

# Produire du texte proprement

documentation, présentation, gestion

Bruno BEAUFILS

# 1. Introduction

## 2. Outils

## 3. Markdown

## 4. pandoc

## 5. Liens

# Introduction

- Lors de vos différents projets vous allez devoir **documenter** vos travaux
  - ▶ en rédigeant des **procédures** de mise en place ;
  - ▶ en rédigeant des **comptes-rendus** des séances de travail ;
  - ▶ en rédigeant des **rapports** et **diaporamas** de présentation de vos travaux.
- Ce travail devra **obligatoirement** être fait
  - ▶ via des fichiers textes au format **Markdown**
  - ▶ gérés dans un dépôt **Git**
- Lors de vos séances d'autonomie vous devez donc
  - ▶ vous **documenter** sur le fonctionnement de ces outils
  - ▶ vous **familiariser** à leur usage
- **La qualité de cette documentation est une partie importante des évaluations**
- Ce document vous donne quelques bases et des liens pour vous aider
- Il est disponible en
  - ▶ [HTML](#)
  - ▶ [PDF](#)
  - ▶ Markdown dans [utils.md](#) (pratique pour comparer source et résultats)

## 1. Introduction

## 2. Outils

## 3. Markdown

## 4. pandoc

## 5. Liens

# Base

- **Markdown** est un langage de **balisage simple**
  - ▶ facile à lire et écrire par les humains et les programmes
  - ▶ qui permet de transformer un fichier balisé vers HTML ou PDF *automatiquement*
  - ▶ dont il existe plusieurs **variantes**
- **Pandoc** est un logiciel de **conversion** de documents
  - ▶ il permet de convertir des fichiers d'un format de balisage à un autre
  - ▶ [markup.rock](https://pandoc.org/)s permet de manipuler du texte directement via Pandoc
- **Git** est un outil de **contrôle de version** distribué qui permet
  - ▶ de suivre l'historique des modifications d'un ensemble de fichiers
  - ▶ de gérer des versions différentes en parallèle
  - ▶ de travailler à plusieurs sur les mêmes fichiers
  - ▶ de mettre en place des modes de collaborations différents
- **make** est un logiciel **d'automatisation de la construction** de fichiers
  - ▶ qui gère les dépendances temporelles entre fichiers
  - ▶ dans lequel on déclare les règles de construction par des fichiers texte (`makefile`)
  - ▶ historiquement plutôt utilisé pour le développement logiciel (notamment en C)
  - ▶ très adapté à tous projets devant gérer des dépendances entre fichiers

# Autres outils utiles

- **Org mode** est un mode **Emacs** d'édition de documents en texte simple pour
  - ▶ prendre des notes,
  - ▶ gérer des listes de tâches
  - ▶ planifier des projets
  - ▶ rédiger des documents ou des calepins de calculs
  - ▶ faire de la programmation littéraire
  - ▶ publier sous divers formats (HTML, PDF, etc.)
- **Mdoc** est le langage d'écriture de pages de manuel UNIX en texte simple
  - ▶ `man(7)`
  - ▶ `mdoc(7)`
  - ▶ `groff(7)`

## 1. Introduction

## 2. Outils

## 3. Markdown

## 4. pandoc

## 5. Liens

# Syntaxe générale

La syntaxe de Markdown permet

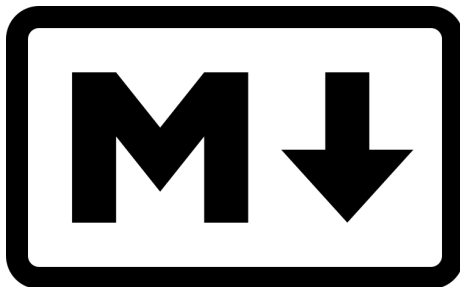
- de donner une **structure** au document
  - ▶ **paragraphes** : séparation par des lignes vides
  - ▶ **titres de différents niveaux** : lignes débutant par des #
  - ▶ **listes libres** : indentation de paragraphe avec lignes débutant par des - ou des \*
  - ▶ **listes ordonnées** : indentation avec lignes débutant par des 1.
  - ▶ **blocs de citations** : paragraphes préfixés par ">"
  - ▶ **blocs de codes** : paragraphes indentés par 4 espaces
- d'inclure des éléments de mise en forme **logique**  
*emphase*, **accentuation forte** et code
- d'inclure des liens vers des URL
  - ▶ <https://daringfireball.net/projects/markdown>
  - ▶ CommonMark
  - ▶ Markdown à la sauce Pandoc
  - ▶ Markdown à la sauce GitLab
  - ▶ Markdown à la sauce Github

Elle est décrite en détails ailleurs (cf liens de cette diapo)



# Images

```
![Texte alternatif pour  
HTML] (https://www.lifl.fr/~beaufils/logos/logo-markdown.png){  
width=50% }
```



**Figure** – Texte alternatif pour HTML

# Tableaux

- Plusieurs supports de tableaux accessibles via le [Markdown de pandoc](#)
- Celui utilisé par GitLab et GitHub est très basique
  - ▶ défini par `pipe_tables` dans pandoc

Outil	Utilité
pandoc	conversion de formats
pdflatex	composition de LaTeX en PDF
beamer	jeux de balisage LaTeX pour la composition de présentation

# Code source

Le code source peut être coloré en fonction de la syntaxe du langage

```
#include <stdio.h>
int main(int argc, char ** argv) {
    printf("Hello world !\n");
}
```

## 1. Introduction

## 2. Outils

## 3. Markdown

## 4. pandoc

## 5. Liens

# pandoc

- **Pandoc** est un logiciel de conversion de documents
  - ▶ il permet de convertir des fichiers d'un format de balisage à un autre
  - ▶ il définit une extension de Markdown : **Pandoc's Markdown**
- Options importantes
  - ▶ `-f` : format du fichier de départ
  - ▶ `-t` : format du fichier de sortie
  - ▶ `-o` : nom du fichier de sortie
- Plus de détails
  - ▶ `pandoc(1)`
  - ▶ **Documentation de la dernière version de Pandoc**

## 1. Introduction

## 2. Outils

## 3. Markdown

## 4. pandoc

## 5. Liens

# Markdown

- [Markdown Tutorial](#) : un tutoriel sur la syntaxe Markdown
- [Markdown guide](#) : un guide de référence sur Markdown
- Éditer du Markdown
  - ▶ [sous Emacs](#)
  - ▶ [sous VSCode](#)

# Git

- **Introduction à Git : cours et exercices**
  - ▶ par Pierre-Antoine CHAMPIN (IUT / Université de Lyon 1)
- **Learning Git Branching** : apprendre à gérer des branches
- **Pro Git** : le livre de référence
  - ▶ Comment faire pour
    - identifier l'auteur d'un commit
    - empêcher le suivi de certains fichiers
- **Oh Shit, Git !?!**
- **A hacker's guide to Git**
- **Git Commit**
  - ▶ How to Write a Git Commit Message
  - ▶ Best Practices for Git Commit Message
  - ▶ How to Write Better Git Commit Messages – A Step-By-Step Guide



# Make

- [Une présentation courte de make](#)
  - ▶ [par Christophe RIPPET \(Université de Grenoble Alpes\)](#)
- [Le manuel de GNU Make](#)
- [Makefile tutorial](#)