
Tutoriel Marp : MarkDown vers slides

Ce document explique comment installer et utiliser l'outil Marp, dans le but de créer une présentation à partir d'un fichier texte Markdown.

Il a été écrit par les élèves de la formation Ingénierie des low-tech à l'Ecole Centrale de Nantes en Octobre 2023

Version	Mise à jour	License
v1.0	14/10/2023	Creative Commons Zero

Table des matières

Comment créer une présentation à partir de Markdown via Marp	3
Qu'est-ce que Marp ?	3
Spécificités de Marp	3
Installation de Marp for VS Code	3
Création d'une présentation avec Marp for VS Code	3
Paramètres du fichier	3
Directives globales	4
Directives locales	4
Transitions inter-slides	5
Animations	5
Exporter la présentation	6
Images	6
Annexes	7
Arborescence conseillée	7
Modèle	7

Comment créer une présentation à partir de Markdown via Marp

Ce tuto présente une manière simple de créer des slides en Markdown (et un peu de CSS) avec Marp.

En annexe de ce tuto se trouve un modèle de document, aux directives globales déjà écrites, que vous pouvez réutiliser.

Qu'est-ce que Marp ?

Marp est une collection d'outils open source : Marpit, Marp Core, Marp CLI, Marp for VS Code.

En tant qu'utilisateur, on ne s'intéresse qu'à Marp CLI ou Marp for VS Code.

- Marp CLI : Génération des slides en lignes de commandes (dans un terminal). Intéressant pour automatiser la génération, mais moins évident visuellement.
- Marp for VS Code : Extension à ajouter à Microsoft Visual Studio Code (logiciel gratuit mais non open source). Permet de combiner un éditeur de texte pratique, une fenêtre de prévisualisation en direct, et un outil d'exportation.

Pour sa simplicité d'utilisation, ce tuto utilise Marp for VS Code.

Spécificités de Marp

Marp utilise Markdown et ses balises prédéfinies en le complétant par :

- la balise `---` pour indiquer une **nouvelle slide** (*et non pas une ligne horizontale*)
- la commande `marp: true` pour indiquer que ce n'est pas du Markdown classique mais du Marp
- des styles et thèmes en CSS, personnalisables
- un logiciel pour transformer le Markdown en différents formats (HTML, PDF)

Installation de Marp for VS Code

1. Installer VS Code
2. Ouvrir le menu des extensions (accessible à gauche ou avec Ctrl+Maj+X) et y chercher `marp-team.marp-vscode`
3. Cliquer sur le résultat "Marp for VS Code" puis sur le bouton d'installation.

Création d'une présentation avec Marp for VS Code

Créer un fichier Markdown (`.md`) et l'ouvrir avec VS Code.

Vous pouvez utiliser le modèle en annexe de ce tuto.

Paramètres du fichier

Un fichier Markdown pour une présentation Marp commence par un certain nombre d'instructions qui paramètrent le style de la présentation.

Une fois ces paramètres personnalisés (comme dans le modèle en annexe), vous pourrez simplement les copier-coller dans vos autres présentations pour garder le même style !

Directives globales

Elles se situent au tout début du document Markdown, entre deux balises `---` (qui ne sont exceptionnellement pas interprétées comme une ligne horizontale ou un changement de slide).

Voici un exemple de directives globales mises en début de document :

```
---
marp: true
theme: gaia
markdown.marp.enableHtml: true
paginate: true
style: |
  section {
    background-color: #fefefe;
    color: #333;
    font-size: 23px
  }
---
```

- `marp: true` sert à informer VS Code que l'on va utiliser du Markdown pour Marp. C'est utile pour que la prévisualisation se fasse bien, utilisant donc l'extension Marp for VS Code.
- `theme: gaia` définit le thème général de la présentation. Vous avez le choix parmi les trois thèmes prédéfinis par Marp : `default`, `gaia` et `uncover`. Le thème peut aussi être customisé mais j'avoue ne pas maîtriser la question est la documentation n'est pas bien claire ni explicite.
- `markdown.marp.enableHtml: true` permet la reconnaissance de balises HTML au sein du Markdown. On pourrait en avoir besoin pour réaliser quelques petites folies non supportées par Markdown, comme insérer une vidéo dans la présentation avec la balise HTML `<video>` : `html` `<div id=video margin="auto">` `<video controls width="40%" src="img/video_name.mp4"></video>` `</div>`
- `paginate: true` indique qu'on veut numéroter les slides. Cela pourrait être une directive locale.
- `style: |` et le texte indenté qui suit donne des instructions de style en CSS à appliquer au thème. Ces instructions sont valables sur l'ensemble du document mais pourront être adaptées avec des directives locales.

On pourrait ajouter à cette liste la directive `headingDivider: 2` pour demander de changer automatiquement de slide lorsqu'un titre est de niveau 2 ou supérieur (donc 1 ou 2), le chiffre 2 étant bien sûr un exemple arbitraire.

Pour rappel, le changement de slide se fait sinon en écrivant la balise `---`.

Directives locales

Elles se situent au début d'une slide entre la balise ouvrante `<!--` et la balise fermante `-->` et s'appliquent à la slide en cours et aux suivantes.

Une directive locale peut s'appliquer à seulement la slide en cours si son nom est précédé de `_` (underscore).

Exemple :

```
---
<!-- backgroundColor: black -->
```

Slide avec fond noir, fond appliqué ici et à tout ce qui suit.

```
<!-- _backgroundColor: white -->
```

Slide en fond blanc, fond appliqué seulement ici.

Slide en fond noir à nouveau, comme pour tout ce qui suivra.

Ce sont des paramètres qui auraient éventuellement pu être définis en directives globales.
Voici une liste non exhaustive de directives locales :

- **paginate** pour la numérotation des pages
Valeurs possibles : **true**, **false**, **hold**, **skip**
- **header** et **footer** pour l'en-tête et le pied de page des slides
Valeur textuelle entre ' ou "
- **class** pour changer l'attribut de classe de l'élément **<section>** (de la slide). Les classes peuvent être définies en CSS dans la directive globale **style**.
- **backgroundColor**, **backgroundImage**, **backgroundPosition**, **backgroundRepeat**, **backgroundSize** pour paramétrer l'arrière-plan. Ce sont des dérivées des options CSS **background-color** etc dont les valeurs possibles sont les mêmes.
- **color** pour la couleur du texte, idem même valeurs possibles que son équivalent CSS.

Pour plus de personnalisation, notamment au sein de la page, il sera généralement nécessaire de connaître un minimum les langages CSS (pour créer des classes en directive globale) et HTML (pour utiliser des balises plus poussées que celles de Markdown, d'où la directive globale **markdown.marp.enableHtml: true**).

On utilisera par exemple en début de slidee, pour ne pas avoir à définir une nouvelle classe en directive globale, le code HTML :

```
<style scoped>
h1 {
  color: blue;
}
</style>
```

Cependant, on peut faire déjà énormément avec Markdown et ces quelques directives globales et locales.

Transitions inter-slides

Une transition est un cas spécifique de directive locale. Marp inclut 33 transitions prédéfinies.
Exemple :

```
<!-- transition: melt -->
```

La transition est un effet qui s'applique à la fin de la slide actuelle et non au début, donc entre la slide actuelle et la suivante et non pour arriver à la slide actuelle.

Animations

On peut vouloir faire apparaître les éléments un par un à chaque clic de souris.
Deux options existent :

- pour faire apparaître progressivement des éléments de listes (on parle alors de listes fragmentées) :
 - utiliser `*` au lieu de `-` pour les listes non numérotées
 - utiliser `1)` au lieu de `1.` pour les listes numérotées
- pour fragmenter l'apparition d'autres éléments, Markdown ne suffit pas, il faut placer chaque élément dans une section HTML `<div>` avec l'option `data-marpit-fragment` :

Texte qui apparaît directement.

```
<div data-marpit-fragment>  
Texte qui apparaît après un clic de souris.  
</div>
```

Exporter la présentation

L'extension Marp for VS Code ajoute un bouton en haut à droite de la fenêtre d'édition, avec le logo de Marp. Ce bouton permet d'accéder à certaines commandes de Marp et aux paramètres de l'extension.

Quand on clique dessus on a notamment la possibilité d'exporter la présentation (le *slide deck*). Dans la fenêtre qui s'ouvre, vous pourrez choisir le format voulu.

Pour modifier le format d'exportation par défaut, allez dans les paramètres de l'extension.

Images

Pour intégrer des images (ou vidéos) à la présentation, vous pouvez consulter la documentation.

Il est fortement conseillé de créer un dossier contenant votre projet, c'est-à-dire regroupant votre fichier Markdown et un dossier `img` qui contiendra les images et vidéos utilisées. Les liens vers vos images, si elles ne sont pas récupérées en ligne, ressembleront donc à `img/image.png`. Notez qu'en exportant la présentation sous certains formats, les images et vidéos sont toujours récupérées dans le dossier `img` à la lecture et non pas enregistrées dans le fichier de présentation. Il faut en ce cas conserver le dossier `img` au même endroit que la présentation.

Pour résumer le principal de la documentation :

- La syntaxe `` de Markdown est toujours valable, mais a été complétée.
- On peut ajouter plusieurs éléments entre les crochets :
 - on peut paramétrer la taille d'une image avec les options `width` et `height` d'unités possibles `cm` `mm` ou `px` (pixels, unité par défaut si on n'en écrit pas).
Exemple :

```
![width:200px](img/image.jpg)  
![height:30cm](img/image.jpg)  
![width:200 height:300mm](img/image.jpg)
```
 - pour chaque image on peut paramétrer des filtres (flou `blur`, luminosité `brightness`, contraste `contrast`, ombre projetée `drop-shadow`, niveau de gris `grayscale`, inversion des couleurs `invert`, opacité `opacity`, saturation `saturate`, sepia `sepia`) et les combiner.
Exemple :

```
![width:200 brightness:.8 sepia:50%](img/image.jpg)
```

- pour mettre une image en arrière-plan on met le paramètre `bg` entre les crochets.
On peut toujours spécifier la taille de l'image avec `width` et `height`, mais on peut maintenant le faire aussi avec :

- * `contain` ou `fit` pour que l'image rentre entièrement
- * `cover` (par défaut) pour remplir l'espace, possiblement en tronquant l'image
- * `auto` pour conserver la taille originale de l'image
- * `90%` ou autre chiffre pour appliquer un facteur de redimensionnement

On peut également partager l'espace (par défaut horizontalement, verticalement avec le paramètre `vertical`) avec les paramètres `right` et `left`. Cela permet d'afficher une image sur une partie de la slide, et que le texte soit contraint à rester de l'autre côté.

On peut choisir le pourcentage de partage, par exemple `right:40%`.

En mettant plusieurs images en arrière-plan de la même slide (plusieurs images avec `bg`), elles vont se partager l'espace d'arrière-plan (ou par exemple l'espace de droite si l'une d'elles a le paramètre `right`).

Exemple :

```
![bg 90% vertical right:60% blur:3px](img/image.jpg)
![bg contain blur:3px](img/image.jpg)
![bg width:200 brightness:.8 sepia:50%](img/image.jpg)
```

Annexes

Arborescence conseillée

- *Dossier parent*
 - *img*
 - * `image.png`
 - `presentation.md`
 - `presentation.html`

Le fichier `presentation.html` serait issu de l'exportation du fichier Markdown au format HTML.

Modèle

Fichier `templateMarp.md` :

```
---
marp: true
theme: gaia
markdown.marp.enableHtml: true
paginate: true
style: |
  section {
    background-color: #fefefe;
    color: #333;
    font-size: 23px
  }

  section.invert {
```

```
background-color: #333;
color: #fefefe;
font-size: 23px
}

section h1 {
  text-align: center
}

section h2 {
  color: blue
}

img[alt~="center"] {
  display: block;
  margin: 0 auto;
}

blockquote {
  background: #f2f2f2;
  border-left: 10px solid #ac9393;
  margin: 1.5em 10px;
  padding: 0.5em 10px;
  color: #333;
  font-size: 18px;
}
blockquote:before{
  content: unset;
}
blockquote:after{
  content: unset;
}

/* Styling page number */
/* section::after {
  font-weight: bold;
  text-shadow: 1px 1px 0 #fff;
} */
/* Add total page number */
section::after {
  content: attr(data-marpit-pagination) '/' attr(data-marpit-pagination-total);
}
---
```



```
<!--
_class: invert
_transition: melt
_footer: Cette slide finira par une merveilleuse transition !
-->
```

La class `invert` a été définie dans le style CSS en directive globale pour inverser les couleurs.

> Une superbe citation (`blockquote`), stylisée dans le style CSS en directive globale.

Titre 1

<!--

_paginate: hold

_backgroundImage: "

linear-gradient(217deg, rgba(255,0,0,.8), rgba(255,0,0,0) 70.71%),

linear-gradient(127deg, rgba(0,255,0,.8), rgba(0,255,0,0) 70.71%),

linear-gradient(336deg, rgba(0,0,255,.8), rgba(0,0,255,0) 70.71%);"

_backgroundPosition: top 20px left 60px

_backgroundSize: 35%

_backgroundRepeat: no-repeat

-->

![bg 60% vertical right:35% blur:3px](https://marp.app/assets/marp.svg)

![bg contain drop-shadow:5px,5px,15px,rgba(0,0,255,0.4)](https://marp.app/assets/marp.svg)

![bg width:200 brightness:.8 sepia:50%](https://marp.app/assets/marp.svg)

Titre 2

* La pagination ici n'est pas incrémentée mais affichée grâce au paramètre `hold`.

* Et ces éléments de liste apparaissent quand on clique.

* Cette image n'est pas en arrière-plan : ![width:200 center brightness:.8 sepia:50%](https://marp.app/assets/marp.svg)

Texte qui apparaît directement.

<div data-marpit-fragment>

Texte qui apparaît après un clic de souris.

</div>

Texte qui était déjà présent.