

# Ascididoctor

# Me

---

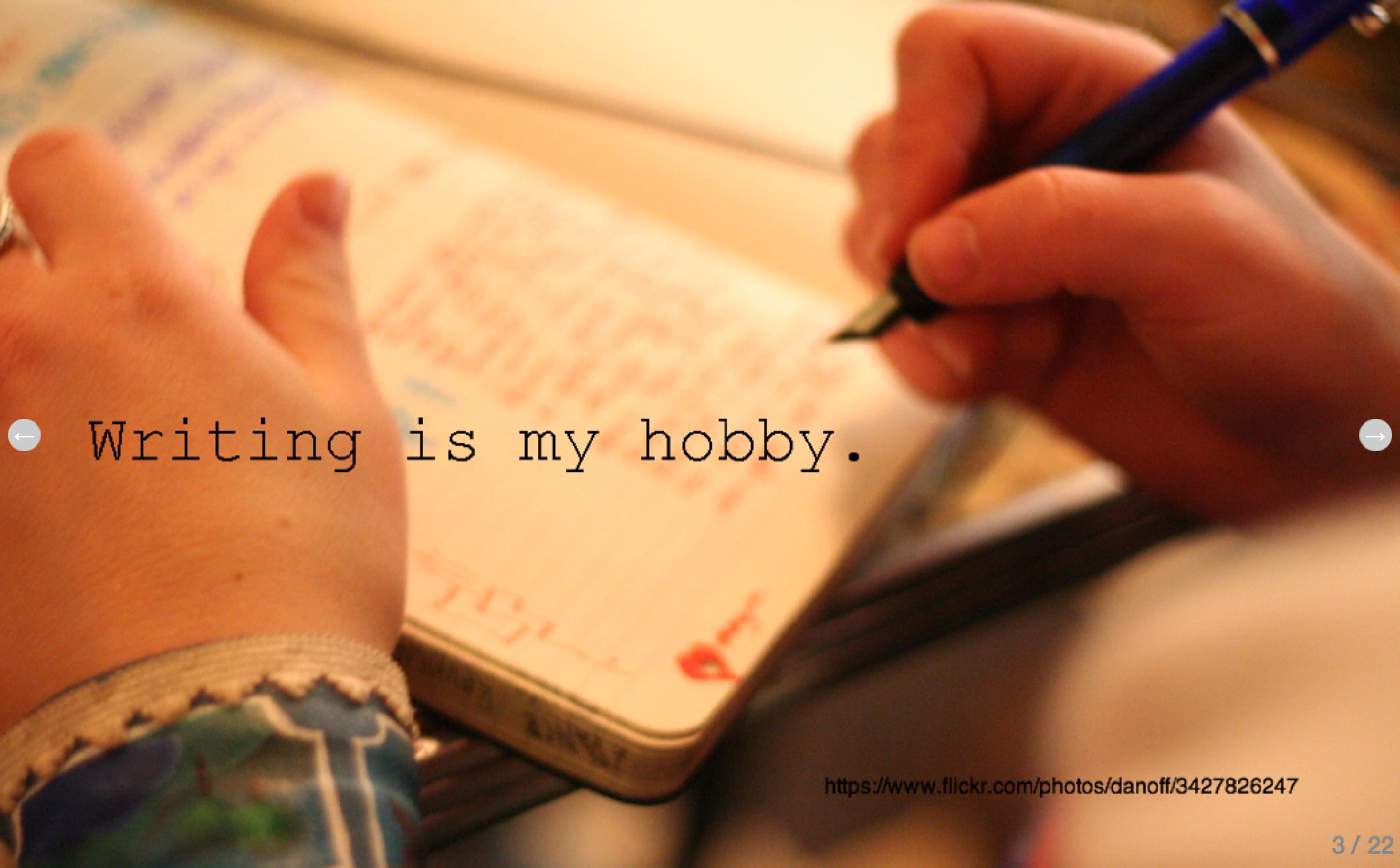
 @binout

 <https://github.com/binout>



- Développeur Java depuis presque 10 ans
- Technical Leader chez Lectra, numéro un mondial des solutions dédiées à l'industrie du textile (machines et logiciels)
- Membre du Bordeaux JUG





Writing is my hobby.

<https://www.flickr.com/photos/danoff/3427826247>

# Une solution : Asciidoc

---

Langage de *balisage*, créé en 2002 avec un processeur  
en Python

**A s c i i D o c**  
Text based document generation



# Une solution : Asciidoc

---

Langage de *balisage*, créé en 2002 avec un processeur  
en Python



Parce que :



# Une solution : Asciidoc

---

Langage de *balisage*, créé en 2002 avec un processeur  
en Python



Parce que :

- on se concentre plus sur le fond que sur la forme

# Une solution : Asciidoc

Langage de *balisage*, créé en 2002 avec un processeur  
en Python



Parce que :

- on se concentre plus sur le fond que sur la forme
- c'est du texte, donc un éditeur classique suffit



# Une solution : Asciidoc

Langage de *balisage*, créé en 2002 avec un processeur  
en Python



Parce que :

- on se concentre plus sur le fond que sur la forme
- c'est du texte, donc un éditeur classique suffit
- on peut gérer l'historique avec un SCM



# Une solution : Asciidoc

Langage de *balisage*, créé en 2002 avec un processeur  
en Python



Parce que :

- on se concentre plus sur le fond que sur la forme
- c'est du texte, donc un éditeur classique suffit
- on peut gérer l'historique avec un SCM
- à partir d'une même source, on peut publier vers plusieurs formats

# Une solution : Asciidoc

Langage de *balisage*, créé en 2002 avec un processeur  
en Python



Parce que :

- on se concentre plus sur le fond que sur la forme
- c'est du texte, donc un éditeur classique suffit
- on peut gérer l'historique avec un SCM
- à partir d'une même source, on peut publier vers plusieurs formats
- on a un peu l'impression de hacker ;-)

# Exemple

```
= Hello, Bdx.io !
```

```
Benoit Prioux <benoit.prioux@gmail.com>
```

```
Introduction à http://asciidoc.org[AsciiDoc].
```

```
== Première Section
```

- ```
* foo  
* bar
```

```
== Deuxième Section
```

- ```
1. item 1  
2. item 2
```

```
NOTE: C'est l'heure de la démo !
```



# Asciidoctor

<https://github.com/asciidoctor>

- Implémentation open-source écrite en Ruby
- Permet de convertir des fichiers asciidoc vers différents backends : docbook, html5, epub, pdf et même deckjs !
- 12 août 2014 : sortie de la version 1.5.0
  - 1<sup>ère</sup> version majeure après 2 ans de développement
  - 50 contributeurs, 1800 commits, 1500 tests



Dan Allen  
mojavelinux

# Ils utilisent déjà asciidoctor



O'REILLY®

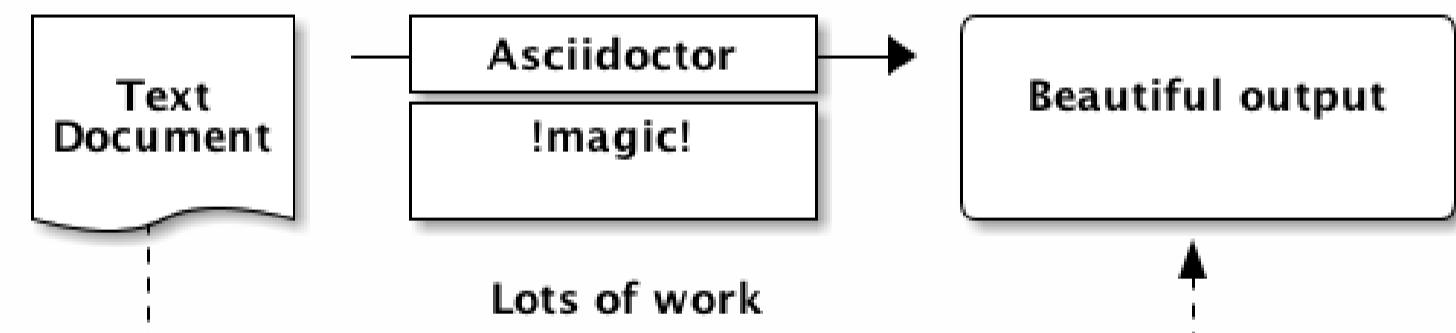
The O'Reilly logo features the word "O'REILLY" in a large, bold, black sans-serif font, with a red "O". A registered trademark symbol (®) is positioned to the right of the "Y".

# Asciidoctor est extensible ...



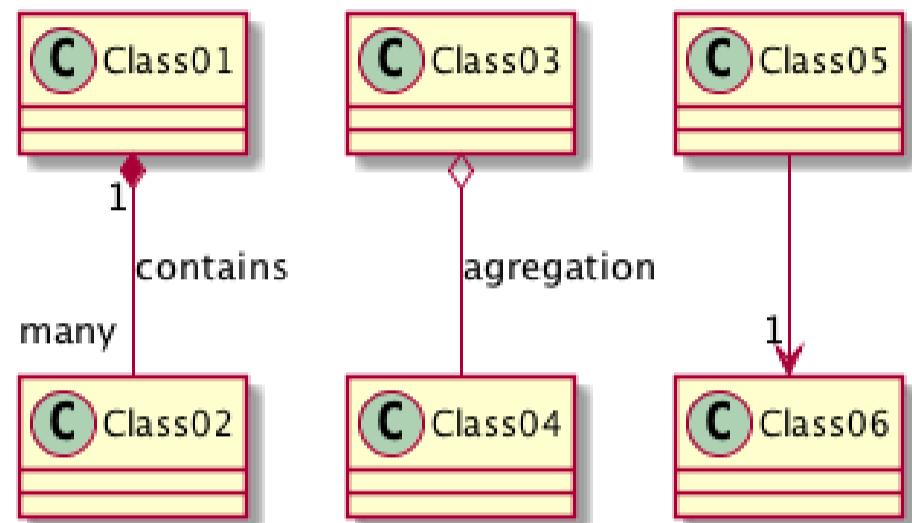
# asciidocor-diagram

```
[ditaa]
-----
+-----+ +-----+----+ /-----\
|     | ---+ Asciidocor +-> | |
| Text | +-----+ |Beautiful output|
| Document| !magic! | |
| {d}| | |
+-----+ +-----+ \-----/
:
| Lots of work |
+-----+
-----
```



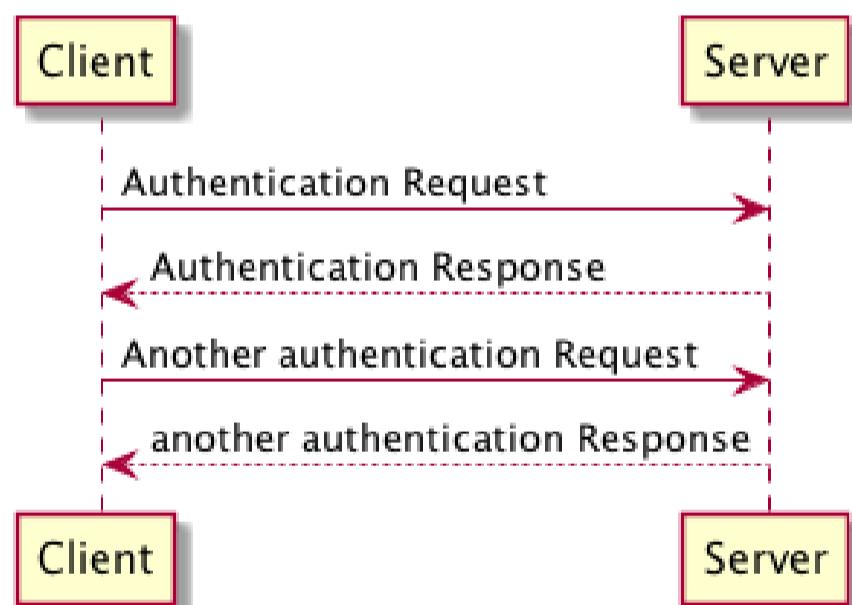
# asciidocctor-diagram et UML

```
[plantuml]
-----
Class01 "1" *-- "many" Class02 : contains
Class03 o-- Class04 : aggregation
Class05 --> "1" Class06
-----
```



```
[plantuml]
-----
Client -> Server: Authentication Request
Server --> Client: Authentication Response

Client -> Server: Another authentication Request
Client <-- Server: another authentication Response
-----
```



# Asciidoctor et les développeurs

in  
code  
we  
trust



# Je suis développeur Ruby

## Installation

```
gem install asciidoctor
```

## CLI

```
$ asciidoctor -D output README.adoc
```

## Ruby API

```
require 'asciidoctor'  
...  
Asciidoctor.convert_file 'README.adoc', to_file: true,  
safe: 'safe'
```

# Je suis développeur Java

---



# Je suis développeur Java

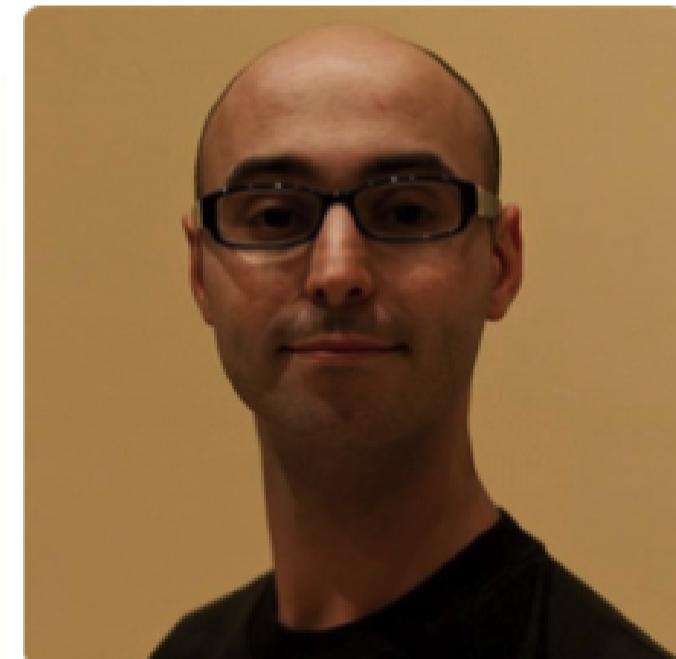
---



# Je suis développeur Java



+



Alex Soto  
lordofthejars

= AsciidoctorJ

# Asciidoctorj

<https://github.com/asciidoctor/asciidoctorj>

```
Asciidoctor asciidoctor = Asciidoctor.Factory.create();  
  
Options options = options().backend("html5").get();  
String rendered = asciidoctor.convert("*Gras* ou  
_italique_?", options);  
  
System.out.println(rendered);
```



# Asciidoctorj

<https://github.com/asciidoctor/asciidoctorj>

```
Asciidoctor asciidoctor = Asciidoctor.Factory.create();  
  
Options options = options().backend("html5").get();  
String rendered = asciidoctor.convert("*Gras* ou  
_italique_?", options);  
  
System.out.println(rendered);
```

## Console

```
<div class="paragraph">  
<p><strong>Gras</strong> ou <em>italique</em> ?</p>  
</div>
```

# Asciidoctor Maven Plugin

<https://github.com/asciidoctor/asciidoctor-maven-plugin>

```
<plugin>
  <groupId>org.asciidoctor</groupId>
  <artifactId>asciidoctor-maven-plugin</artifactId>
  <version>1.5.0</version>
  <executions>
    <execution>
      <id>output-html</id>
      <phase>generate-resources</phase>
      <goals>
        <goal>process-asciidoc</goal>
      </goals>
    </execution>
  </executions>
</plugin>
```

# Asciidoctor Gradle Plugin

<https://github.com/asciidoctor/asciidoctor-gradle-plugin>

```
buildscript {  
    repositories {  
        jcenter()  
    }  
  
    dependencies {  
        classpath 'org.asciidoctor:asciidoctor-gradle-  
plugin:1.5.0'  
    }  
}  
  
apply plugin: 'org.asciidoctor.gradle.asciidoctor'
```

# Asciidoctor Ant Task

<https://github.com/asciidoctor/asciidoctor-ant>

```
<project
  xmlns:asciidoctor="antlib:org.asciidoctor.ant">
  ...
  <target name="doc">
    <taskdef uri="antlib:org.asciidoctor.ant"
      resource="org/asciidoctor/ant/antlib.xml"
      classpath="lib/asciidoctor-ant-
      1.5.0.jar"/>

      <asciidoctor:convert
      sourceDirectory="src/asciidoc"
      outputDirectory="target"/>
    </target>
  ...
</project>
```

# Asciidoclet

<https://github.com/asciidoctor/asciidoclet>

```
/**  
 * This class has the following  
features:  
 *  
 * - Support for *foo*  
 * - Support for bar  
 */  
public class Thing implements  
Something { ... }
```

PACKAGE CLASS USE TREE DEPRECATED INDEX HELP

PREV CLASS NEXT CLASS FRAMES NO FRAMES

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

net.binout.asciidoc

**Class Thing**

java.lang.Object  
net.binout.asciidoc.Thing

All Implemented Interfaces:

Something

public class Thing  
extends Object  
implements Something

This class has the following features:

- Support for foo
- Support for bar

**Constructor Summary**

Constructors

Constructor and Description

Thing()

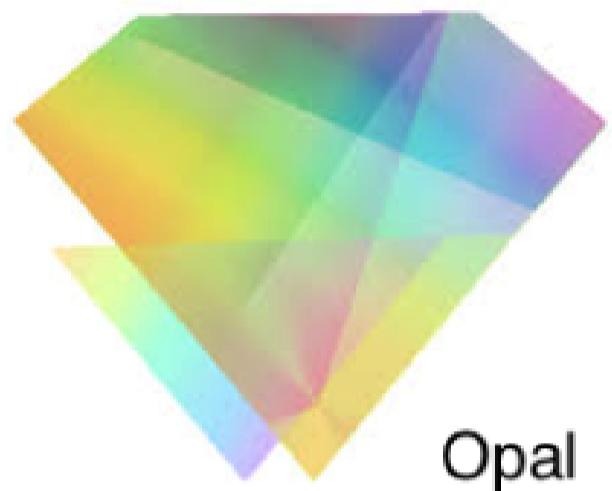
**Method Summary**

# Je suis développeur Javascript

---



# Je suis développeur Javascript



# Je suis développeur Javascript



Opal



+



Anthony  
Quérouil  
anthonny

= Asciidoctor.js

# Asciidoctor.js

<https://github.com/asciidoctor/asciidoctor.js>

```
var content = "http://asciidoctor.org[*Asciidoctor*] "
+
"running on http://opalrb.org[_Opal_] " +
"brings AsciDoc to the browser!";

var options = Opal.hash2(['doctype', 'attributes'],
{doctype: 'inline', attributes: ['showtitle']});

var html = Opal.Asciidoctor.$convert(asciidoc,
options);

console.log(html);
```

# Asciidoctor.js et Chrome

The screenshot shows the Asciidoctor.js extension page on the Chrome Web Store. The page includes a puzzle piece icon with a red 'A', a title 'Asciidoctor.js Live Preview', a rating of 4.5 stars from 25 reviews, and categories like 'Productivité' and 'Auteur: Guillaume Grossetie'. It also shows 1,328 users and a green button 'AJOUTÉ À CHROME'. Below the header is a navigation bar with tabs: PRÉSENTATION (selected), DÉTAILS, AVIS, and ARTICLES SIMILAIRES. A slide navigation bar with five dots is centered below the tabs. The main content area displays a screenshot of a browser window showing the Asciidoctor README.adoc file. The browser title is 'file:///asciidoctor/README.adoc'. The page content includes the Asciidoctor title, a usage section, and a terminal command '\$ asciidoctor --version'. To the right of the browser screenshot is a sidebar with the following text:

Render AsciiDoc (.ad, .adoc, .asc, .asciidoc) as HTML inside Chrome!

This project uses Asciidoctor.js to render AsciiDoc as HTML.

= Usage

1. Check Allow access to file URLs in chrome://extensions
2. Open local or remote .ad, .adoc, .asc, .asciidoc file in Chrome
3. Enjoy!

= Options

The extension can be configured via an options page.

To open the options page, right-click the extension icon and choose Options on the menu. You can also go to chrome://extensions and click the Options link.

The options page let you add custom attributes or change the theme of the

# Merci

<https://github.com/binout/asciidoc-quickie/>

## Asciidoctor User Manual

Sarah White – [@carbonfray](#) · Dan Allen – [@mojavelinux](#)



*This document is under active development and discussion!*

If you find errors or omissions in this document, please don't hesitate to [submit an issue or open a pull request](#) with a fix. We also encourage you to ask questions and discuss any aspects of the project on the [mailing list](#) or [IRC](#). New contributors are always welcome!

This manual assumes you are using Asciidoctor to produce and render your document. Asciidoctor implements more syntax, attributes and functions than the legacy AsciiDoc.py processor. [Appendix A](#) lists which features are available to the Asciidoctor and AsciiDoc processors.

## Introduction to Asciidoctor



Section Pending

## 1. What is Asciidoctor?

Asciidoctor is a *fast* text processor and publishing toolchain for converting AsciiDoc content to HTML5, EPUB3, PDF, DocBook 5 (or 4.5) slidedecks and other formats. Asciidoctor is written in Ruby, packaged as a RubyGem and published to [RubyGems.org](#). The gem is also packaged in several Linux distributions, including Fedora, Debian and Ubuntu. Asciidoctor is open source, [hosted on GitHub](#), and released under the MIT license.

### Table of Contents

#### *Introduction to Asciidoctor*

1. What is Asciidoctor?
  - 1.1. The Big Picture
  - 1.2. Asciidoctor on the JVM
  - 1.3. Asciidoctor.js
  - 1.4. Asciidoctor's most notable benefits
  - 1.5. Compared to AsciiDoc
  - 1.6. Compared to MarkDown

#### *Quick Starts*

2. Installation Quick Start
3. Usage Quick Start
  - 3.1. Using the Command Line Interface
  - 3.2. Using the Ruby API
4. Syntax Quick Start
5. Custom Output Quick Start

#### *Getting Started*

6. System Requirements
7. Installing the Asciidoctor Ruby Gem
  - 7.1. Install with Bundler
  - 7.2. Install with yum
  - 7.3. Install with apt-get
8. Upgrading the Asciidoctor Ruby Gem
9. Extensions and Integrations