

Doc-as-code avec Aciidoc & PlantUml

Table of Contents

Introduction.....	1
Everything-as-code.....	1
Documentation-as-code.....	1
PlantUML	1
Avantages d’Asciidoc sur Markdown.....	1
Plusieurs formats de sortie.....	1
Générateurs basés sur AsciiDoctor.....	2
Cette présentation reveal.js est générée via asciidoctor.....	2
Exemple de site.....	2

Introduction

AsciiDoc par la documentation de Asciidoctor : <https://asciidoctor.org/docs/user-manual/>

Everything-as-code

- Regroupe la configuration et les sources
- Versionne avec le code
- Profite de la revue de code collaborative
- Facilite l'automatisation
- Testable en pipeline CI/CD

Documentation-as-code

- Markup Language
 - Latex
- Lightweight/Simple Markup Language
 - Markdown
 - AsciiDoc

PlantUML

PlantUML est un outil diagram-as-code

<http://plantuml.com>

Avantages d'AsciiDoc sur Markdown

- Standard
- Extensible
- Inclusion d'autres fichiers
- Inclusion native de PlantUML



Plus de détail ici sur les défauts de Markdown : <http://ericholscher.com/blog/2016/mar/15/dont-use-markdown-for-technical-docs/>

Plusieurs formats de sortie

- HTML

- PDF
- Docbook
- Reveal.js

Générateurs basés sur Asciidoctor

- Outil ligne de commande (Ruby)
- Plugin Maven
- Editeur AsciiDocFX (sans options)

Cette présentation reveal.js est générée via asciidoctor

Source : [PRES-asciidoc.adoc](#)

Exemple de site

<https://bcouetil.gitlab.io/academy/>

- Sans écrire de code frontend
- Incluant différents formats
 - Source adoc
 - HTML
 - PDF
 - Reveal.js