votre guise, sans vous soucier d'un sélecteur universel remplaçant votre CSS. Nous avons approfondi cette technique et les raisons qui la sous-tendent dans ["Hériter box-sizingprobablement de meilleures pratiques optimales"](https://css-tricks.com/inheriting-box-sizing-probably-slightly-better-best-practice/) . Un problème potentiel est que ce box-sizingn’est pas normalement hérité, il s’agit donc d’un comportement spécialisé, qui n’est pas tout à fait la même chose que ce que vous mettriez normalement dans une réinitialisation.

### **Préfixes du vendeur**

Chaque navigateur actuel prend en charge les box-sizing: border-box;préfixes sans préfixe, de sorte que la nécessité de préfixes de fournisseur s'estompe. Toutefois, si vous devez prendre en charge les anciennes versions de Safari (<5.1), Chrome (<10) et Firefox (<29), vous devez inclure les préfixes -webkitet -moz, comme ceci:

html {

-webkit-box-sizing: border-box;

-moz-box-sizing: border-box;

box-sizing: border-box;

}

\*, \*:before, \*:after {

-webkit-box-sizing: inherit;

-moz-box-sizing: inherit;

box-sizing: inherit;

}

### **Problèmes connus**

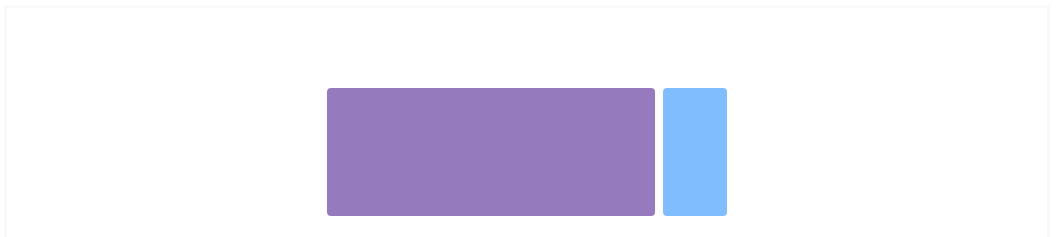
box-sizing: border-box;est pris en charge dans les versions actuelles de tous les principaux navigateurs. Le moins couramment utilisé padding-boxn'est supporté que par Firefox pour le moment. Notre [box-sizing](https://css-tricks.com/almanac/properties/b/box-sizing/)entrée almanach contient des informations plus complètes sur la prise en charge du navigateur .

Il existe quelques problèmes avec les anciennes versions d'Internet Explorer (8 et plus anciennes). IE 8 ne reconnaît pas border-boxles éléments avec min/max-widthou min/max-height(ceci affectait également Firefox, mais cela a été [corrigé en 2012](https://bugzilla.mozilla.org/show_bug.cgi?id=308801) ). IE 7 et les versions antérieures ne reconnaissent pas box-sizingdu tout, mais il existe un [polyfill qui peut aider](https://github.com/Schepp/box-sizing-polyfill) .

## **Les conteneurs**

Les conteneurs sont l’élément de présentation le plus élémentaire de Bootstrap et sont **indispensables pour l’utilisation de notre système de grille par défaut** . Choisissez un conteneur sensible à largeur fixe (ce qui signifie ses max-widthmodifications à chaque point d'arrêt) ou une largeur de fluide (ce qui signifie qu'il est 100%large tout le temps).

Bien que les conteneurs puissent être imbriqués, la plupart des mises en page ne nécessitent pas de conteneur imbriqué.

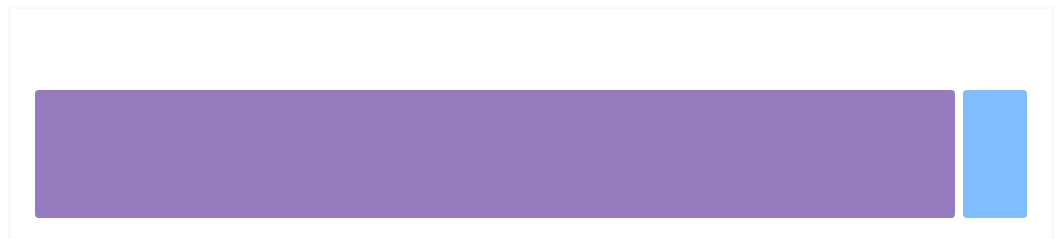


<div class="container">

<!-- Content here -->

</div>

Utilisez ce paramètre .container-fluidpour un conteneur de largeur complète, couvrant toute la largeur de la fenêtre.



<div class="container-fluid">

...

</div>

## **Points de rupture réactifs**

Bootstrap étant conçu pour être mobile d’abord, nous utilisons une poignée de [requêtes](https://developer.mozilla.org/en-US/docs/Web/CSS/Media_Queries/Using_media_queries) de [support](https://developer.mozilla.org/en-US/docs/Web/CSS/Media_Queries/Using_media_queries) pour créer des points de rupture sensibles pour nos présentations et nos interfaces. Ces points d'arrêt sont principalement basés sur les largeurs minimales de la fenêtre d'affichage et nous permettent de redimensionner les éléments à mesure que la fenêtre d'affichage change.

Bootstrap utilise principalement les plages de requête de média suivantes (ou points d'arrêt) dans nos fichiers Sass source pour notre structure, notre système de grille et nos composants.

// Extra small devices (portrait phones, less than 576px)

// No media query for `xs` since this is the default in Bootstrap

// Small devices (landscape phones, 576px and up)

@media (min-width: 576px) { ... }

// Medium devices (tablets, 768px and up)

@media (min-width: 768px) { ... }

// Large devices (desktops, 992px and up)

@media (min-width: 992px) { ... }

// Extra large devices (large desktops, 1200px and up)

@media (min-width: 1200px) { ... }

Depuis que nous écrivons notre code source CSS dans Sass, toutes nos requêtes multimédia sont disponibles via les mixins Sass:

// No media query necessary for xs breakpoint as it's effectively `@media (min-width: 0) { ... }`

@include media-breakpoint-up(sm) { ... }

@include media-breakpoint-up(md) { ... }

@include media-breakpoint-up(lg) { ... }

@include media-breakpoint-up(xl) { ... }

// Example: Hide starting at `min-width: 0`, and then show at the `sm` breakpoint

.custom-class {

display: none;

}

@include media-breakpoint-up(sm) {

.custom-class {

display: block;

}

}

Nous utilisons occasionnellement des requêtes multimédia qui vont dans le sens opposé (taille d'écran donnée ou inférieure ):

// Extra small devices (portrait phones, less than 576px)

@media (max-width: 575.98px) { ... }

// Small devices (landscape phones, less than 768px)

@media (max-width: 767.98px) { ... }

// Medium devices (tablets, less than 992px)

@media (max-width: 991.98px) { ... }

// Large devices (desktops, less than 1200px)

@media (max-width: 1199.98px) { ... }

// Extra large devices (large desktops)

// No media query since the extra-large breakpoint has no upper bound on its width

Notez que, dans la mesure où les navigateurs ne prennent actuellement pas en charge les [requêtes de contexte de plage](https://www.w3.org/TR/mediaqueries-4/#range-context) , nous contournons les limitations [min-et max-préfixes,](https://www.w3.org/TR/mediaqueries-4/" \l "mq-min-max) ainsi que les fenêtres de présentation avec des largeurs fractionnaires (pouvant se produire dans certaines conditions sur les appareils à haute résolution, par exemple) en utilisant des valeurs plus précises pour ces comparaisons. .

Une fois encore, ces requêtes médiatiques sont également disponibles via les mixins Sass:

@include media-breakpoint-down(xs) { ... }

@include media-breakpoint-down(sm) { ... }

@include media-breakpoint-down(md) { ... }

@include media-breakpoint-down(lg) { ... }

// No media query necessary for xl breakpoint as it has no upper bound on its width

// Example: Style from medium breakpoint and down

@include media-breakpoint-down(md) {

.custom-class {

display: block;

}

}

Il existe également des requêtes multimédia et des mixines permettant de cibler un seul segment de tailles d'écran en utilisant les largeurs minimale et maximale des points d'arrêt.

// Extra small devices (portrait phones, less than 576px)

@media (max-width: 575.98px) { ... }

// Small devices (landscape phones, 576px and up)

@media (min-width: 576px) and (max-width: 767.98px) { ... }

// Medium devices (tablets, 768px and up)

@media (min-width: 768px) and (max-width: 991.98px) { ... }

// Large devices (desktops, 992px and up)

@media (min-width: 992px) and (max-width: 1199.98px) { ... }

// Extra large devices (large desktops, 1200px and up)

@media (min-width: 1200px) { ... }

Ces requêtes médiatiques sont également disponibles via les mixins Sass:

@include media-breakpoint-only(xs) { ... }

@include media-breakpoint-only(sm) { ... }

@include media-breakpoint-only(md) { ... }

@include media-breakpoint-only(lg) { ... }

@include media-breakpoint-only(xl) { ... }

De même, les requêtes multimédias peuvent s'étendre sur plusieurs largeurs de point d'arrêt:

Co // Example

// Apply styles starting from medium devices and up to extra large devices

@media (min-width: 768px) and (max-width: 1199.98px) { ... }

Le mixin Sass pour cibler la même plage de taille d’écran serait:

@include media-breakpoint-between(md, xl) { ... }

## **Z-index**

Plusieurs composants Bootstrap utilisent z-index, la propriété CSS qui permet de contrôler la mise en page en fournissant un troisième axe pour organiser le contenu. Dans Bootstrap, nous utilisons une échelle d'index-z par défaut conçue pour superposer correctement la navigation, les info-bulles et les popovers, les modaux, etc.

Ces valeurs supérieures commencent par un nombre arbitraire, élevé et suffisamment spécifique pour éviter idéalement les conflits. Nous avons besoin d'un ensemble standard de ces composants dans nos composants en couches (info-bulles, popovers, barres de navigation, menus déroulants, modaux) pour que nous puissions être raisonnablement cohérents dans les comportements. Il n'y a aucune raison pour laquelle nous n'aurions pas pu utiliser 100+ ou 500+.

Nous n'encourageons pas la personnalisation de ces valeurs individuelles; Si vous en changez un, vous devrez probablement les changer tous.

$zindex-dropdown: 1000 !default;

$zindex-sticky: 1020 !default;

$zindex-fixed: 1030 !default;

$zindex-modal-backdrop: 1040 !default;

$zindex-modal: 1050 !default;

$zindex-popover: 1060 !default;

$zindex-tooltip: 1070 !default;

Pour gérer les frontières qui se chevauchent à l' intérieur des composants (par exemple, des boutons et des entrées dans les groupes d'entrée), on utilise de faibles chiffres unique z-indexvaleurs 1, 2et 3pour défaut, en vol stationnaire, et des états actifs. En survol / focus / actif, nous mettons au premier plan un élément particulier avec une z-indexvaleur plus élevée pour afficher sa bordure au-dessus des éléments frères.

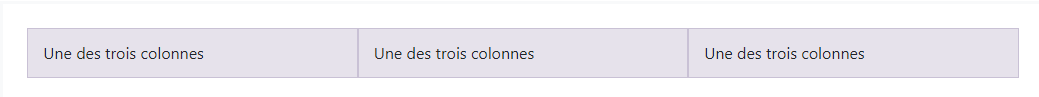
# Système de grille

Utilisez notre puissante grille flex-mobile d'abord pour créer des mises en page de formes et de tailles variées grâce à un système à douze colonnes, à cinq niveaux réactifs par défaut, à des variables Sass et à des mixines, ainsi qu'à des dizaines de classes prédéfinies.

## **Comment ça marche**

Le système de grille de Bootstrap utilise une série de conteneurs, de lignes et de colonnes pour mettre en forme et aligner le contenu. Il est construit avec [flexbox](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Flexible_Box_Layout/Basic_Concepts_of_Flexbox) et est entièrement réactif. Vous trouverez ci-dessous un exemple et un aperçu détaillé de la manière dont la grille est réalisée.

**Nouveau ou peu familier avec flexbox?**[Lisez ce guide](https://css-tricks.com/snippets/css/a-guide-to-flexbox/#flexbox-background) sur les [astuces CSS flexbox](https://css-tricks.com/snippets/css/a-guide-to-flexbox/#flexbox-background) pour [connaître l’](https://css-tricks.com/snippets/css/a-guide-to-flexbox/#flexbox-background) arrière-plan, la terminologie, les directives et les extraits de code.



<div class="container">

<div class="row">

<div class="col-sm">

One of three columns

</div>

<div class="col-sm">

One of three columns

</div>

<div class="col-sm">

One of three columns

</div>

</div>

</div>

L'exemple ci-dessus crée trois colonnes de largeur égale sur des périphériques petits, moyens, grands et très grands à l'aide de nos classes de grille prédéfinies. Ces colonnes sont centrées dans la page avec le parent .container.

Voici comment cela fonctionne:

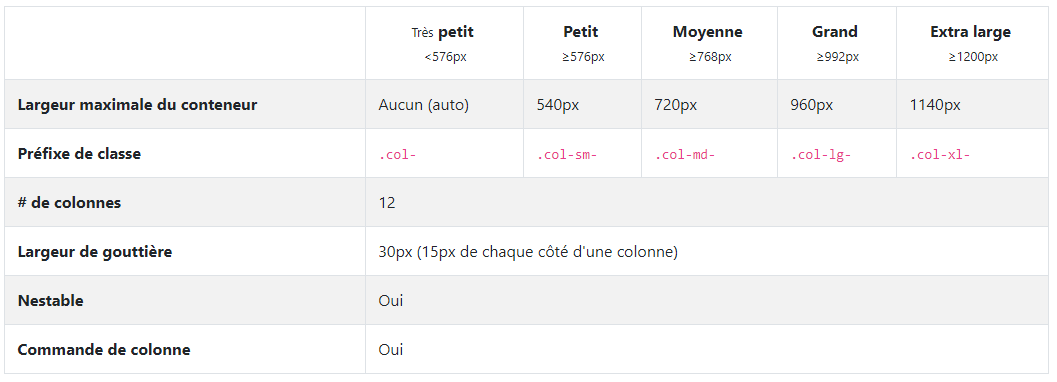
* Les conteneurs permettent de centrer horizontalement le contenu de votre site. Utilisez cette option .containerpour une largeur de pixel sensible ou .container-fluidpour width: 100%toutes les tailles de fenêtres et de périphériques.
* Les lignes sont des wrappers pour les colonnes. Chaque colonne a horizontal padding(appelé gouttière) pour contrôler l’espace entre elles. Ceci paddingest ensuite compensé sur les lignes avec des marges négatives. De cette manière, tout le contenu de vos colonnes est aligné visuellement sur le côté gauche.
* Dans une présentation en grille, le contenu doit être placé dans des colonnes et seules les colonnes peuvent être des enfants immédiats de lignes.
* Grâce à flexbox, les colonnes de grille non spécifiées widthseront automatiquement mises en page sous forme de colonnes de largeur égale. Par exemple, quatre occurrences de .col-smchacune auront automatiquement une largeur de 25% à partir du petit point d'arrêt et plus. Voir la section des [colonnes de mise en page automatique](https://getbootstrap.com/docs/4.2/layout/grid/#auto-layout-columns)pour plus d'exemples.
* Les classes de colonnes indiquent le nombre de colonnes que vous souhaitez utiliser sur les 12 possibles par ligne. Donc, si vous voulez trois colonnes de largeur égale, vous pouvez utiliser .col-4.
* Les colonnes widths sont définies en pourcentages, elles sont donc toujours fluides et dimensionnées par rapport à leur élément parent.
* Les colonnes sont horizontales paddingpour créer les gouttières entre les colonnes individuelles. Toutefois, vous pouvez supprimer les marginlignes et les paddingcolonnes avec .no-guttersle .row.
* Pour que le réseau soit réactif, il existe cinq points d'arrêt de réseau, un pour chaque [point d'arrêt réactif](https://getbootstrap.com/docs/4.2/layout/overview/#responsive-breakpoints) : tous les points d'arrêt (très petits), petits, moyens, grands et très grands.
* Les points d'arrêt de la grille sont basés sur des requêtes de support de largeur minimale, ce **qui** signifie **qu'elles s'appliquent à ce point d'arrêt et à tous ceux qui le précèdent** (par exemple, .col-sm-4s'applique aux périphériques petits, moyens, grands et très grands, mais pas au premier xspoint d'arrêt).
* Vous pouvez utiliser des classes de grille prédéfinies (comme .col-4) ou des [mixins Sass](https://getbootstrap.com/docs/4.2/layout/grid/" \l "sass-mixins) pour un balisage plus sémantique.

Soyez conscient des limitations et des [bugs liés à flexbox](https://github.com/philipwalton/flexbugs) , tels que l' [impossibilité d'utiliser certains éléments HTML en tant que conteneurs flex](https://github.com/philipwalton/flexbugs#flexbug-9) .

## **Options de grille**

Alors que Bootstrap utilise ems ou rems pour définir la plupart des tailles, pxs est utilisé pour les points d'arrêt de la grille et les largeurs de conteneur. En effet, la largeur de la fenêtre d'affichage est en pixels et ne change pas avec la [taille de](https://drafts.csswg.org/mediaqueries-3/#units) la [police](https://drafts.csswg.org/mediaqueries-3/#units) .

Découvrez comment certains aspects du système de grille Bootstrap fonctionnent sur plusieurs périphériques grâce à un tableau pratique.

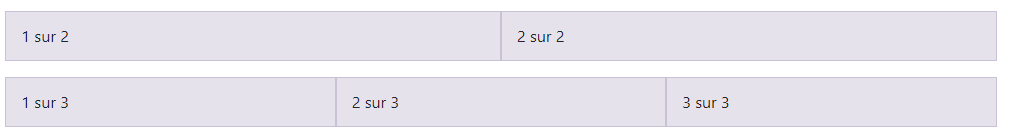


## **Colonnes à mise en page automatique**

Utilisez des classes de colonnes spécifiques aux points d'arrêt pour faciliter le dimensionnement des colonnes sans classe explicite numérotée .col-sm-6.

### **Largeur égale**

Par exemple, voici deux dispositions de grille qui s'appliquent à chaque périphérique et fenêtre d'affichage, de xsà xl. Ajoutez un nombre illimité de classes sans unité pour chaque point d'arrêt dont vous avez besoin et chaque colonne aura la même largeur.



<div class="container">

<div class="row">

<div class="col">

1 of 2

</div>

<div class="col">

2 of 2

</div>

</div>

<div class="row">

<div class="col">

1 of 3

</div>

<div class="col">

2 of 3

</div>

<div class="col">

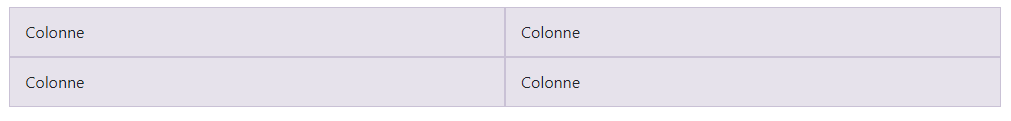
3 of 3

</div>

</div>

</div>

Les colonnes de largeur égale peuvent être divisées en plusieurs lignes, mais il y avait un [bogue Safari flexbox](https://github.com/philipwalton/flexbugs#flexbug-11) qui l'empêchait de fonctionner sans un flex-basisou explicite border. Il existe des solutions de contournement pour les anciennes versions de navigateur, mais elles ne devraient pas être nécessaires si vous êtes à jour.



<div class="container">

<div class="row">

<div class="col">Column</div>

<div class="col">Column</div>

<div class="w-100"></div>

<div class="col">Column</div>

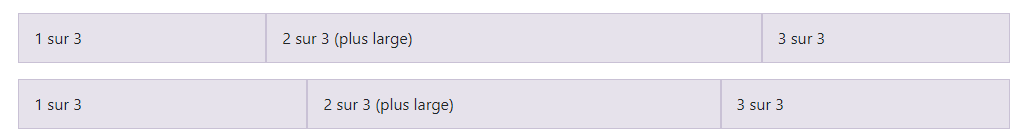
<div class="col">Column</div>

</div>

</div>

### **Définir une largeur de colonne**

La mise en page automatique des colonnes de la grille flexbox signifie également que vous pouvez définir la largeur d'une colonne et redimensionner automatiquement les colonnes sœurs. Vous pouvez utiliser des classes de grille prédéfinies (comme indiqué ci-dessous), des combinaisons de grille ou des largeurs en ligne. Notez que les autres colonnes seront redimensionnées quelle que soit la largeur de la colonne centrale.



<div class="container">

<div class="row">

<div class="col">

1 of 3

</div>

<div class="col-6">

2 of 3 (wider)

</div>

<div class="col">

3 of 3

</div>

</div>

<div class="row">

<div class="col">

1 of 3

</div>

<div class="col-5">

2 of 3 (wider)

</div>

<div class="col">

3 of 3

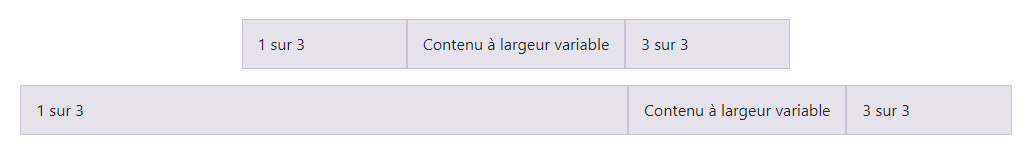
</div>

</div>

</div>

### **Contenu à largeur variable**

Utilisez des col-{breakpoint}-autoclasses pour dimensionner les colonnes en fonction de la largeur naturelle de leur contenu



<div class="container">

<div class="row justify-content-md-center">

<div class="col col-lg-2">

1 of 3

</div>

<div class="col-md-auto">

Variable width content

</div>

<div class="col col-lg-2">

3 of 3

</div>

</div>

<div class="row">

<div class="col">

1 of 3

</div>

<div class="col-md-auto">

Variable width content

</div>

<div class="col col-lg-2">

3 of 3

</div>

</div>

</div><div class="container">

<div class="row">

<div class="col">

1 of 3

</div>

<div class="col-6">

2 of 3 (wider)

</div>

<div class="col">

3 of 3

</div>

</div>

<div class="row">

<div class="col">

1 of 3

</div>

<div class="col-5">

2 of 3 (wider)

</div>

<div class="col">

3 of 3

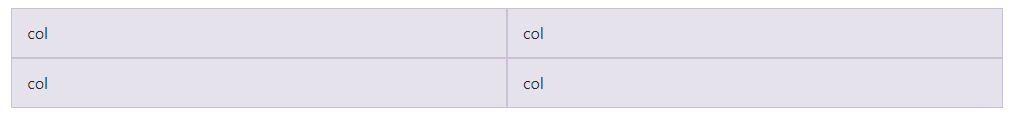
</div>

</div>

</div>

### **Plusieurs rangées à largeur égale**

Créez des colonnes de largeur égale qui s'étendent sur plusieurs lignes en insérant un   
.w-100emplacement où vous souhaitez que les colonnes se transforment en une nouvelle ligne. Rendez les pauses réactives en les associant .w-100à des [utilitaires d’affichage réactifs](https://getbootstrap.com/docs/4.2/utilities/display/) .



<div class="container">

<div class="row">

<div class="col">col</div>

<div class="col">col</div>

<div class="w-100"></div>

<div class="col">col</div>

<div class="col">col</div>

</div>

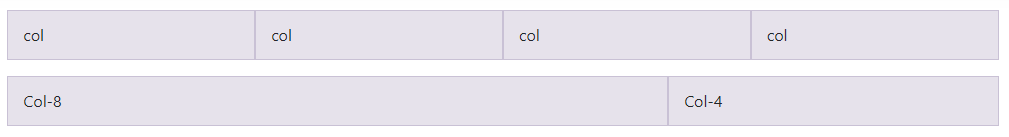
</div>

## **Classes réactives**

La grille de Bootstrap comprend cinq niveaux de classes prédéfinies pour la création de dispositions complexes sensibles. Personnalisez la taille de vos colonnes sur des périphériques de taille très petite, petite, moyenne, grande ou très grande, comme bon vous semble.

### **Tous les points d'arrêt**

Pour les grilles identiques des plus petits aux plus grands, utilisez les classes .col et  
 .col-\*. Spécifiez une classe numérotée lorsque vous avez besoin d'une colonne de taille particulière. sinon, n'hésitez pas à vous en tenir à .col.



<div class="container">

<div class="row">

<div class="col">col</div>

<div class="col">col</div>

<div class="col">col</div>

<div class="col">col</div>

</div>

<div class="row">

<div class="col-8">col-8</div>

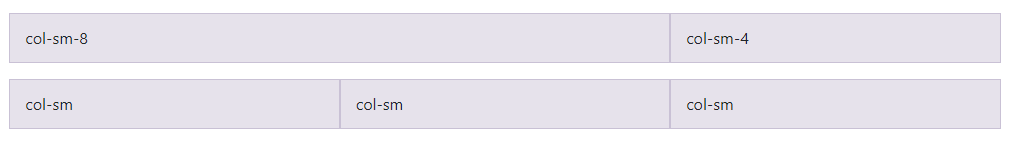
<div class="col-4">col-4</div>

</div>

</div>

### **Empilé à l'horizontale**

En utilisant un seul ensemble de .col-sm-\*classes, vous pouvez créer un système de grille de base qui commence par être empilé et devient horizontal au petit point d'arrêt ( sm).



<div class="container">

<div class="row">

<div class="col-sm-8">col-sm-8</div>

<div class="col-sm-4">col-sm-4</div>

</div>

<div class="row">

<div class="col-sm">col-sm</div>

<div class="col-sm">col-sm</div>

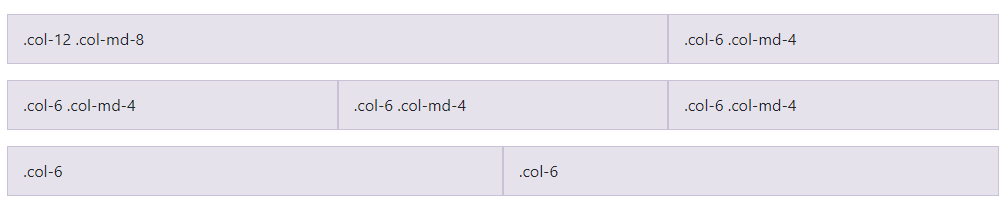
<div class="col-sm">col-sm</div>

</div>

</div>

### **Mélanger et assortir**

Vous ne voulez pas que vos colonnes s'empilent simplement dans des niveaux de grille? Utilisez une combinaison de différentes classes pour chaque niveau, selon vos besoins. Voir l'exemple ci-dessous pour une meilleure idée du fonctionnement de tout cela.



<div class="container">

<!-- Stack the columns on mobile by making one full-width and the other half-width -->

<div class="row">

<div class="col-12 col-md-8">.col-12 .col-md-8</div>

<div class="col-6 col-md-4">.col-6 .col-md-4</div>

</div>

<!-- Columns start at 50% wide on mobile and bump up to 33.3% wide on desktop -->

<div class="row">

<div class="col-6 col-md-4">.col-6 .col-md-4</div>

<div class="col-6 col-md-4">.col-6 .col-md-4</div>

<div class="col-6 col-md-4">.col-6 .col-md-4</div>

</div>

<!-- Columns are always 50% wide, on mobile and desktop -->

<div class="row">

<div class="col-6">.col-6</div>

<div class="col-6">.col-6</div>

</div>

</div>

### **Gouttières**

Les gouttières peuvent être ajustées en fonction des classes de remplissage et des marges négatives spécifiques aux points d'arrêt. Pour changer les gouttières d'une ligne donnée, associez un utilitaire de marge négative à l' .rowutilitaire de remplissage et les utilitaires de remplissage correspondants du fichier .cols. Vous devrez peut-être également ajuster le parent .containerou .container-fluidpour éviter les débordements indésirables, en utilisant à nouveau l’utilitaire de remplissage correspondant.

Voici un exemple de personnalisation de la grille Bootstrap au lgpoint d'arrêt large ( ) et aux niveaux supérieurs. Nous avons augmenté le .colremplissage avec .px-lg-5, compensé cela avec .mx-lg-n5sur le parent .row, puis ajusté le .containerwrapper avec .px-lg-5.



<div class="container px-lg-5">

<div class="row mx-lg-n5">

<div class="col py-3 px-lg-5 border bg-light">Custom column padding</div>

<div class="col py-3 px-lg-5 border bg-light">Custom column padding</div>

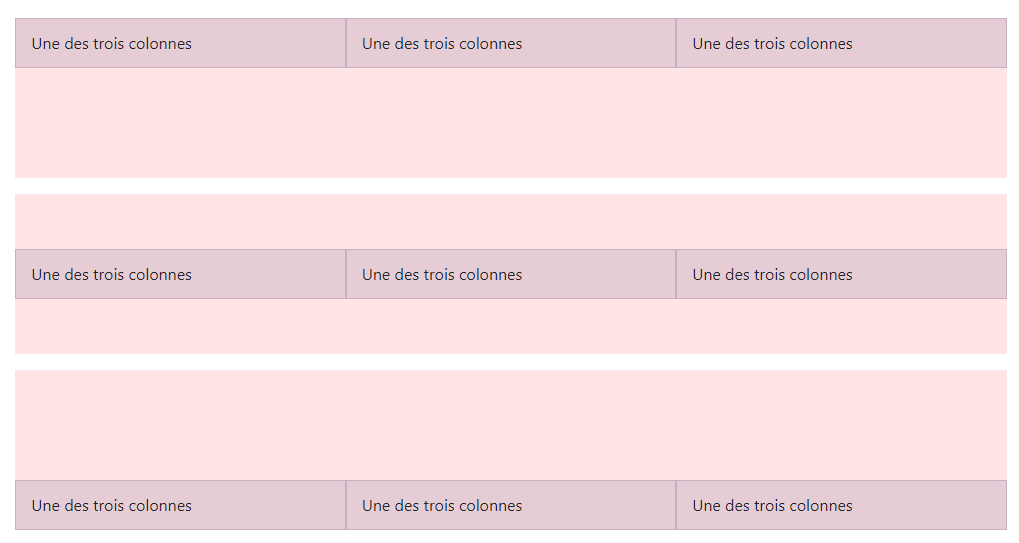
</div>

</div>

## **Alignement**

Utilisez les utilitaires d'alignement flexbox pour aligner les colonnes verticalement et horizontalement.

### **Alignement vertical**



<div class="container">

<div class="row align-items-start">

<div class="col">

One of three columns

</div>

<div class="col">

One of three columns

</div>

<div class="col">

One of three columns

</div>

</div>

<div class="row align-items-center">

<div class="col">

One of three columns

</div>

<div class="col">

One of three columns

</div>

<div class="col">

One of three columns

</div>

</div>

<div class="row align-items-end">

<div class="col">

One of three columns

</div>

<div class="col">

One of three columns

</div>

<div class="col">

One of three columns

</div>

</div>

</div>

Une des trois colonnes

Une des trois colonnes

Une des trois colonnes

Copie

<div class="container">

<div class="row">

<div class="col align-self-start">

One of three columns

</div>

<div class="col align-self-center">

One of three columns

</div>

<div class="col align-self-end">

One of three columns

</div>

</div>

</div>

### **Alignement horizontal**

Une des deux colonnes

Une des deux colonnes

Une des deux colonnes

Une des deux colonnes

Une des deux colonnes

Une des deux colonnes

Une des deux colonnes

Une des deux colonnes

Une des deux colonnes

Une des deux colonnes

Copie

<div class="container">

<div class="row justify-content-start">

<div class="col-4">

One of two columns

</div>

<div class="col-4">

One of two columns

</div>

</div>

<div class="row justify-content-center">

<div class="col-4">

One of two columns

</div>

<div class="col-4">

One of two columns

</div>

</div>

<div class="row justify-content-end">

<div class="col-4">

One of two columns

</div>

<div class="col-4">

One of two columns

</div>

</div>

<div class="row justify-content-around">

<div class="col-4">

One of two columns

</div>

<div class="col-4">

One of two columns

</div>

</div>

<div class="row justify-content-between">

<div class="col-4">

One of two columns

</div>

<div class="col-4">

One of two columns

</div>

</div>

</div>

### **Pas de gouttières**

Les gouttières entre les colonnes de nos classes de grille prédéfinies peuvent être supprimées avec .no-gutters. Cela supprime les négatifs margins de .rowet l'horizontale paddingde toutes les colonnes des enfants immédiats.

Voici le code source pour créer ces styles. Notez que les remplacements de colonne ne concernent que les premières colonnes enfants et sont ciblés via le [sélecteur d'attribut](https://developer.mozilla.org/en-US/docs/Web/CSS/Attribute_selectors) . Bien que cela génère un sélecteur plus spécifique, le remplissage des colonnes peut toujours être personnalisé avec [des utilitaires d'espacement](https://getbootstrap.com/docs/4.2/utilities/spacing/) .

**Besoin d'un design de bout en bout?**Supprimez le parent .containerou .container-fluid.

Copie

.no-gutters {

margin-right: 0;

margin-left: 0;

> .col,

> [class\*="col-"] {

padding-right: 0;

padding-left: 0;

}

}

En pratique, voici à quoi cela ressemble. Notez que vous pouvez continuer à l'utiliser avec toutes les autres classes de grille prédéfinies (y compris les largeurs de colonne, les niveaux réactifs, les réordonnances, etc.).

.col-12 .col-sm-6 .col-md-8

.col-6 .col-md-4

Copie

<div class="row no-gutters">

<div class="col-12 col-sm-6 col-md-8">.col-12 .col-sm-6 .col-md-8</div>

<div class="col-6 col-md-4">.col-6 .col-md-4</div>

</div>

### **Enveloppement de colonne**

Si plus de 12 colonnes sont placées dans une seule ligne, chaque groupe de colonnes supplémentaires sera inséré dans une nouvelle ligne.

.col-9

.col-4   
Puisque 9 + 4 = 13> 12, cette division de 4 colonnes s’allonge sur une nouvelle ligne sous la forme d’une unité contiguë.

.col-6   
Les colonnes suivantes continuent le long de la nouvelle ligne.

Copie

<div class="container">

<div class="row">

<div class="col-9">.col-9</div>

<div class="col-4">.col-4<br>Since 9 + 4 = 13 &gt; 12, this 4-column-wide div gets wrapped onto a new line as one contiguous unit.</div>

<div class="col-6">.col-6<br>Subsequent columns continue along the new line.</div>

</div>

</div>

### **Sauts de colonne**

Réduire les colonnes en une nouvelle ligne dans flexbox nécessite un petit hack: ajoutez un élément avec l' width: 100%endroit où vous souhaitez insérer vos colonnes dans une nouvelle ligne. Normalement, ceci est accompli avec plusieurs .rows, mais toutes les méthodes d'implémentation ne peuvent en rendre compte.

.col-6 .col-sm-3

.col-6 .col-sm-3

.col-6 .col-sm-3

.col-6 .col-sm-3

Copie

<div class="container">

<div class="row">

<div class="col-6 col-sm-3">.col-6 .col-sm-3</div>

<div class="col-6 col-sm-3">.col-6 .col-sm-3</div>

<!-- Force next columns to break to new line -->

<div class="w-100"></div>

<div class="col-6 col-sm-3">.col-6 .col-sm-3</div>

<div class="col-6 col-sm-3">.col-6 .col-sm-3</div>

</div>

</div>

Vous pouvez également appliquer cette pause à des points d'arrêt spécifiques avec nos [utilitaires d'affichage réactifs](https://getbootstrap.com/docs/4.2/utilities/display/) .

.col-6 .col-sm-4

.col-6 .col-sm-4

.col-6 .col-sm-4

.col-6 .col-sm-4

Copie

<div class="container">

<div class="row">

<div class="col-6 col-sm-4">.col-6 .col-sm-4</div>

<div class="col-6 col-sm-4">.col-6 .col-sm-4</div>

<!-- Force next columns to break to new line at md breakpoint and up -->

<div class="w-100 d-none d-md-block"></div>

<div class="col-6 col-sm-4">.col-6 .col-sm-4</div>

<div class="col-6 col-sm-4">.col-6 .col-sm-4</div>

</div>

</div>

## **Réordonner**

### **Cours de commande**

Utilisez des .order-classes pour contrôler l' **ordre visuel** de votre contenu. Ces classes sont sensibles, vous pouvez donc définir le orderpoint d'arrêt par (par exemple, .order-1.order-md-2). Inclut le support pour 1par l' 12ensemble des cinq niveaux de la grille.

D'abord, mais non ordonné

Deuxième, mais dernier

Troisièmement, mais d'abord

Copie

<div class="container">

<div class="row">

<div class="col">

First, but unordered

</div>

<div class="col order-12">

Second, but last

</div>

<div class="col order-1">

Third, but first

</div>

</div>

</div>

Il existe également des classes responsive .order-firstet .order-lastqui modifient le contenu orderd'un élément en appliquant order: -1et order: 13( order: $columns + 1), respectivement. Ces classes peuvent également être mélangées avec les .order-\*classes numérotées selon les besoins.

D'abord, mais dernier

Deuxièmement, mais non ordonné

Troisièmement, mais d'abord

Copie

<div class="container">

<div class="row">

<div class="col order-last">

First, but last

</div>

<div class="col">

Second, but unordered

</div>

<div class="col order-first">

Third, but first

</div>

</div>

</div>

### **Compenser les colonnes**

Vous pouvez décaler les colonnes de la grille de deux manières: nos .offset-classes de grille réactives et nos [utilitaires de marge](https://getbootstrap.com/docs/4.2/utilities/spacing/) . Les classes de grille sont dimensionnées pour correspondre aux colonnes tandis que les marges sont plus utiles pour les présentations rapides où la largeur du décalage est variable.

#### **Classes de décalage**

Déplacez les colonnes vers la droite en utilisant des .offset-md-\*classes. Ces classes augmentent la marge gauche d'une colonne par \*colonnes. Par exemple, .offset-md-4déplace .col-md-4sur quatre colonnes.

.col-md-4

.col-md-4 .offset-md-4

.col-md-3 .offset-md-3

.col-md-3 .offset-md-3

.col-md-6 .offset-md-3

Copie

<div class="container">

<div class="row">

<div class="col-md-4">.col-md-4</div>

<div class="col-md-4 offset-md-4">.col-md-4 .offset-md-4</div>

</div>

<div class="row">

<div class="col-md-3 offset-md-3">.col-md-3 .offset-md-3</div>

<div class="col-md-3 offset-md-3">.col-md-3 .offset-md-3</div>

</div>

<div class="row">

<div class="col-md-6 offset-md-3">.col-md-6 .offset-md-3</div>

</div>

</div>

En plus de l'effacement des colonnes aux points d'arrêt réactifs, vous devrez peut-être réinitialiser les décalages. Voir cela en action dans [l'exemple de la grille](https://getbootstrap.com/docs/4.2/examples/grid/) .

.col-sm-5 .col-md-6

.col-sm-5 .offset-sm-2 .col-md-6 .offset-md-0

.col-sm-6 .col-md-5 .col-lg-6

.col-sm-6 .col-md-5 .offset-md-2 .col-lg-6 .offset-lg-0

Copie

<div class="container">

<div class="row">

<div class="col-sm-5 col-md-6">.col-sm-5 .col-md-6</div>

<div class="col-sm-5 offset-sm-2 col-md-6 offset-md-0">.col-sm-5 .offset-sm-2 .col-md-6 .offset-md-0</div>

</div>

<div class="row">

<div class="col-sm-6 col-md-5 col-lg-6">.col-sm-6 .col-md-5 .col-lg-6</div>

<div class="col-sm-6 col-md-5 offset-md-2 col-lg-6 offset-lg-0">.col-sm-6 .col-md-5 .offset-md-2 .col-lg-6 .offset-lg-0</div>

</div>

</div>

#### **Utilitaires de marge**

Avec le passage à flexbox dans la v4, vous pouvez utiliser des utilitaires de marge comme .mr-autopour forcer les colonnes sœurs à s’éloigner les unes des autres.

.col-md-4

.col-md-4 .ml-auto

.col-md-3 .ml-md-auto

.col-md-3 .ml-md-auto

.col-auto .mr-auto

.col-auto

Copie

<div class="container">

<div class="row">

<div class="col-md-4">.col-md-4</div>

<div class="col-md-4 ml-auto">.col-md-4 .ml-auto</div>

</div>

<div class="row">

<div class="col-md-3 ml-md-auto">.col-md-3 .ml-md-auto</div>

<div class="col-md-3 ml-md-auto">.col-md-3 .ml-md-auto</div>

</div>

<div class="row">

<div class="col-auto mr-auto">.col-auto .mr-auto</div>

<div class="col-auto">.col-auto</div>

</div>

</div>

## **Nidification**

Pour imbriquer votre contenu avec la grille par défaut, ajoutez un nouveau .rowet un ensemble de .col-sm-\*colonnes dans une .col-sm-\*colonne existante . Les lignes imbriquées doivent inclure un ensemble de colonnes pouvant aller jusqu'à 12 ou moins (il n'est pas nécessaire que vous utilisiez les 12 colonnes disponibles).

Niveau 1: .col-sm-9

Niveau 2: .col-8 .col-sm-6

Niveau 2: .col-4 .col-sm-6

Copie

<div class="container">

<div class="row">

<div class="col-sm-9">

Level 1: .col-sm-9

<div class="row">

<div class="col-8 col-sm-6">

Level 2: .col-8 .col-sm-6

</div>

<div class="col-4 col-sm-6">

Level 2: .col-4 .col-sm-6

</div>

</div>

</div>

</div>

</div>

## **Sass mixins**

Lorsque vous utilisez les fichiers Sass source de Bootstrap, vous avez la possibilité d’utiliser des variables et des mixins Sass pour créer des présentations de page personnalisées, sémantiques et réactives. Nos classes de grille prédéfinies utilisent ces mêmes variables et combinaisons pour fournir une suite complète de classes prêtes à l'emploi pour des présentations rapides et réactives.

### **Variables**

Les variables et les cartes déterminent le nombre de colonnes, la largeur de la gouttière et le point d'interrogation du support auquel commencer les colonnes flottantes. Nous les utilisons pour générer les classes de grille prédéfinies décrites ci-dessus, ainsi que pour les mixins personnalisés répertoriés ci-dessous.

Copie

$grid-columns: 12;

$grid-gutter-width: 30px;

$grid-breakpoints: (

// Extra small screen / phone

xs: 0,

// Small screen / phone

sm: 576px,

// Medium screen / tablet

md: 768px,

// Large screen / desktop

lg: 992px,

// Extra large screen / wide desktop

xl: 1200px

);

$container-max-widths: (

sm: 540px,

md: 720px,

lg: 960px,

xl: 1140px

);

### **Mixins**

Les mixins sont utilisés avec les variables de grille pour générer des CSS sémantiques pour des colonnes de grille individuelles.

Copie

// Creates a wrapper for a series of columns

@include make-row();

// Make the element grid-ready (applying everything but the width)

@include make-col-ready();

@include make-col($size, $columns: $grid-columns);

// Get fancy by offsetting, or changing the sort order

@include make-col-offset($size, $columns: $grid-columns);

### **Exemple d'utilisation**

Vous pouvez modifier les variables selon vos propres valeurs personnalisées ou simplement utiliser les mixins avec leurs valeurs par défaut. Voici un exemple d'utilisation des paramètres par défaut pour créer une mise en page à deux colonnes avec un espace entre les deux.

Copie

.example-container {

width: 800px;

@include make-container();

}

.example-row {

@include make-row();

}

.example-content-main {

@include make-col-ready();

@include media-breakpoint-up(sm) {

@include make-col(6);

}

@include media-breakpoint-up(lg) {

@include make-col(8);

}

}

.example-content-secondary {

@include make-col-ready();

@include media-breakpoint-up(sm) {

@include make-col(6);

}

@include media-breakpoint-up(lg) {

@include make-col(4);

}

}

Contenu principal

Contenu secondaire

Copie

<div class="example-container">

<div class="example-row">

<div class="example-content-main">Main content</div>

<div class="example-content-secondary">Secondary content</div>

</div>

</div>

## **Personnaliser la grille**

En utilisant nos cartes et variables Sass de grille intégrées, il est possible de personnaliser complètement les classes de grille prédéfinies. Modifiez le nombre de niveaux, les dimensions de la requête multimédia et les largeurs de conteneur, puis recompilez.

### **Colonnes et gouttières**

Le nombre de colonnes de la grille peut être modifié via les variables Sass. $grid-columnsest utilisé pour générer les largeurs (en pourcentage) de chaque colonne individuelle tout en $grid-gutter-widthdéfinissant la largeur des gouttières de colonne.

Copie

$grid-columns: 12 !default;

$grid-gutter-width: 30px !default;

### **Niveaux de grille**

Au-delà des colonnes, vous pouvez également personnaliser le nombre de niveaux de grille. Si vous vouliez seulement quatre niveaux de grille, vous mettriez à jour le $grid-breakpointset $container-max-widthsvers quelque chose comme ceci:

Copie

$grid-breakpoints: (

xs: 0,

sm: 480px,

md: 768px,

lg: 1024px

);

$container-max-widths: (

sm: 420px,

md: 720px,

lg: 960px

);

Lorsque vous apportez des modifications aux variables Sass ou aux cartes, vous devez enregistrer vos modifications et recompiler. Cela produira un nouvel ensemble de classes de grille prédéfinies pour les largeurs de colonne, les décalages et le classement. Les utilitaires de visibilité responsive seront également mis à jour pour utiliser les points d'arrêt personnalisés. Assurez - vous de définir des valeurs de grille dans px(non rem, emou %).