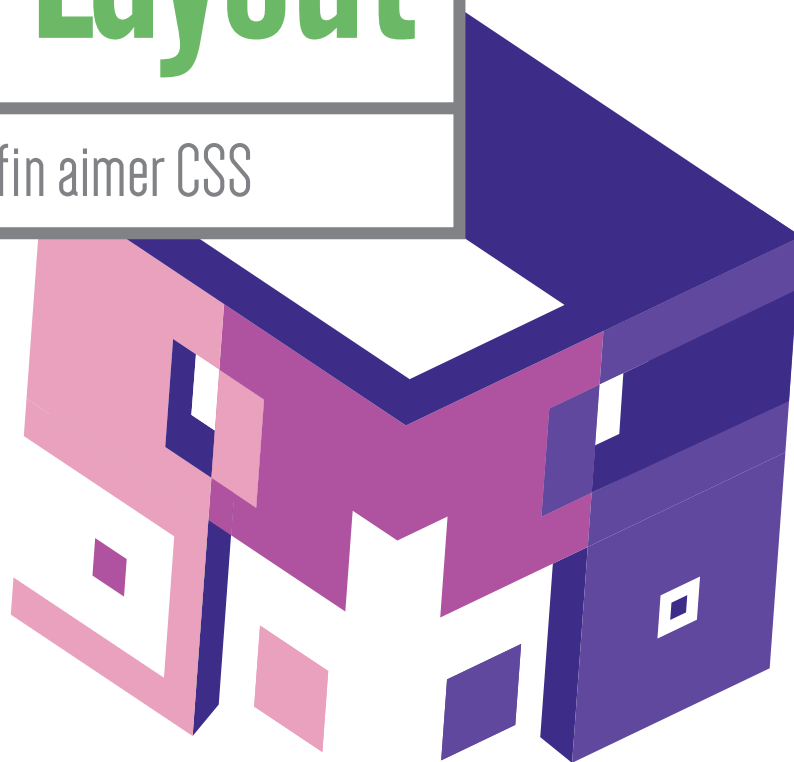


Raphaël Goetter

CSS 3 Grid Layout

Vous allez enfin aimer CSS



● Éditions
EYROLLES

CSS 3 Grid Layout

Un ouvrage de référence pour les webdesigners et intégrateurs

Complémentaire de Flexbox, le module de positionnement Grid Layout est une spécification du W3C à l'état de Candidate Recommendation dont les premiers jets datent de 2004. Désormais exploitable en production sur l'ensemble des navigateurs, le concept général de Grid Layout (ou « positionnement en grille ») est de structurer l'espace en zones majeures, au sein d'une page web ou d'une application. Une grille virtuelle, indépendante des éléments HTML qui peuvent s'y placer, présente l'avantage flagrant d'être très souple lorsqu'on souhaite la modifier en fonction du contexte.

Parfaitement adapté au responsive webdesign, Grid Layout permet à votre page de s'adapter immédiatement aux différentes tailles d'écran. Associé aux CSS 3 Media Queries, le modèle de positionnement en grille donne toute sa puissance, plus particulièrement si vous avez opté pour une trame sous forme de variante « template ». Seul le patron de départ nécessite d'être modifié : inutile d'intervenir sur la structure HTML, ni même sur le positionnement de chaque élément de page.

Agrémenté de nombreuses illustrations en couleurs et d'exercices pratiques, cet ouvrage vous accompagne de façon progressive dans la découverte des concepts et propriétés associés à ce nouveau module et vous permet d'élaborer efficacement l'architecture de vos pages sous forme de grilles.

À qui s'adresse cet ouvrage ?

- Aux webdesigners et intégrateurs avancés ou experts qui souhaitent appréhender et maîtriser en production ce nouveau modèle de positionnement excitant.
- Aux développeurs et chefs de projet Web qui s'interrogent sur les possibilités offertes par CSS en termes de design d'interfaces.

Au sommaire

Les concepts de grille • Les propriétés de grille • TP : notre premier gabarit • Grille explicite et grille implicite • TP : un gabarit explicite • Le flux des éléments • Positionnement via repères nommés • TP : un effet de calque superposé • Unités et valeurs de grille • Gérer les gouttières • Modifier l'ordre d'affichage • TP : trier des ressources • Aligner dans la grille • Fusionner des colonnes et des rangées • TP : une grille simple monoligne • TP : Grid Layout et responsive webdesign • TP : une galerie d'images parfaite • Compatibilité et amélioration progressive • Annexe - Mémo des propriétés

Webdesigner et gérant d'une agence web strasbourgeoise, **Raphaël Goetter** partage ses connaissances à travers son site Alsacreations.com, et s'intéresse de près aux domaines des normes du Web et de l'accessibilité. Expert et formateur en langages HTML et CSS et auteur de livres techniques chez Eyrolles.

www.editions-eyrolles.com

CSS 3 Grid Layout

DANS LA MÊME COLLECTION

C. BLAESS. – **Solutions temps réel sous Linux.**

N° 67711, 3^e édition, 2019, 320 pages.

C. PIERRE DE GEYER, J. PAULI, P. MARTIN, E. DASPET. – **PHP 7 avancé.**

N° 67720, 2^e édition, 2018, 736 pages.

H. WICKHAM, G. GROLEMUND. – **R pour les data sciences.**

N° 67571, 2018, 496 pages.

F. PROVOST, T. FAWCETT. – **Data science pour l'entreprise.**

N° 67570, 2018, 370 pages.

J. CHOKOGOUE. – **Maîtrisez l'utilisation des technologies Hadoop.**

N° 67478, 2018, 432 pages.

H. BEN REBAH, B. MARIAT. – **API HTML 5 : maîtrisez le Web moderne !**

N° 67554, 2018, 294 pages.

W. MCKINNEY. – **Analyse de données en Python.**

N° 14109, 2015, 488 pages.

E. BIERNAT, M. LUTZ. – **Data science : fondamentaux et études de cas.**

N° 14243, 2015, 312 pages.

SUR LE MÊME THÈME

F. DRAILLARD. – **Premiers pas en CSS 3 et HTML 5.**

N°67753, 8^e édition, 2019, 496 pages.

R. GOETTER. – **Mémento CSS 3.**

N° 67432, 4^e édition, 2017, 14 pages.

D. CEDERHOLM – **CSS 3 pour les web designers**

N° 11765, 2016, 152 pages.

R. GOETTER. – **CSS 3 Flexbox.**

N° 14363, 2016, 152 pages.

H. GIRADEL, R. GOETTER. – **CSS 3 : pratique du design web.**

N° 14023, 2015, 372 pages.

K. DELOUMEAU-PRIGENT. – **CSS maintenables avec Sass et Compass.**

N° 13640, 2^e édition, 2014, 238 pages.

M. KABAB, R. GOETTER. – **Sass et Compass avancé.**

N° 13677, 2013, 280 pages.

Retrouvez nos bundles (livres papier + e-book) et livres numériques sur

<http://izibook.eyrolles.com>

Raphaël Goetter

CSS 3 Grid Layout

Vous allez enfin aimer CSS

● Éditions
EYROLLES

ÉDITIONS EYROLLES
61, bd Saint-Germain
75240 Paris Cedex 05
www.editions-eyrolles.com

En application de la loi du 11 mars 1957, il est interdit de reproduire intégralement ou partiellement le présent ouvrage, sur quelque support que ce soit, sans l'autorisation de l'Éditeur ou du Centre Français d'exploitation du droit de copie, 20, rue des Grands Augustins, 75006 Paris.

© Éditions Eyrolles, 2019, ISBN : 978-2-212-67683-9

Avant propos

« Grid Layout » est la spécification CSS du W3C officiellement déclarée « Révolution CSS la plus attendue de tous les temps » par les développeurs web.

Mon ouvrage précédent fut un éloge au modèle de positionnement CSS Flexbox, alors première amorce d'une complète refonte du modèle de positionnement CSS depuis trop longtemps engourdi, voire enlaidi, de détournements de propriétés et approximations en tout genre.

Avec l'avènement de Grid Layout, aujourd'hui exploitable en production sur l'ensemble des navigateurs modernes, un second palier de taille est franchi. Le couple Flexbox-Grid offre une continuité et une complémentarité exemplaires dans la construction de pages et d'applications web. À tel point que toutes les autres techniques de positionnement s'en trouvent reléguées à la préhistoire.

Les possibilités accordées par Grid Layout à nous autres webdesigners sont sans conteste à mille lieues de ce que l'on était capable de produire auparavant.

Le prix à payer est que les spécifications du modèle Grid Layout comptent également parmi les plus fournies et aussi les plus indigestes du langage CSS.

Cet ouvrage, consacré au modèle Grid Layout, se veut pratique et complémentaire de celui dédié à Flexbox, qui était paru en 2016. L'un et l'autre vont explorer différentes méthodes de conception de gabarits et de composants web à l'aide de cas d'usages très concrets, rencontrés lors de nos intégrations dans l'agence web Alsacrétions.

À qui s'adresse ce livre ?

Ce livre est destiné à toute une génération de pionniers du Web et de l'intégration HTML et CSS ; à toutes celles et ceux qui ont eu l'aubaine de connaître nos glorieux ancêtres Internet Explorer 6, pour ne pas oser citer Netscape, de côtoyer les incohérences d'affichage quotidiennes ; à quiconque ayant un jour pesté contre CSS parce que « ça ne marche jamais comme l'on voudrait ».

Vous qui êtes webdesigner ou développeur, et qui avez longtemps été limité dans votre travail, les spécifications du W3C vous offrent désormais de véritables solutions stables et pérennes sous la forme des modules Flexbox et Grid Layout.

Vous êtes les premières cibles de cet ouvrage, témoins d'une époque de bidouille révolue et précurseurs d'une renaissance annoncée du webdesign.

Qu'allez-vous y trouver ?

Entièrement dédié à un seul module de CSS, ce livre a pour mission de vous faire découvrir le phénomène incontournable qu'est devenu Grid Layout. Vous l'appréhenderez de façon théorique, mais aussi et surtout vous découvrirez comment le mettre en action au sein de projets concrets et pratiques.

Composants essentiels de ce manuel, plusieurs travaux pratiques décortiqués sont proposés pour soulager votre quotidien de webdesigner : créer des gabarits de page web évidemment, mais également des effets de calque superposé, des tris de ressources, une grille de mise en forme et une galerie d'images comme on les aime.

Site web associé

Pour accompagner cet ouvrage, une page de support en ligne est consultable à l'adresse <http://goetter.fr/livres/gridlayout/>.

Vous trouverez sur ce mini-site web diverses informations utiles durant votre lecture :

- les codes sources des différents exercices et travaux pratiques de l'ouvrage ;
- quelques illustrations ou extraits textuels du livre ;
- les éventuelles erreurs qui auraient échappé à ma relecture attentive ou à celle de mon éditeur.

Bonne lecture,

Raphaël,
Alsacrérations

Table des matières

CHAPITRE 1

Introduction	1
Qu'apporte Grid Layout par rapport aux autres ?	2
Meilleur que Flexbox ou Bootstrap ?	3
Un bref historique de Grid Layout	4
La préhistoire de Grid Layout	4
L'arrivée des frameworks de grille	4
Les premières implémentations de Grid Layout	5
Le coup de boost Bloomberg	5

CHAPITRE 2

Les concepts de grille	7
Les éléments physiques	8
Grid Container	8
Grid Item	8
Les éléments logiques	9
Grid Line	9
Grid Track	9
Grid Cell	9
Grid Area	9
En image	10

CHAPITRE 3

Les propriétés de grille	11
Les propriétés qui s'appliquent sur le parent	11
Propriétés d'initialisation de grille	11
<i>display: grid.</i>	12
<i>display: inline-grid.</i>	12
<i>display: subgrid.</i>	12
Propriétés de déclaration des colonnes et rangées	12
<i>grid-template-rows.</i>	12
<i>grid-template-columns.</i>	12
<i>grid-template-areas.</i>	12
Propriétés liées au flux des éléments	12
<i>grid-auto-rows.</i>	12
<i>grid-auto-columns.</i>	12
<i>grid-auto-flow.</i>	13

Propriétés liées aux gouttières	13
<i>row-gap</i>	13
<i>column-gap</i>	13
Propriétés d'alignement général	13
<i>justify-content</i>	13
<i>justify-items</i>	13
<i>align-content</i>	13
<i>align-items</i>	13
Des raccourcis	13
Les propriétés qui s'appliquent aux enfants	14
Propriétés de placement dans la grille	14
<i>grid-row-start</i>	14
<i>grid-row-end</i>	14
<i>grid-column-start</i>	14
<i>grid-column-end</i>	14
<i>grid-area</i>	15
Propriétés d'alignement individuel	15
<i>justify-self</i>	15
<i>align-self</i>	15
Propriété d'ordonnancement	15
<i>order</i>	15

TP 1

Notre premier gabarit	17
La technique Float	18
La technique Flexbox	19
La technique Grid Layout	20
Ajouter des gouttières	21
Changer l'ordre d'affichage	22

CHAPITRE 4

Grille explicite et grille implicite	25
Une grille explicite	25
Créer et dimensionner des pistes explicites	25
Représentation d'une grille explicite	27
Positionnement explicite des éléments	28
<i>Lignes négatives</i>	29
<i>Variantes de positionnement</i>	29
<i>Quelques exemples concrets</i>	30
Une grille implicite	33
Prévoir et dimensionner des pistes implicites	34
Une question de taille	35

TP 2

Un gabarit explicite	37
---------------------------------------	-----------

CHAPITRE 5

Le flux des éléments 41

La distribution des éléments	41
Distribution par défaut	41
Changer l'axe de distribution	42
Regroupement avec la valeur dense	43
Gérer le flux des éléments	44
Placement automatique (auto-placement)	45
Placement défini (definite position)	45
Placement semi-défini	45
Algorithme de placement dans la grille	45
Mélanger les placements	46
Un flux complexe	46

CHAPITRE 6

Positionnement via repères nommés 49

Lignes nommées	50
Zones nommées	51
Fusion de zones	52
Préfixes -start et -end	53
Et la version raccourcie	54
Équivalence entre les propriétés de placement	55

TP 3

Un effet de calque superposé..... 57

Les zones nommées à la rescousse	58
Le choix du placement hybride	59
Un calque superposé	59
Positionnement du calque	60
Variantes de positionnements	61
Un calque toujours au-dessus	61
Récapitulatif	62

CHAPITRE 7

Unités et valeurs de grille..... 63

L'unité fr	63
Les valeurs-fonctions	65
minmax()	65
repeat()	66
<i>Des nombres et des motifs.</i>	66
<i>Nombre de colonnes variables</i>	67

CHAPITRE 8

Gérer les gouttières..... 69

Gouttières horizontales et verticales	70
Anciennes et nouvelles spécifications	71
Tenir compte de la gouttière dans la taille	71

CHAPITRE 9

Modifier l'ordre d'affichage..... 73

Placer explicitement	74
Réordonner avec order	75
Valeurs de la propriété order	75
Comment fonctionne order ?	75
Placement explicite ou order ?	76

TP 4

Trier des ressources..... 77

Distinguer visuellement les liens	78
Créer la grille et placer les éléments	78
Remplir les emplacements libres	79

CHAPITRE 10

Aligner dans la grille..... 81

Aligner les cellules de la grille	82
Aligner les grid items	83
Aligner un grid item particulier	84
En résumé	85

CHAPITRE 11

Fusionner des colonnes et des rangées..... 87

Fusion avec start et end	87
Fusion avec span	88
span opère dans les deux sens	89
Les pistes implicites fusionnent aussi	90
Fusion avec zones nommées	91

TP 5

Une grille simple monoligne..... 93

Méthode minmax()	94
Méthode grid-auto-flow	95

TP 6

Grid Layout et responsive webdesign..... 97

Structure HTML	99
Stratégie responsive	99
Première étape : casser le tableau !	100

Deuxième étape : transformer en zones nommées	101
Troisième étape : il n'y a plus rien à faire !	103

TP 7

Une galerie d'images parfaite..... 105

Objectif à atteindre	105
Structure HTML	106
Construction de la grille	107
La petite touche responsive	107
Du responsive sans Media Queries	108
Gestion des formats d'images	108
Dimensionnement des images	108
Portrait ou paysage ?	109
Remplir les brèches	109
Conserver le ratio : object-fit	110
Code HTML final	111
Code CSS final	111
Le niveau supérieur : l'automatisation	112

CHAPITRE 12

Compatibilité et amélioration progressive..... 113

Une compatibilité acceptable	113
Un support partiel d'Internet Explorer	114
Une question de version	114
Fonctionnalités manquantes	114
<i>Pas de placement automatique</i>	114
<i>Pas de gouttières</i>	115
<i>Pas de zones nommées</i>	115
<i>Autres manques</i>	115
Des fonctionnalités divergentes	116
<i>Des propriétés différentes... et préfixées</i>	116
<i>repeat()</i>	117
<i>Des bogues</i>	117
Alors, on fait quoi concrètement ?	117
<i>Autoprefixer</i>	117
<i>@supports()</i>	119
Quelles sont mes options ?	119
<i>Enfin libre ! (Edge minimum)</i>	119
<i>Tenir compte d'IE10 et IE11</i>	120
<i>Tenir compte d'IE9</i>	120
Conclusion	120

ANNEXE A

Ressources..... 121

Tutoriels et documentation	121
Blogs de référence	121
Conférences à retenir	122
Ludique pour expérimenter	122

Outils	122
Démos	122
Articles notables	122
Sites web en Grid Layout	123
ANNEXE B	
Mémo des propriétés	125
Propriétés de Grid Layout	125
Propriétés d'alignement	126
Index	127

1

Introduction

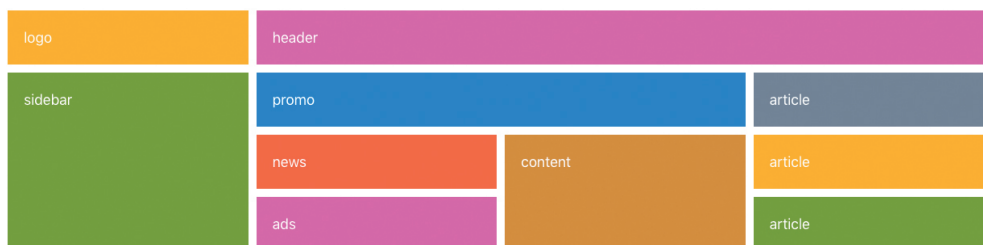


Figure 1-1 Exemple de construction qui ne vous effrayera plus après avoir parcouru cet ouvrage !

Je ne sais pas pour vous, mais pour ma part je me suis toujours senti profondément limité par CSS et ses propriétés de positionnement des éléments au sein d'une page web. J'ai l'intime conviction que je ne suis pas seul à m'en plaindre.

Pour avoir longtemps subi les `frameset`, `<table>`, `position absolute` et `float` à toutes les sauces et à grand renfort de `clearfix` et bidouilles multiples, j'ai versé ma petite larme d'émotion lorsque j'ai parcouru pour la première fois les spécifications Flexbox en 2011.

Le modèle de boîte flexible posait enfin les bases d'une nouvelle ère de positionnement en CSS, adaptée aux besoins et usages modernes. Je fus immédiatement conquis et devins rapidement un ambassadeur Flexbox, en prêchant à tout va la bonne parole, lors des conférences ou des formations que je dispensais.

L'arrivée de Grid Layout plante un second jalon dans le processus de renouvellement des techniques CSS. Flexbox est parfait pour gérer les composants, les contenus et les éléments internes dans la page, cependant la construction globale des gabarits, les grilles de mise en

forme et leurs gouttières demeurent problématiques sur certains aspects ou liés à certains contextes d'affichage ou de périphériques.

Grid Layout propose un modèle de positionnement qui remplit à la perfection sa mission : concevoir des pages web plus ou moins complexes de manière simple, intuitive et sans code parasite ou inutile. Tout est dit.

La complémentarité entre Flexbox et Grid Layout est aussi parfaite que délectable (les spécifications sont d'ailleurs rédigées par les mêmes éditeurs actuellement). À tel point qu'on reprendrait presque goût à pratiquer du CSS, et à remanier un à un tous ses anciens projets web codés à l'ancienne pour les améliorer. En tout cas, c'est clairement mon cas !

Les spécifications du module Grid Layout sont aujourd'hui considérées comme stables et l'ensemble des navigateurs récents les prennent en charge depuis déjà un petit moment.

La documentation officielle de « CSS Grid Layout Module Level 1 », c'est son petit nom technique, peut être consultée, si vous êtes courageux à l'adresse suivante : <https://www.w3.org/TR/css-grid-1/>.

Par ailleurs, sachez que la spécification continue d'évoluer puisque le « Level 2 » est déjà disponible à l'état de brouillon avancé (« Working Draft ») à cette adresse : <https://www.w3.org/TR/css-grid-2/>.

REMARQUE Juste une mise au point

Ce livre ne couvre volontairement pas les nombreuses subtilités des très vastes spécifications W3C de ce module. Il ne constitue pas une bible exhaustive sur le thème de Grid Layout, mais il se veut plutôt un guide pratique rempli de multiples exemples, codes et travaux pratiques (TP).

Dans cette optique, certaines fonctionnalités, propriétés ou valeurs ont été écartées de cet ouvrage. Référez-vous à la documentation officielle si vous êtes en quête d'exhaustivité sur un thème précis. Sachez être indulgent si telle ou telle fonctionnalité, que j'aurais jugée peu indispensable, en fut écartée.

Qu'apporte Grid Layout par rapport aux autres ?

Grid Layout est un véritable framework de positionnement à lui tout seul, conçu pour agencer vos pages web.

Ça a l'air trivial dit comme ça, mais saviez-vous que toutes les techniques CSS employées jusqu'alors pour positionner les blocs principaux d'un document relèvent de l'imposture ? Ni `position`, ni `float`, ni `inline-block`, ni même `Flexbox` (oui je mets une majuscule à `Flexbox`) n'ont été élaborés pour créer des designs de page. Nous les utilisons en les détournant de leur objectif initial.

Quelle importance, me direz-vous ?

C'est vrai, après tout, on a toujours fait comme ça et ça a toujours fonctionné bien avant Grid Layout.

Si vous me demandiez quels sont les réels apports de Grid Layout, j'avancerais sans aucun doute les arguments suivants :

- **Parce que c'est fait pour ça.** Voilà un argument irréfutable, Grid Layout a été forgé sur la base de toutes les ancestrales bidouilles CSS dans un unique but : organiser la composition générale des sections de la page.
- **Parce que c'est plus rapide à écrire.** Quelques lignes de code suffisent à produire un affichage élaboré, moderne et responsive.
- **Pour des raisons de performances.** À quelques millisecondes près, un gabarit réalisé avec Grid Layout s'affiche plus rapidement sur votre écran que son équivalent conçu avec d'autres techniques. La première analyse avait été réalisée par Igalia en 2015, mais depuis, la capacité des navigateurs à implémenter Grid Layout s'est encore accrue.
- **Parce que c'est plus facile à maintenir.** Les styles utilisés dans le modèle Grid Layout sont totalement indépendants de la structure HTML. De ce fait, il est possible de remodeler intégralement votre visuel sans même devoir toucher à sa structure... et inversement.

Meilleur que Flexbox ou Bootstrap ?

Mettons les choses au clair afin de partir sur de bonnes bases : Grid Layout, Flexbox et Bootstrap sont à la fois différents et complémentaires.

Pour commencer, Grid Layout n'est pas tout à fait une révolution. La véritable révolution a été initiée par Flexbox il y a plusieurs années déjà. Grid Layout s'inscrit dans la continuité de Flexbox, et devient aujourd'hui son compagnon de tous les jours, son meilleur allié.

Il existe une différence fondamentale entre Grid Layout et Flexbox : **la gestion des deux axes horizontaux et verticaux de façon simultanée**. Flexbox n'est optimal que dans une seule direction, tandis que Grid Layout gère parfaitement les rangées tout autant que les colonnes sans artifice ni conteneurs intermédiaires.

L'autre différence réside dans leur nature même : Grid Layout a été conçu pour une prise en compte globale de la page, tandis que Flexbox est prévu pour agencer les composants internes et gérer leur fluidité.

REMARQUE Grid Layout ne remplace pas Flexbox

Face à l'incompréhension d'un bon nombre de développeurs, le site CSS-Tricks a publié un article (en anglais) expliquant les usages de ces deux technologies : <https://css-tricks.com/css-grid-replace-flexbox/>

Pour ce qui est de Bootstrap, il est encore plus improbable de faire l'amalgame. En effet, Bootstrap est un framework composé de toutes pièces et mélangeant allègrement diverses techniques CSS (et JavaScript), tandis que Grid Layout est une spécification officielle du W3C.

Bootstrap est un outil polyvalent, intégrant une convention de nommage, un fichier CSS « reset » des composants JavaScript et un modèle de positionnement par grille. C'est bien sûr cette dernière fonctionnalité qui prête à confusion aujourd'hui lorsqu'on la confronte à Grid Layout.