

Départ immédiat pour :

Flexbox

Thomas Zilliox

 le **train** de **13H37**
PUBLICATIONS POUR
LA CONCEPTION WEB

Départ immédiat pour :

Flexbox

Thomas Zilliox

Publié par Les Contrôleurs du train de 13h37

<http://letrainde13h37.fr>

ISBN : 979-10-91997-08-9

Wagon 42 SAS

La Gare

42380 Périgueux

France

<http://wagon42.fr>

Première édition : v1.0 le 11 juillet 2017

Édition en cours : v1.1 le 21 janvier 2018

Si vous trouvez une erreur qui nous aurait échappé, n'hésitez pas à nous la reporter sur <http://t13h37.fr/feedback-flexbox> ou par mail à support@letrainde13h37.fr.

À propos du train de 13h37

Le train de 13h37 est un éditeur indépendant qui publie des articles et des livres autour de la conception web. En tant que travailleurs du web nous-mêmes, notre volonté à travers *le train de 13h37* est de participer à l'amélioration de la vie de nos pairs en leur proposant des ressources pratiques et utiles.

Déjà parus

- *La dette technique*
<http://t13h37.fr/livre-dette-tech>,
par Bastien Jaillot ;
- *Design d'icônes : le manuel*
<http://t13h37.fr/livre-icônes>,
par Sébastien Desbenoit.
- *Petit Précis de Créativité*
<http://t13h37.fr/livre-creativite>,
par Virginie Caplet.

Remerciements de l'auteur

Un grand merci à toute l'équipe du *train de 13h37*, et plus spécifiquement à Marie, pour leur patience. Le livre que vous tenez dans vos mains est le fruit de plusieurs mois de réflexion et n'aurait pas vu le jour sans leur confiance.

Merci à ma compagne, qui m'a soutenu, qui a relu des dizaines de brouillons, et qui est désormais très à l'aise avec Flexbox !

Merci à Simon Alberny et Romaric Pascal d'avoir été des relecteurs passionnés et intransigeants.

Enfin tout spécialement merci à Raphaël Goetter, pour avoir écrit autant d'articles et donné autant de conférences sur le sujet. Je lui dois, comme tant d'autres, une bonne partie de mes connaissances.

Table des matières

Flexbox par l'exemple _____ 15

Le support de Flexbox _____ 29

Le référentiel des propriétés __ 35

Vous avez en main votre billet pour découvrir Flexbox, le premier standard CSS dédié à la mise en page ! Il simplifie grandement l'alignement des éléments et améliore ainsi le confort d'écriture et de lecture des gabarits des pages web ; il permet aussi de créer des structures complexes qu'il nous était impossible de réaliser par le passé. Pour pouvoir proposer toutes ces fonctionnalités, Flexbox a dû imaginer une nouvelle logique d'affichage. Ce livre va vous accompagner dans sa découverte.

Le but est de vous rendre autonome le plus rapidement possible. Dans la première partie de notre voyage, nous allons nous concentrer sur un lot de fonctionnalités réduites. À l'aide d'exemples successifs, vous apprendrez à utiliser ensuite les règles, celles qui vous seront nécessaires pour pouvoir commencer à utiliser Flexbox dans vos prochaines réalisations.

Nous vous aiderons aussi à sauter le pas en répondant aux questions récurrentes : Quel est le support navigateur actuel ? Combien de syntaxes différentes dois-je maintenir ? Comment faire si mon projet doit aussi supporter IE8 ?

Pour aller plus loin, la dernière partie de notre périple sera marquée par une courte description de toutes les nouvelles propriétés CSS. Sans se plonger dans les détails, connaître l'étendue des fonctionnalités de ce standard et son vocabulaire vous sera essentiel pour savoir quand l'utiliser et pour pouvoir approfondir par vous-même.

Alors, installez-vous confortablement, nous sommes prêts à partir !

Flexbox par l'exemple

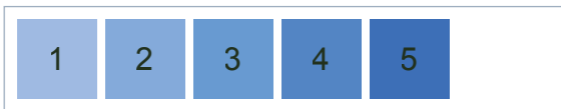
Assez régulièrement, Flexbox est utilisé pour aligner des éléments horizontalement. Sans Flexbox, des éléments `inline-block` ou des éléments flottants auraient été utilisés pour ce besoin-là. Partons donc d'un de ces cas pour notre premier essai :

```
<!-- Extrait du HTML de notre démo -->
```

```
<div class="parent">  
  <div class="child">1</div>  
  <div class="child">2</div>  
  <div class="child">3</div>  
  <div class="child">4</div>  
  <div class="child">5</div>  
</div>
```

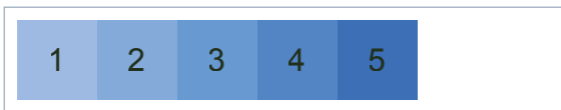
```
/* Extrait des CSS associées */
```

```
.parent {}  
.child { display: inline-block; width: 100px;  
height: 100px; }
```

En ajoutant `display: flex` à l'élément parent, il va devenir un **conteneur flexible**. Flexbox diffère beaucoup des autres modes d'affichage, car il va forcer le mode d'affichage de ses enfants directs. Les enfants d'un conteneur flexible deviennent des **éléments flexibles**. Ils le resteront quelle que soit la valeur de la propriété `display` qui leur est appliquée.

```
.parent { display: flex; }  
.child { width: 100px; height: 100px; }
```



Voilà ! C'est notre première mise en page flexible ! Simple, non ?

Le rendu est finalement assez proche de celui obtenu avec des éléments `inline-block`. Comme ceux-ci, les éléments flexibles s'adaptent par défaut aux dimensions de leur contenu. Notons quand même qu'il n'y a pas de problème d'espaces indésirables entre les éléments flexibles. Il n'y a

pas non plus de problème d'éléments qui sortent du flux, comme avec une mise en page à l'aide d'éléments flottants.

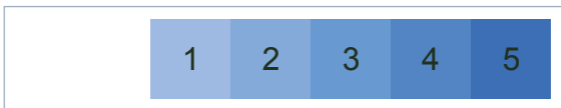
1.1 La répartition des espaces vides

Jouons maintenant avec notre première propriété spécifique à Flexbox : `justify-content`. Elle doit être appliquée sur le conteneur flexible pour indiquer la répartition des espaces vides. On va pouvoir ainsi aligner les éléments au centre ou sur un côté.

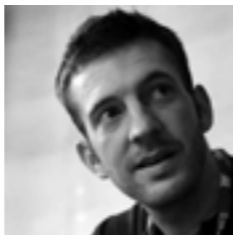
```
.parent { display: flex; justify-content: center; }  
.child { width: 100px; height: 100px; }
```



```
.parent { display: flex; justify-content: flex-end; }  
.child { width: 100px; height: 100px; }
```



À propos de l'auteur



Thomas est aujourd'hui un expert CSS. Avant ça, il a été développeur web sur des projets variés, du Java au PHP, du framework au tout fait maison, du site vitrine au webmail avec dix millions d'utilisateurs. À chaque fois ce fut le même constat : le bug tracker était rempli à 70 % de problèmes d'intégration et ne se vidait jamais. Bien sûr, l'arrivée du responsive n'a pas aidé.

Alors depuis 2011, Thomas a pour but d'aider les équipes de développement à produire des CSS maintenables. Il alterne le rôle de consultant, de formateur et d'auteur pour faire découvrir les bonnes pratiques, les méthodologies et le fonctionnement des CSS.

Plein d'idéaux, il a contribué au projet Piwik afin d'encourager le respect de la vie privée. Il intervient à l'Université de Lyon et régulièrement lors de conférences web comme Paris Web, Sud Web, Kiwi Party ou Blend Web Mix...

Colophon

Ce livre a été voulu comme un objet agréable à lire et à manipuler.

La police utilisée pour le corps du livre est la Gandhi Serif. Celle destinée aux titres est la Amaranth.

Les Contrôleurs Marie Alhomme et Loïc Mathaud espèrent que vous avez effectué un agréable voyage.

