



AsciiDoc  
*Docs as Code*

# Table des matières

1. Introduction .....	1
2. Le format asciidoc .....	2
2.1. Compiler sa documentation .....	2
2.2. Une proposition d'environnement de travail .....	3
3. Références .....	4

---

# Chapitre 1. Introduction

La **Document as Code** (*Docs as Code*) est une philosophie de rédaction documentaire.

Pour favoriser la rédaction de documentations de qualité, les mêmes outils que ceux utilisés pour coder sont utilisés :

- gestionnaire de bugs,
- gestionnaire de version (**git**),
- revue de code,
- tests automatiques.

Plusieurs langages de balisage légers répondant à ces besoins sont apparus :

- le Markdown,
- le reStructuredText,
- l'Asciidoc.

## Le contenu avant la forme

Les avantages d'un tel dispositif sont nombreux :

- Le rédacteur se concentre uniquement sur le contenu et non sur la mise en forme comme c'est malheureusement trop souvent le cas avec les éditeurs de texte WYSIWYG (openoffice, word, etc.),
- Le même code source permet de générer des formats différents : pdf, html, word, confluence, epub, manpage, etc.
- Le travail collaboratif est grandement simplifié : code review, merge request, etc.
- Gestion des versions par branches ou tags git (pas une copie de fichier .doc).
- Edition de la documentation accessible à tous avec un simple éditeur de texte puisque la documentation est composée de fichiers plats (texte).

## Chapitre 2. Le format asciidoc

Ce langage de balisage léger a été créé en 2002 !

Le processeur initial était développé en python mais à depuis été réécrit en ruby (<https://asciidoctor.org/>).

La syntaxe asciidoc a eu du mal à percer au bénéfice du markdown, mais sa grande lisibilité et le lead de Dan Allen (<https://twitter.com/mojavelinux>) lui permettent de rattraper aujourd'hui son retard.

Voici un petit exemple de source asciidoc :

```
= Hello, AsciiDoc!  
Doc Writer <doc@example.com>  
  
An introduction to http://asciidoc.org[AsciiDoc].  
  
== First Section  
  
* item 1  
* item 2
```

Comme vous pouvez le constater, le texte (malgré le balisage) reste lisible, la syntaxe est facilement assimilable et n'est pas trop lourde comparée à ce qui se faisait en latex.

Use AsciiDoc for document markup. Really. It's actually readable by humans, easier to parse and way more flexible than XML.

— Linus Torvalds

Une page référence les possibilités de la syntaxe asciidoc :

- <https://asciidoctor.org/docs/asciidoc-syntax-quick-reference/>

### 2.1. Compiler sa documentation



La documentation d'installation est disponible ici : <https://asciidoctor.org/docs/install-toolchain/>.

Comme pour un programme, la documentation nécessite une phase de compilation.

Après avoir installé l'environnement asciidoc, la commande suivante permet de compiler son document `.adoc` en `html5` (format par défaut) :

```
asciidoctor index.adoc
```

Une image docker existant, il est facile d'utiliser la CI d'un environnement tel que GitLab pour générer automatiquement la documentation à chaque commit, tag, etc.

```
build:
  stage: build
  image: asciidoctor/docker-asciidoctor
  script:
    - asciidoctor-pdf -a icons="font" -a lang="fr" -D public index.adoc
  only:
    - master
  artifacts:
    name: "compilation-pdf"
    paths:
      - public/
```

## 2.2. Une proposition d'environnement de travail

Atom (<https://atom.io/>) est un éditeur de code.

Après avoir installé Atom en suivant cette documentation : <https://flight-manual.atom.io/getting-started/sections/installing-atom/>, activer les packages suivants :

- **asciidoc-image-helper**
- **asciidoc-assistant**
- **asciidoc-preview**
- **autocomplete-asciidoc**
- **language-asciidoc**

## Chapitre 3. Références

- <http://www.writethedocs.org/guide/docs-as-code/>
- <https://www.technologies-ebusiness.com/enjeux-et-tendances/moi-code-madoc>