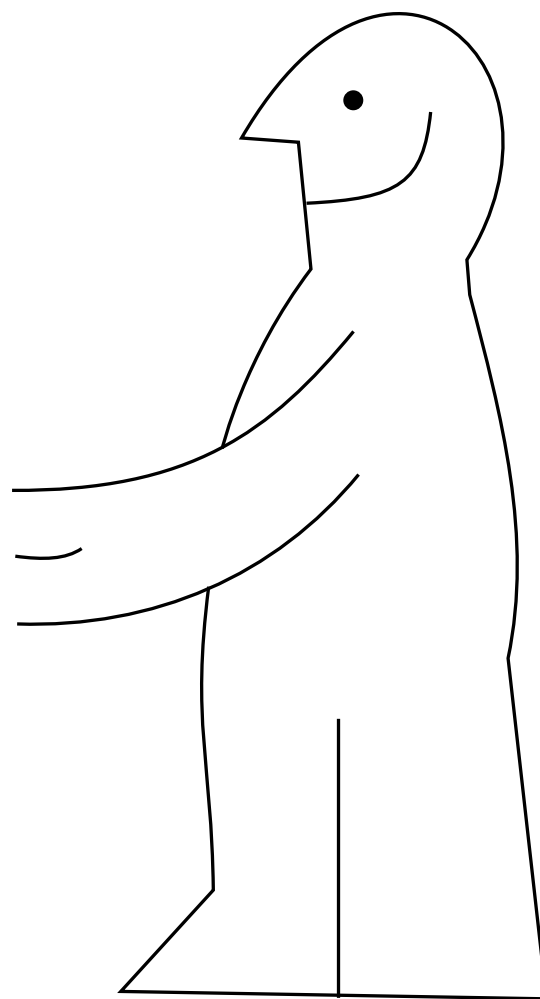
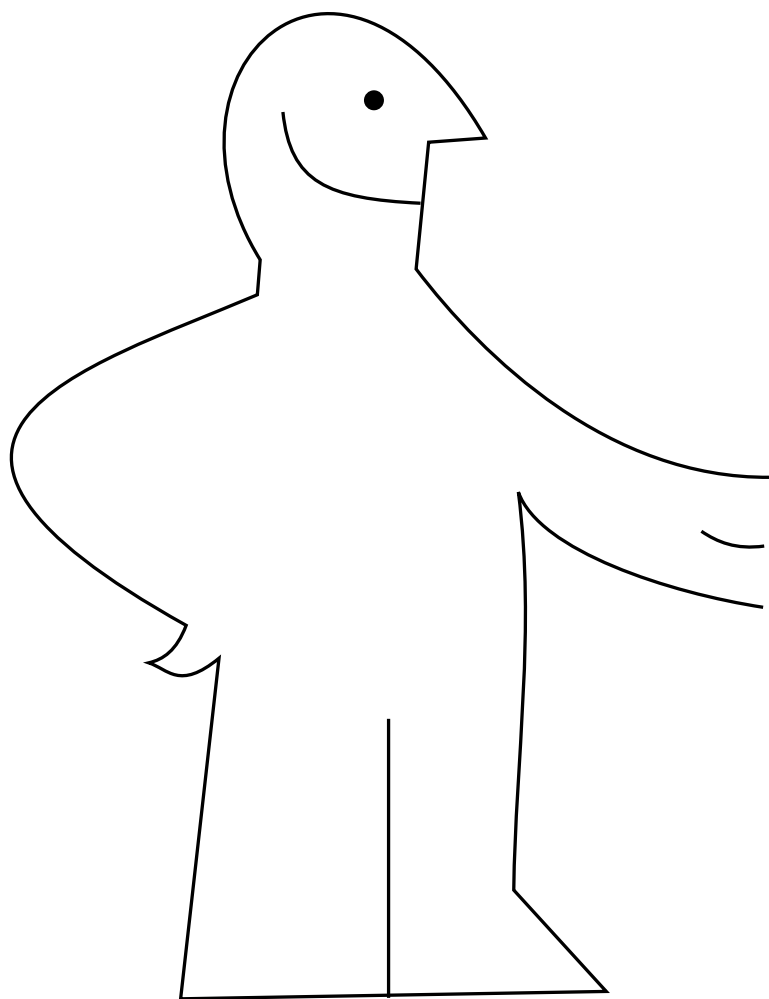




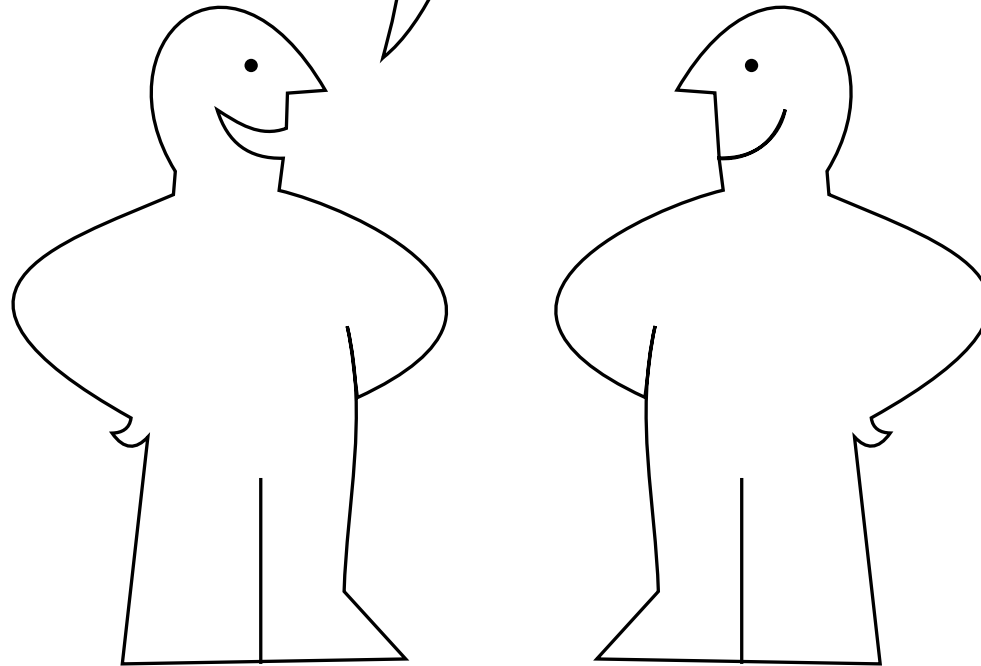
# DISCOVER THE ZEN OF WRITING WITH **ASCIIDOCTOR**

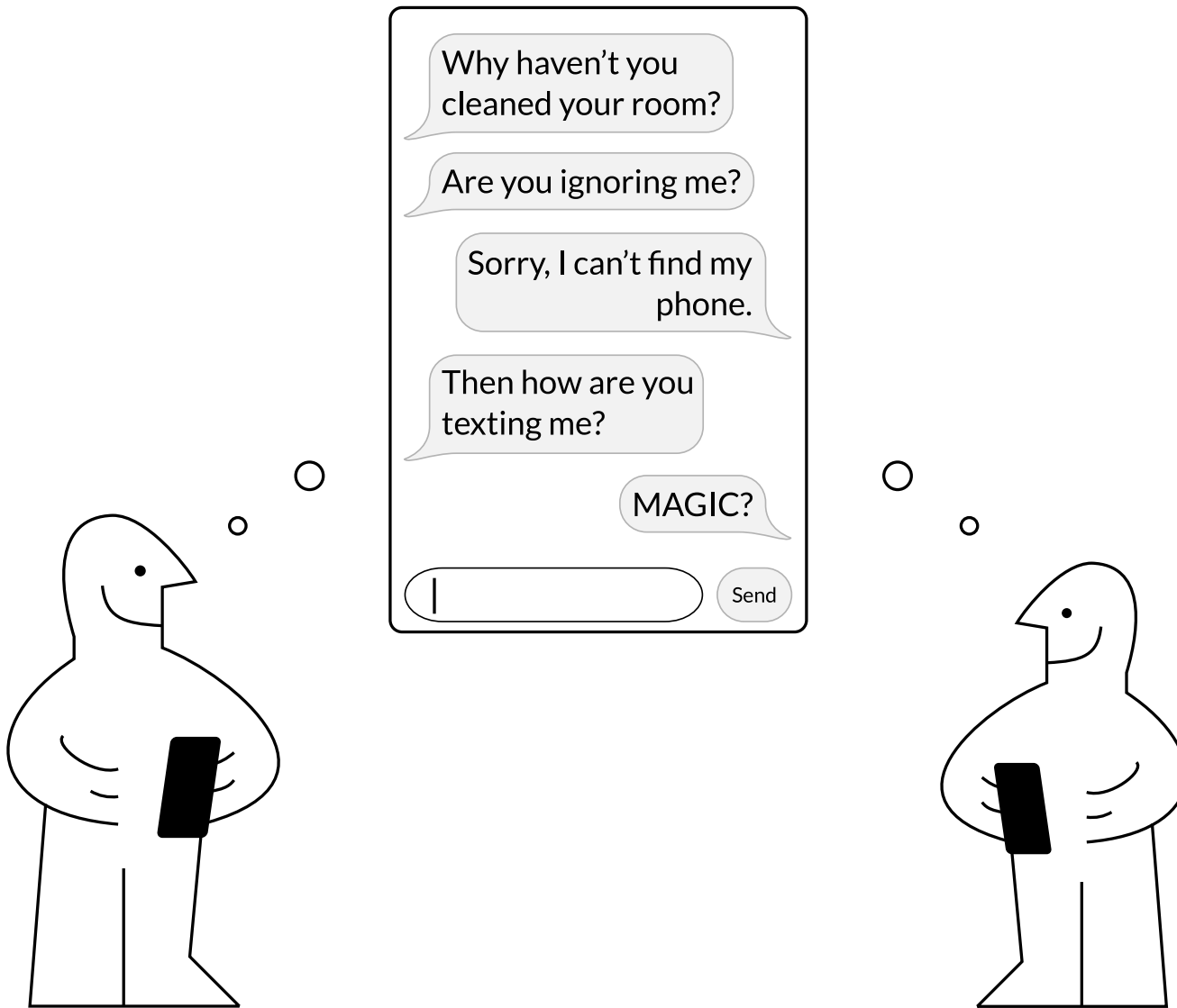
Dan Allen - [@mojavelinux](#)





Wait 'til you hear this!





Why haven't you  
cleaned your room?

Are you ignoring me?

Sorry, I can't find my  
phone.

Then how are you  
texting me?

MAGIC?

Send

What's happening?

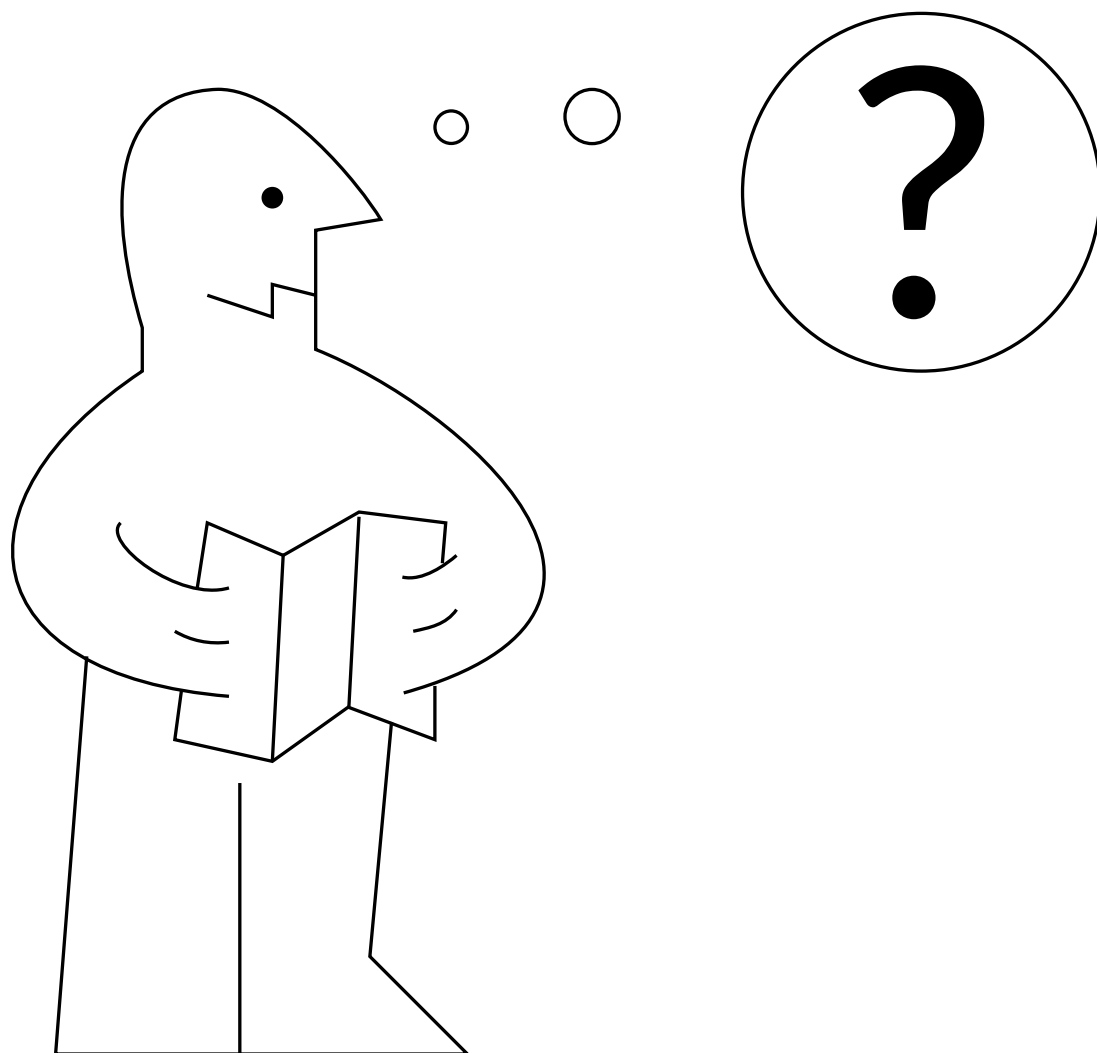


|



140

Tweet



*Writing is a #\$\$%@! ordeal.*

— Martijn Dashorst



**POURQUOI ?**

# A: STRUCTURE

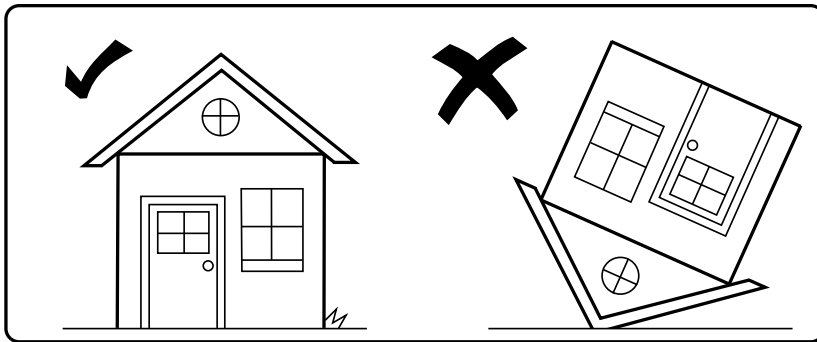
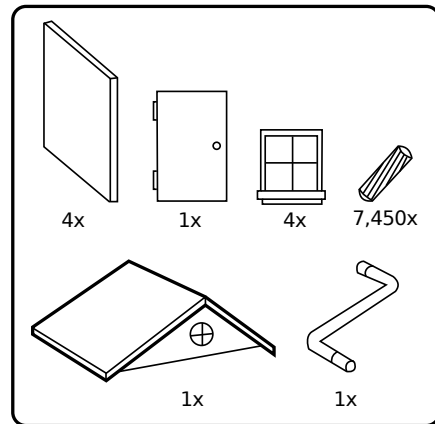
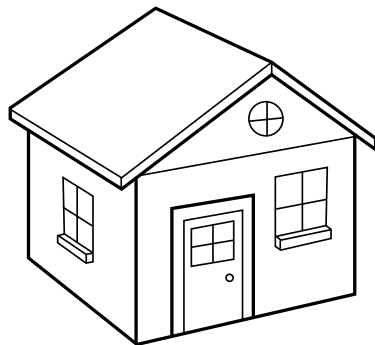
- Why Gradle?
  - Broad Platform Support
    - Java & JVM languages
    - Android's build system
    - ...
  - Robust Dependency Management
    - Straightforward dependencies
    - ...
  - Powerful Yet Concise Logic
    - ...

# **STATUS:** IT'S COMPLICATED

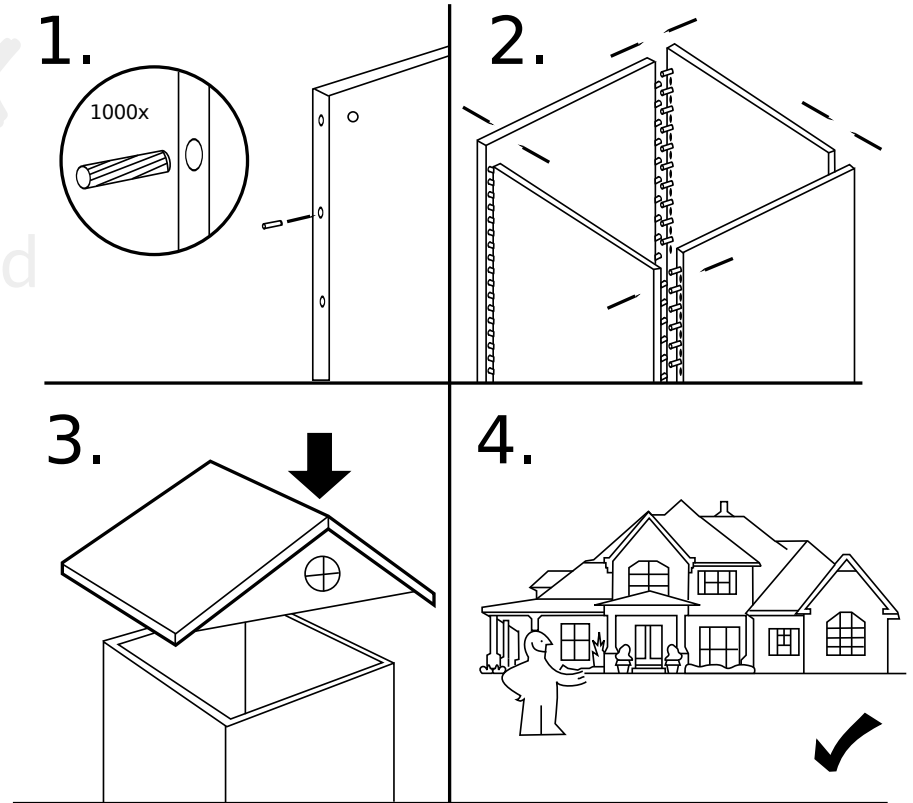
We fabricate even more barriers.

Let's fix that.

# HÖUSS



Bad



```
<class accessibility="public"
  name="HelloWorld">
  <anonymousBlock>
    <method type="static"
      accessibility="public"
      inheritability="final"
      name="main">
      <arguments>
        <variableArguments name="args"
          type="java.lang.String"/>
      </arguments>
      <anonymousBlock>
        <methodCall method="println"
          object="java.lang.System.out">
          <argument value="Hello World!"
            type="java.lang.String"/>
          </methodCall>
        </anonymousBlock>
      </method>
```

## Text Block

Content

Colors

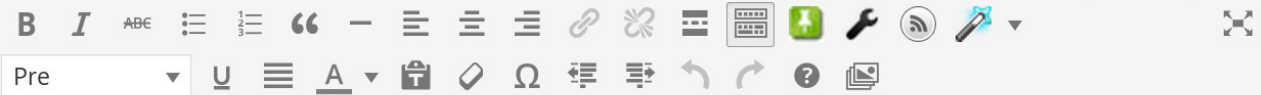
### Content

Enter some content for this textblock

 Add Media

Visual

Text



```
repositories {  
    jcenter()  
}  
  
dependencies {  
    compile project(':api')  
    compile 'org.slf4j:slf4j-api:1.7.10'  
    compile 'com.google.guava:guava:18.0'  
    runtime 'org.slf4j:slf4j-simple:1.7.10'  
    testCompile 'junit:junit:4.12'  
    testCompile 'org.mockito:mockito-all:1.10.19'  
}
```

pre

Save



Good

= Getting Started with Java  
Author Name

Here's your first Java application.

.HelloWorld.java  
[source,java]

----

```
public class HelloWorld {  
    public static void main(String args[]) {  
        System.out.println("Hello, world!"); <1>  
    }  
}
```

----

<1> Prints "Hello, world!" to the console.

Compile this source to a class file using `javac`.  
Then, run the compiled class file using `java`.



**DOCS == CODE!**



# ASCIIDOC

syntax



# ASCIIDOCTOR

parser + tools



edit



read



version



share



convert

# SYNTAX TOUR

# PARAGRAPHS

No special markup is required.

A paragraph is just consecutive lines of text.

To start a paragraph, offset it by a blank line.

# HEADINGS

= Document Title

== Section Level 1

=== Section Level 2

==== Section Level 3

===== Section Level 4

===== Section Level 5

= Section Level 0 (book doctype only)

# LISTS

## Unordered

```
* item 1  
** sub-item  
* item 2  
* item 3
```

## Ordered

```
. item 1  
.. sub-item  
. item 2  
. item 3
```

# INLINE FORMATTING

\*bold\*

**bold**

*\_italic\_*

*italic*

`<code>`

`<code>`

[role]#custom#

custom



# LINKS

`http://google.com`

`http://google.com[Google]`

See `<<syntax>>`

`<<syntax,Take a tour>>`

<http://google.com>

[Google](#)

See [Syntax Tour](#)

[Take a tour](#)

# “FENCED” BLOCKS

-- open

---

----, ``` listing, source

---

... literal

---

==== example, admonition

---

\*\*\* sidebar

---

----, "" quote

# SOURCE CODE

```
:source-highlighter: coderay
```

```
[source,java]
```

```
----
```

```
public class HelloWorld {  
    public static void main(String args[]) {  
        System.out.println("Hello, world!"); <1>  
    }  
}
```

```
----
```

```
<1> Prints "Hello, world!" to the console.
```

# ADMONITIONS

NOTE: For your info.

IMPORTANT: Don't forget!

# MEDIA

## Image

```
image::tux.png[Tux, 265, 314]
```

## Video

```
video::SCZF6I-Rc4I[youtube, 853, 480]
```

## Icon

```
icon:heart[2x]
```

# MACROS

## Inline macro

```
<name>:<target> [<attributes>]
```

## Block macro

```
<name>::<target> [<attributes>]
```

Limit the markup you *have* to type,  
not the markup you *can* type.

# EXTEND IT!

- Custom syntax (issue:<id>[])
- Custom converters (PDF, reveal.js, ...)
- AST transforms



**TOOLS**



+

`gists`

+

*gitbook.io*

branch: master asciidoctor-chrome-extension / README.adoc

 **Mogztter** on Dec 3, 2014 Prepare 1.5.2.100 release

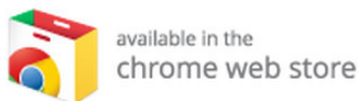
2 contributors  

124 lines (83 sloc) | 3.871 kb

Raw Blame History  

# Asciidoctor Chrome Extension

This project uses [Asciidoctor.js](#) to preview AsciiDoc as HTML inside the Chrome (or Chromium) web browser!



Project health: build passing

## Usage

1. Install extension from [Chrome Web Store](#)
2. Check `Allow access to file URLs` in `chrome://extensions`
3. Open local or remote `.ad`, `.adoc`, `.asc`, `.asciidoc` file in Chrome
4. Enjoy!

### Warning

Due to a [Chromium bug](#) the `Allow access to file URLs` is lost on extension update. If preview of local files isn't working, make sure to uncheck and recheck the `Allow access to file URLs`.

DocGist

Page Source

Project Home

Share ▾



GitHub Gist/File / Dropbox URL

# Asciidoctor Chrome Extension

This project uses [Asciidoctor.js](#) to preview AsciiDoc as HTML inside the Chrome (or Chromium) web browser!



available in the  
chrome web store

Project health: build passing

## Usage

1. Install extension from [Chrome Web Store](#)
2. Check `Allow access to file URLs` in `chrome://extensions`
3. Open local or remote `.ad`, `.adoc`, `.asc`, `.asciidoc` file in Chrome
4. Enjoy!



Due to a [Chromium bug](#) the `Allow access to file URLs` is lost on extension update. If preview of local files isn't working, make sure to uncheck and recheck the `Allow access to file URLs`.

|     |  |  |     |  |  |
|-----|--|--|-----|--|--|
| ✱   | @@ -131,7 +131,7 @@ In the HTML output, the footer content is inserted inside the footer div (i.e.,  |  |     |  |  |
| 131 | In the DocBook output, the footer content is inserted immediately before the ending <code>`&lt;/article&gt;`</code> or <code>`&lt;/book&gt;`</code> element.   |  | 131 | In the DocBook output, the footer content is inserted immediately before the ending <code>`&lt;/article&gt;`</code> or <code>`&lt;/book&gt;`</code> element.   |  |
| 132 |  |  | 132 |  |  |
| 133 | .docinfo   |  | 133 | .docinfo   |  |
| 134 | -If you want to add content to the footer of a specific document, put the content in the file <code>`&lt;docname&gt;-footer.html`</code> (for HTML output) or <code>`&lt;docname&gt;-footer.xml`</code> (for DocBook output), where <code>`&lt;docname&gt;`</code> is the name of the document without the AsciiDoc extension. |  | 134 | +If you want to add content to the footer of a specific document, put the content in the file <code>`&lt;docname&gt;-docinfo-footer.html`</code> (for HTML output) or <code>`&lt;docname&gt;-docinfo-footer.xml`</code> (for DocBook output), where <code>`&lt;docname&gt;`</code> is the name of the document without the AsciiDoc extension. |  |
| 135 | Then, set the attribute docinfo in the source document.  |  | 135 | Then, set the attribute docinfo in the source document.  |  |
| 136 |  |  | 136 |  |  |
| 137 | .docinfo1  |  | 137 | .docinfo1  |  |
| ✱   |  |  |     |  |  |



Table 1. Docinfo attributes and file names

Footer docinfo files are differentiated from header docinfo files by adding `- footer+` to the file name. In the HTML output, the footer content is inserted inside the footer div (i.e., `<div id="footer">` ). In the DocBook output, the footer content is inserted immediately before the ending `</article>` or `</book>` element.

If you want to add content to the footer of a specific document, put the content in the `<docname>-footer.html` `<docname>-docinfo-footer.html` (for HTML output) or `<docname>-footer.xml` `<docname>-docinfo-footer.xml` (for DocBook output), where `<docname>` is the name of the document without the AsciiDoc extension. Then, set the attribute `docinfo` in the source document.

If you want to add content to the footer of all documents in the same directory, put the content in the file `docinfo-footer.html` or `docinfo-footer.xml`. Then, set the attribute `docinfo1` in the source document to enable the feature.



**EDITING & LIVE PREVIEW**



# Asciidoctor.js Live Preview

from [Guillaume Grossetie](#)

★★★★★ (28)

[Productivity](#)

2,821 users

+ ADD TO CHROME



OVERVIEW

REVIEWS

SUPPORT

RELATED

g+1

58

file:///asciidoctor/stem.adoc

(2) There is a trick in Sage for finding relevant information. Suppose, for example, that we are doing computations in  $\mathbb{F}_{25}$ , the field of 25 elements. We have set it up in Sage as the field  $K$ .  $\langle t \rangle$  as above. Suppose that we want to find the minimal polynomial of  $t$ . Instead of googling, we type `t`, (note the dot) then press the TAB key. Sage displays a list of methods applicable to the object `t`. One of these is `minimal_polynomial`. Now we know what to do:

```
sage: t.minimal_polynomial()
x^2 + 4*x + 2
```

(3) Guessing is also an effective method for finding the names of Sage methods. Full names like `minimal_polynomial` are preferred over abbreviations like `mpoly`, `mpol`, `minpoly`, `min_poly`, etc. Thus are great many names in Sage are guessable.

References: 1, 2

## 1.2 Ideals in Polynomial Rings

The rings of interest to us in this course are polynomial rings and their quotients. Fix a field  $k$  and let

$$k[x_1, \dots, x_n] = \left\{ f \mid f = \sum c_a x^a \right\},$$

be the set of polynomials in  $n$  variables with coefficients in  $k$ . This set has various structures: it is a  $k$ -vector space, and it is a commutative ring. It is also an integral domain: if  $fg = 0$ , then  $f = 0$  or  $g = 0$ . In other words,  $k[x_1, \dots, x_n]$  has no divisors of zero.

An ideal  $I$  in a ring  $R$  is a subset with two closure properties: (i) if  $f, g \in I$ , then  $f + g \in I$ , (ii) if  $f \in R$  and  $g \in I$ , then  $fg \in I$ . One way of giving an ideal is to give a set of generators  $f_1, \dots, f_r$  in  $k[x_1, \dots, x_n]$ . Consider the vector space

$$I(f_1, \dots, f_r) = \{ a_1 f_1 + \dots + a_r f_r \mid a_i \in k[x] \}.$$

One easily checks that  $I(f_1, \dots, f_r)$ , which we sometimes write simply as  $(f_1, \dots, f_r)$ , is an ideal. We say that  $\{f_i\}$  is a set of generators of  $I$ .

Now suppose given a set  $S \subset k^n$ . Consider the set

$$I(S) = \left\{ f \in k[x_1, \dots, x_n] \mid f(S) = 0 \right\}$$

## 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.

[Website](#)

[Report Abuse](#)

Version: 1.5.2.100

Updated: December 3, 2014

Size: 5.41MB

Language: English



progit2

book

01-introduction

02-git-basics

03-git-branching

04-git-server

05-distributed-git

06-github

callouts

images

sections

1-setting-up-ac

2-contributing.a

3-maintaining.a

4-managing-org

5-scripting.asc

.DS\_Store

1-github.asc

07-git-tools

08-customizing-git

09-git-and-other-scm

10-git-internals

A-git-in-other-enviror

B-embedding-git

C-git-commands

.DS\_Store

contributors.asc

3-maintaining.asc

2,1

Render on save

3-maintaining.asc

3

=== Maintaining a Project

4

5

Now that we're comfortable contributing to a

project, let's look at the other side:

• creating, maintaining and administering your

own project.

6

7

==== Creating a New Repository

8

9

Let's create a new repository to share our

project code with.

• Start by clicking the ``New repository''

button on the right-hand side of the

• dashboard, or from the ``+'' button in the top

toolbar next to your username as seen in

• <<\_new\_repo\_dropdown>>.

10

11

12

.The ``Your repositories'' area.

image::images/newrepo.png[The ``Your

repositories'' area.]

13

14

[[\_new\_repo\_dropdown]]

15

16

.The ``New repository'' dropdown.

image::images/new-repo.png[The ``new

repository'' dropdown.]

17

18

19

This takes you to the ``new repository''

form:

20

21

.The ``new repository'' form.

image::images/newrepoform.png[The ``new

repository'' form.]

22

1. Maintaining a Project

Now that we're comfortable contributing to a project, let's look at the other side: creating, maintaining and administering your own project.

1.1. Creating a New Repository


Let's create a new repository to share our project code with. Start by clicking the "New repository" button on the right-hand side of the dashboard, or from the + button in the top toolbar next to your username as seen in [The "New repository" dropdown.](#)

Your repositories 0


+ New repository

You don't have any repositories yet!

Create your first repository or learn more about Git and GitHub.

schacon

+ -






Figure 1. The "Your repositories" area.

**BUILD AUTOMATION**

# build.gradle

```
buildscript {  
    repositories {  
        jcenter()  
    }  
  
    dependencies {  
        classpath 'org.asciidoctor:asciidoctor-gradle-plugin:1.5.3'  
    }  
}  
  
apply plugin: 'org.asciidoctor.convert'  
  
asciidoctor {  
    attributes toc: 'left', icons: 'font'  
}
```

# issue:<id>[]

```
asciidoctor {
  extensions {
    inlinemacro (name: 'issue') { parent, target, attrs ->
      options = [
        type: ':link',
        target: "http://issues/browse/${target}".toString()
      ]
      createInline(parent, 'anchor',
        target, attrs, options).render()
    }
  }
}
```

# ZEN TECHNIQUES

# SENTENCE PER LINE

— — — — — — — — — — — — — — — .

— — — — — — — — — .

— — — — — — — — — — — .

# COUCH READ



**“CURL YOUR QUOTES”**



# COMMENT IT OUT

```
// ignore this line
```

```
////
```

```
drop
```

```
this
```

```
whole
```

```
section
```

```
////
```

# DRY ATTRIBUTES

```
:uri-project: http://example.org
```

```
Find out more about {uri-project}[project name].
```

# AND INCLUDE FRAGMENTS

## Shared content

```
include::{uri-macros}/how_to_complete_this_guide.adoc[]
```

## Testable source code

```
[source,groovy]
----
include::build.gradle[tag=repositories]
----
```

**AUTHORS ♥ ASCIIDOCTOR**

*“I find going back to Markdown  
akin to trading in utensils and  
eating with my hands.”*

— Greg Turnquist, author of Learning Spring Boot

*“I can do truly amazing things  
with my AsciiDoc source book.”*

— Scott Chacon, author of Pro Git

*“...this has very serious and interesting implications for the publishing industry as a whole.”*

— Scott Chacon, author of Pro Git