

Writing documentation with AsciiDoctor

Jérémie Bresson – 03.11.2015



AsciiDoc

AsciiDoc example

```
= Getting Started with Java  
Author Name <author@example.org>
```

```
== Hello World example
```

```
Copy the HelloWorld.java file.
```

```
TIP: The application prints _Hello World!_ to the console.
```

```
* Compile this source to a class file using `javac`.
```

```
* Run the compiled class file using `java`.
```

```
If you need help with the Java syntax check the  
link:http://docs.oracle.com/javase/8/docs/\[Java 8 Javadoc\].
```

AsciiDoc example

Getting Started with Java

Hello World example

Copy the `HelloWorld.java` file.



The application prints *Hello World!* to the console.

- Compile this source to a class file using `javac`.
- Run the compiled class file using `java`.

If you need help with the Java syntax check the [Java 8 Javadoc](#).

AsciiDoc example

```
= Getting Started with Java  
Author Name <author@example.org>
```

```
== Hello World example
```

```
Copy the HelloWorld.java file.
```

```
TIP: The application prints _Hello World!_ to the console.
```

```
* Compile this source to a class file using `javac`.
```

```
* Run the compiled class file using `java`.
```

```
If you need help with the Java syntax check the  
link:http://docs.oracle.com/javase/8/docs/\[Java 8 Javadoc\].
```

Headings

= Getting Started with Java
Author Name <author@example.com>

Document Title

== Hello World example

Section Level 1

Copy the `*HelloWorld.java*` file.

TIP: The application prints `_Hello World!_` to the console.

* Compile this source to a class file using ``javac``.

* Run the compiled class file using ``java``.

If you need help with the Java syntax check the
link:[http://docs.oracle.com/javase/8/docs/\[Java 8 Javadoc\]](http://docs.oracle.com/javase/8/docs/[Java 8 Javadoc].).

Lists

```
= Getting Started with Java  
Author Name <author@example.org>
```

```
== Hello World example
```

Copy the `*HelloWorld.java*` file.

TIP: The application prints `_Hello World!_` to the console.

```
* Compile Unordered list items file using `javac`.  
* Run the Unordered list items using `java`.
```

If you need help with the Java syntax check the
link:[http://docs.oracle.com/javase/8/docs/\[Java 8 Javadoc\]](http://docs.oracle.com/javase/8/docs/[Java 8 Javadoc]).

Inline formatting

```
= Getting Started with Java  
Author Name <author@example.org>
```

```
== Hello World example
```

```
Copy the *HelloWorld.java* file.
```

```
TIP: The application prints _Hello World!_ to the console.
```

```
* Compile this source to a class file using `javac`.
```

```
* Run the compiled class file using `java`.
```

```
If you need help with the Java syntax check the  
link:http://docs.oracle.com/javase/8/docs/\[Java 8 Javadoc\].
```

Links

```
= Getting Started with Java  
Author Name <author@example.org>
```

```
== Hello World example
```

```
Copy the *HelloWorld.java* file.
```

```
TIP: The application prints _Hello World!_ to the console.
```

```
* Compile this source to a class file using `javac`.
```

```
* Run the compiled class file using `java`.
```

```
If you need help link with title check the
```

```
link: http://docs.oracle.com/javase/8/docs/[Java 8 Javadoc].
```

Admonition

```
= Getting Started with Java  
Author Name <author@example.org>
```

```
== Hello World example
```

Copy the `*HelloWorld.java*` file.

TIP: This application prints `_Hello World!_` to the console.



- * Compile this source to a class file using ``javac``.
- * Run the compiled class file using ``java``.

If you need help with the Java syntax check the
link:[http://docs.oracle.com/javase/8/docs/\[Java 8 Javadoc\].](http://docs.oracle.com/javase/8/docs/[Java 8 Javadoc].)

Asciidoctor

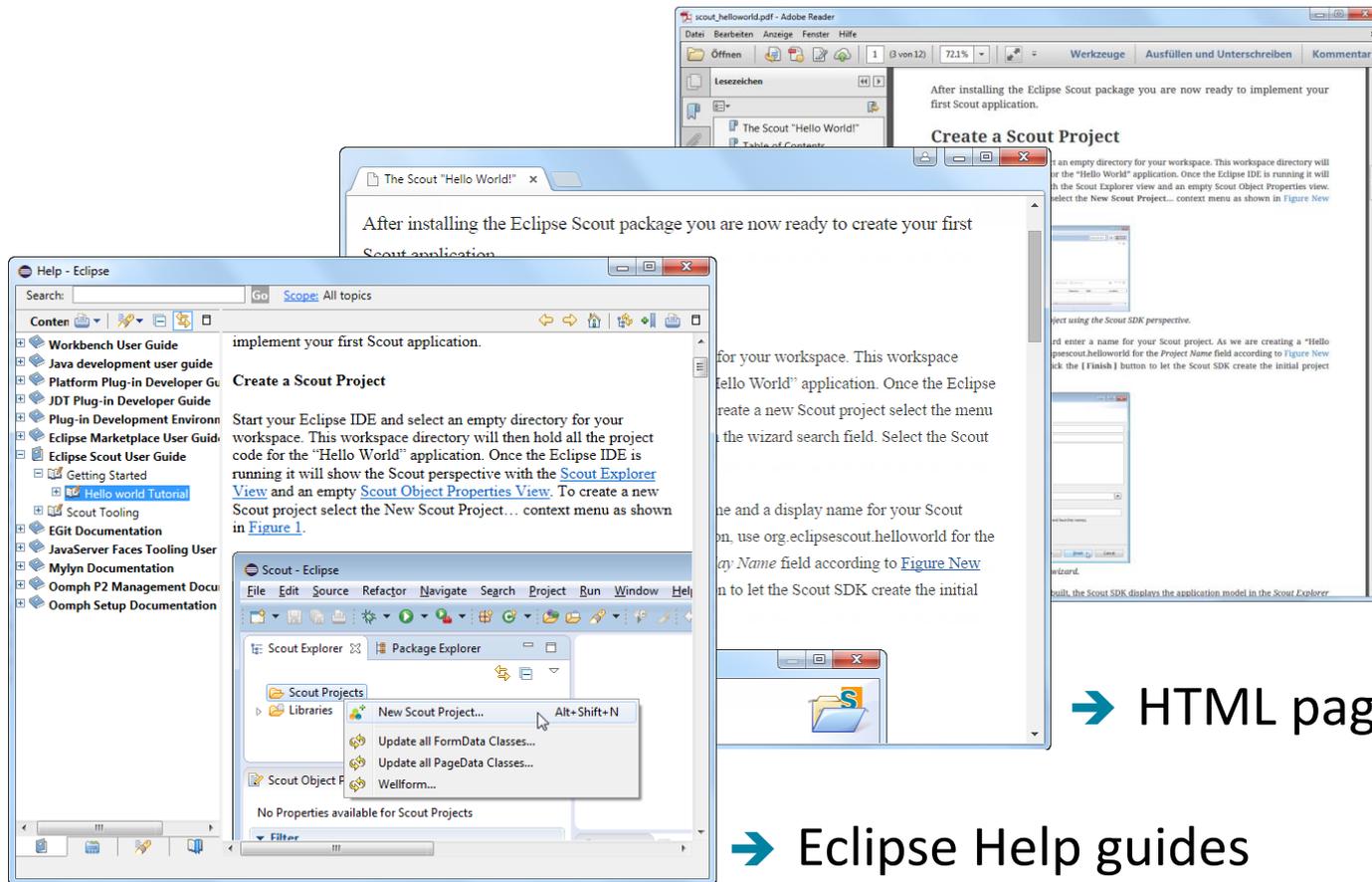
Asciidoctor

- Asciidoctor is **publishing toolchain** for the AsciiDoc writing format.



- Written in **Ruby** (and transpiled to **JavaScript**)
(can be used with JRuby in the **JVM**)
- **Open Source** (MIT Licence)

Multiple Outputs



→ PDF

→ HTML pages

→ Eclipse Help guides

**«Why is asciidoctor a
great tool?»**

Source code listings

A code snippet in your documentation

== Usage of Mylyn wikitext

With the method `toHtml(String)` in `<<lst-wikitext1>>` you can convert a text from Mediawiki Language to HTML.

[[lst-wikitext1, Listing 1]]

[source,java]

```
public static String toHtml(String mediawikiText) {  
    StringWriter writer = new StringWriter();  
    MarkupParser parser = new MarkupParser(new MediaWikiLanguage  
    parser.parse(mediawikiText);  
    return writer.toString();  
}
```

Callout

== Usage of Mylyn wikitext

With the method `toHtml(String)` in `<<lst-wikitext1>>` you can convert a text from Mediawiki Language to HTML.

[[lst-wikitext1, Listing 1]]

[source,java]

```
public static String toHtml(String mediawikiText) {
    StringWriter writer = new StringWriter();
    MarkupParser parser = new MarkupParser(new MediaWikiLanguage
    parser.parse(mediawikiText); // <1>
    return writer.toString();
}
```

<1> Main call of the Mylyn Wikitext framework

Callout

Usage of Mylyn wikitext

With the method `toHtml(String)` in [Listing 1](#) you can convert a text from Mediawiki Language to HTML.

```
public static String toHtml(String mediawikiText) {  
    StringWriter writer = new StringWriter();  
    MarkupParser parser = new MarkupParser(new MediaWikiLanguage(), new HtmlDocumentBuilder(writer));  
    parser.parse(mediawikiText); ❶  
    return writer.toString();  
}
```

- ❶ Main call of the Mylyn Wikitext framework

Callout – Copy paste friendly

Usage of Mylyn wikitext

With the method `toHtml(String)` in [Listing 1](#) you can convert a text from Mediawiki Language to HTML.

```
public static String toHtml(String mediawikiText) {  
    StringWriter writer = new StringWriter();  
    MarkupParser parser = new MarkupParser(new MediaWikiLanguage(), new HtmlDocumentBuilder(writer));  
    parser.parse(mediawikiText); ❶  
    return writer.toString();  
}
```

❶ Main call of the Mylyn Wikitext framework

Real source code in your documentation

- code snippets corresponding to source code under control
 - Always up-to-date
 - Ease maintenance

```
== Usage of Mylyn wikitext
```

```
With the method `toHtml(String)` in <<lst-wikitext1>> you  
can convert a text from Mediawiki Language to HTML.
```

```
[[lst-wikitext1, Listing 1]]
```

```
[source,java]
```

```
----
```

```
include::src/main/java/MediawikiTest.java[tags=method]
```

```
----
```

```
<1> Main call of the Mylyn Wikitext framework
```

Real source code in your documentation

```
public class MediawikiTest {  
    //tag::method[]  
    public static String toHtml(String mediawikiText) {  
        StringWriter writer = new StringWriter();  
        MarkupParser parser = new MarkupParser(new MediaWikiLanguage(  
            parser.parse(mediawikiText); // <1>  
        return writer.toString();  
    }  
    //end::method[]  
    @Test  
    public void test() {  
        StringBuilder sb = new StringBuilder();  
        sb.append("= Heading 1 =\n");  
        sb.append("\n");  
        sb.append("Hello World!\n");  
    }  
}
```

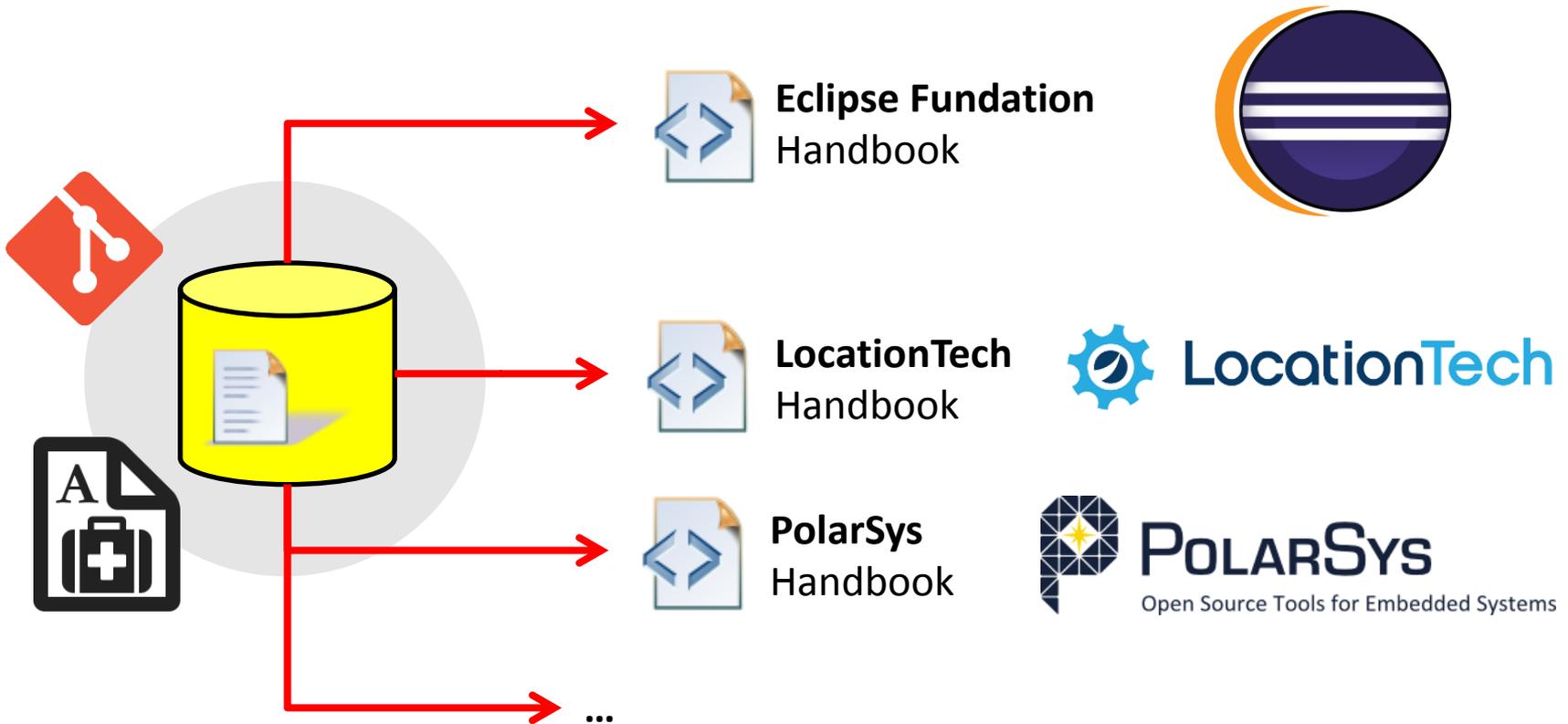
Single source your documents

**«AsciiDoc lets me do
variable substitution and
conditional inclusion»**

Wayne Beaton, Eclipse Foundation

<https://waynebeaton.wordpress.com/2015/08/10/new-eclipse-project-handbook/>

One source, multiple output



Variable substitution

```
:forgeName: Eclipse  
:bugzillaUrl: https://bugs.eclipse.org/bugs
```



Eclipse Project Handbook

=====

```
include::chapters/resources.adoc[]
```

eclipse.adoc

Open source projects at the Eclipse Foundation are required to make use of certain Eclipse Foundation services:

* All project issues must be tracked in the
{bugzillaUrl}[{forgeName} Bugzilla]instance;

resources.adoc

Variable substitution

```
:forgeName: LocationTech  
:bugzillaUrl: https://www.locationtech.org/bugs
```



```
LocationTech Project Handbook
```

```
=====
```

```
include::chapters/resources.adoc[]
```

```
locationtech.adoc
```

Open source projects at the Eclipse Foundation are required to make use of certain Eclipse Foundation services:

* All project issues must be tracked in the
{bugzillaUrl}[{forgeName} Bugzilla]instance;

```
resources.adoc
```

Conditional inclusion

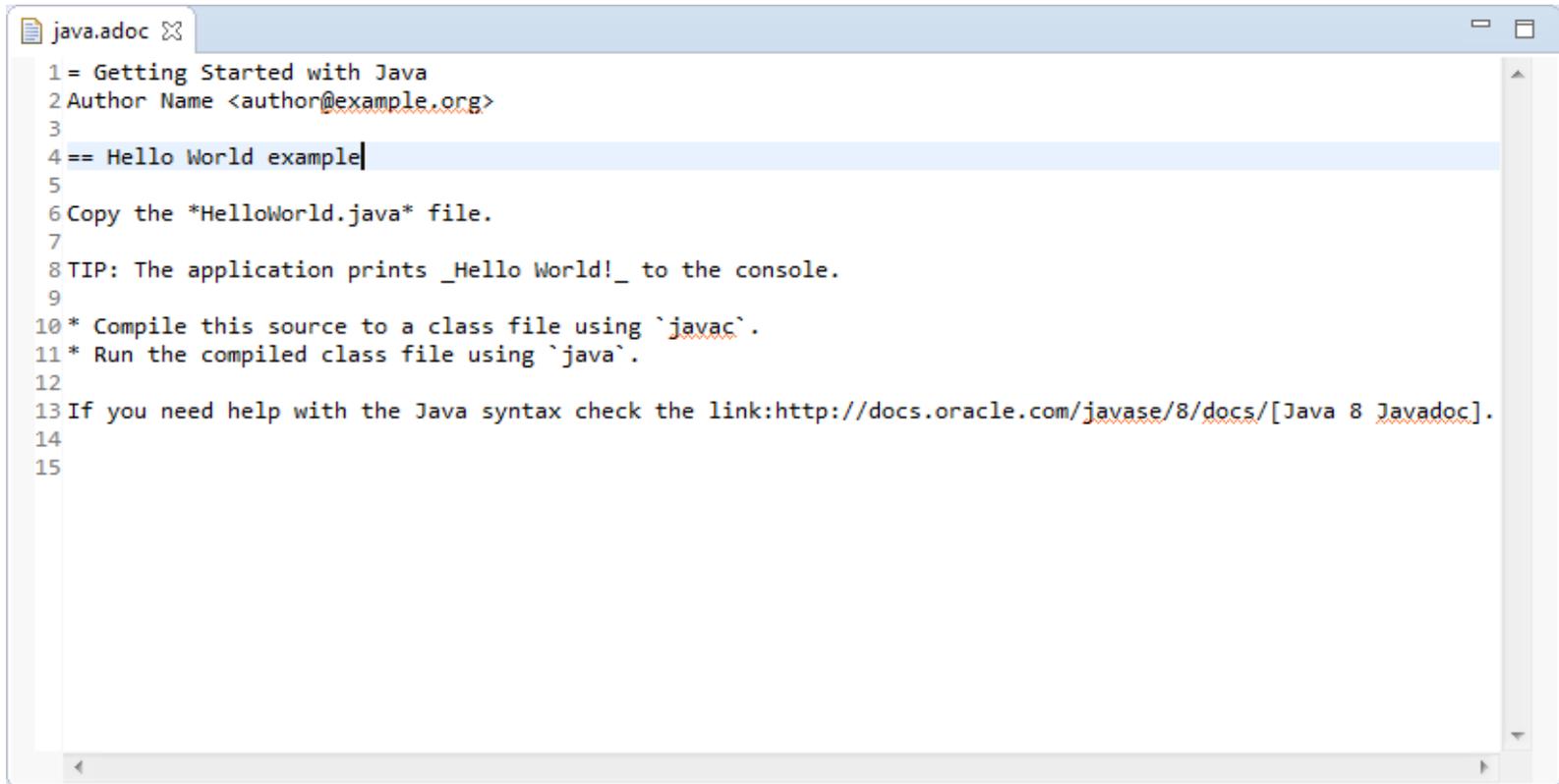
```
ifeval::["{forgeName}"=="Eclipse"]
```

```
_The Eclipse Foundation strongly encourages all projects to  
create an  
maintain and http://marketplace.eclipse.org[Eclipse  
Marketplace]  
presence._
```

```
endif::[]
```

Simple Tools (no setup)

Any text editor

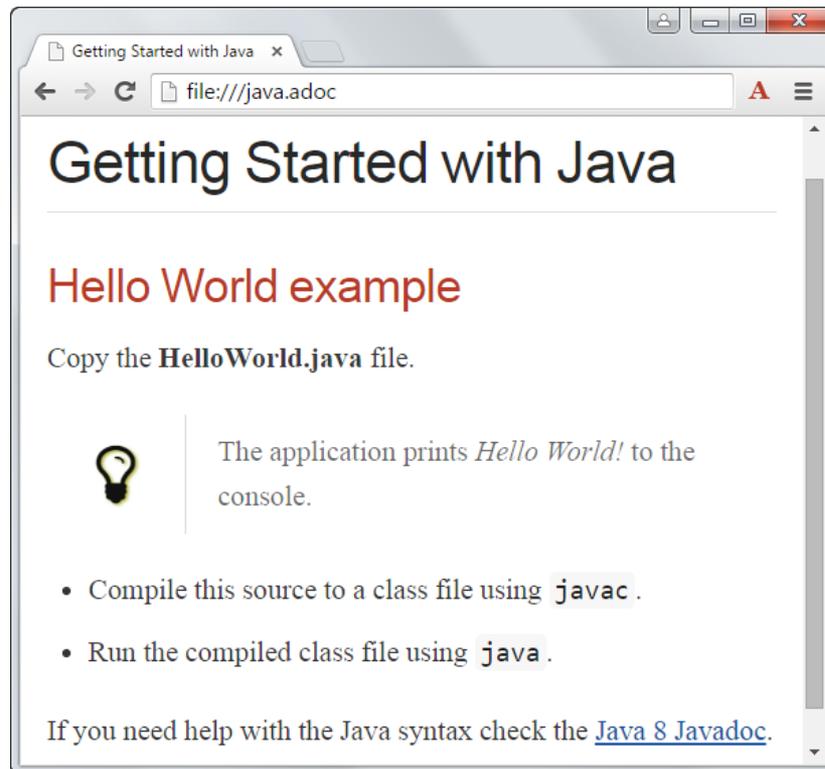


A screenshot of a text editor window titled "java.adoc". The window contains the following text:

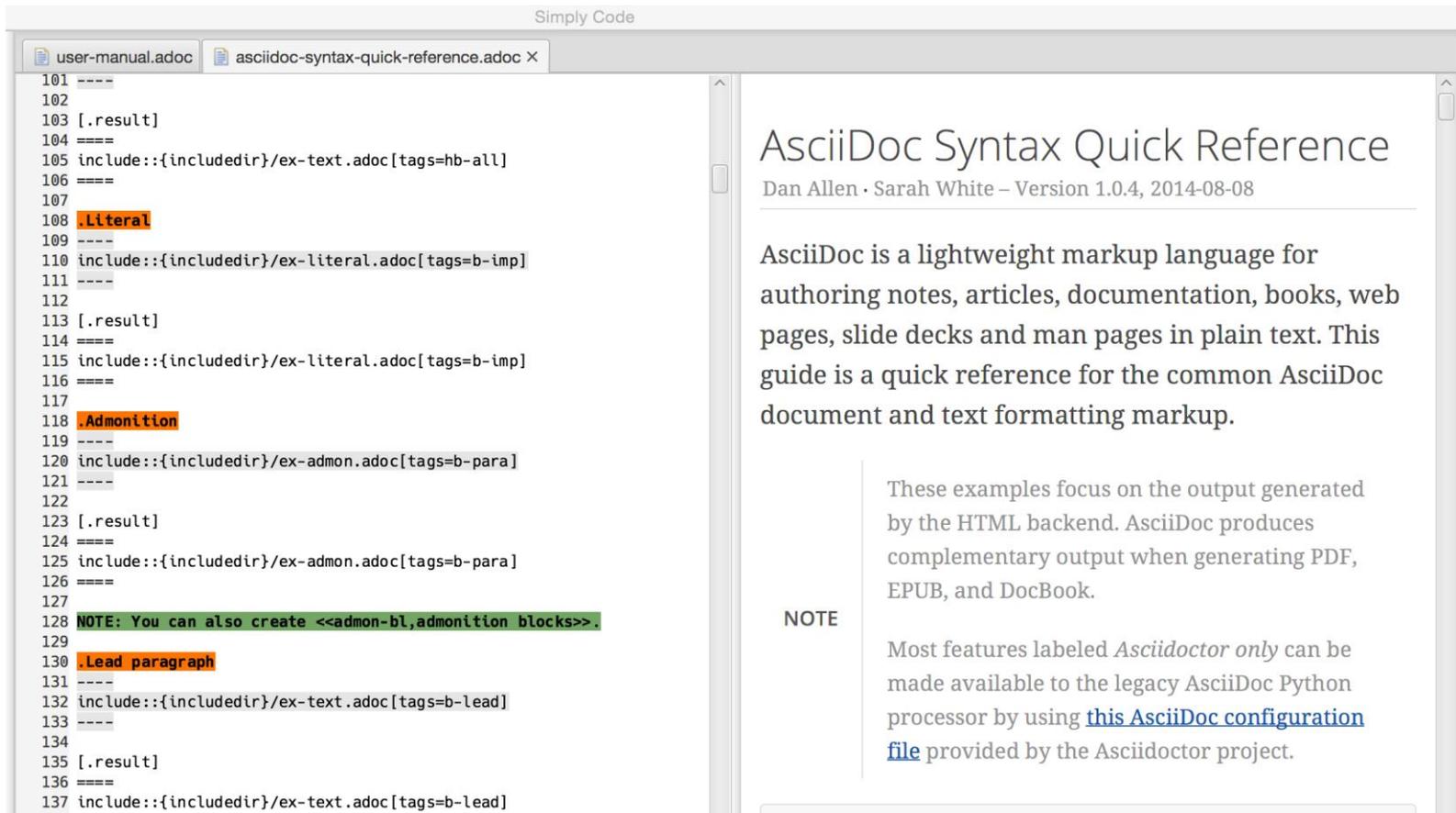
```
1 = Getting Started with Java
2 Author Name <author@example.org>
3
4 == Hello World example|
5
6 Copy the *HelloWorld.java* file.
7
8 TIP: The application prints _Hello World!_ to the console.
9
10 * Compile this source to a class file using `javac`.
11 * Run the compiled class file using `java`.
12
13 If you need help with the Java syntax check the link:http://docs.oracle.com/javase/8/docs/\[Java 8 Javadoc\].
14
15
```

Live preview in Chrome

→ Extension for Chrome: **Asciidoctor.js Live Preview**



JavaFX Codeeditors – Tom Schindl



The screenshot displays a JavaFX application window titled "Simply Code". It features two tabs: "user-manual.adoc" and "asciidoc-syntax-quick-reference.adoc". The left pane shows the source AsciiDoc code, and the right pane shows the rendered HTML output.

```
101 ----
102
103 [.result]
104 ====
105 include::{includedir}/ex-text.adoc[tags=hb-all]
106 ====
107
108 .Literal
109 ----
110 include::{includedir}/ex-literal.adoc[tags=b-imp]
111 ----
112
113 [.result]
114 ====
115 include::{includedir}/ex-literal.adoc[tags=b-imp]
116 ====
117
118 .Admonition
119 ----
120 include::{includedir}/ex-admon.adoc[tags=b-para]
121 ----
122
123 [.result]
124 ====
125 include::{includedir}/ex-admon.adoc[tags=b-para]
126 ====
127
128 NOTE: You can also create <<admon-bl,admonition blocks>>.
129
130 .Lead paragraph
131 ----
132 include::{includedir}/ex-text.adoc[tags=b-lead]
133 ----
134
135 [.result]
136 ====
137 include::{includedir}/ex-text.adoc[tags=b-lead]
---
```

AsciiDoc Syntax Quick Reference

Dan Allen · Sarah White – Version 1.0.4, 2014-08-08

AsciiDoc is a lightweight markup language for authoring notes, articles, documentation, books, web pages, slide decks and man pages in plain text. This guide is a quick reference for the common AsciiDoc document and text formatting markup.

These examples focus on the output generated by the HTML backend. AsciiDoc produces complementary output when generating PDF, EPUB, and DocBook.

NOTE

Most features labeled *AsciiDoctor only* can be made available to the legacy AsciiDoc Python processor by using [this AsciiDoc configuration file](#) provided by the AsciiDoctor project.

Build

**«Generating the
documentation with
Asciidoctor is always a
straightforward
experience»**

Julien Ponge, Eclipse Golo project

<https://julien.ponge.org/blog/an-experiment-in-maven-to-gradle-migration/>

Build system

- Ruby, JavaScript, JVM environments
- Integration with a lot of build systems:
 - Maven
 - Gradle
- Dependencies available in several repositories
 - Maven central
 - JCenter
 - NPM
- You can build on every machine, without a complex setup

Edit on GitHub

Edit on GitHub

The image shows a browser window displaying the Microsoft Azure documentation page for Redis Cache. The page title is "How to use Azure Redis Cache". The breadcrumb navigation shows "DOCUMENTATION > REDIS CACHE". The main heading is "How to use Azure Redis Cache". The author is Steve Danielson, last updated on 10/23/2015. The page features a "Contributors" section with a row of profile pictures and a prominent "Edit on GitHub" link, which is circled in orange. Below the contributors, there are tabs for ".Net", "Node.js", "Java", and "Python", with "Java" selected. The page also includes a "Plan" section and a "Common scenarios" section. The browser's address bar shows the URL "https://azure.microsoft.com/en-us/documentation/...", and the page has a security warning from Microsoft Corporation (US).

Microsoft Azure

Why Azure Products Documentation Pricing Partners Blog Resources Support

SALES 0800-180-8941 MY ACCOUNT PORTAL Search

Redis Cache

DOCUMENTATION > REDIS CACHE

How to use Azure Redis Cache

By Steve Danielson
Last updated: 10/23/2015

Contributors **Edit on GitHub**

.Net Node.js **Java** Python

In this article:
[Prerequisites](#)
[Create a Redis cache on Azure](#)

GitHub Editor

The screenshot displays the GitHub Editor interface. At the top, there is a navigation bar with the GitHub logo, a search box for the repository, and links for Pull requests, Issues, and Gist. On the right side of the navigation bar, there are notification and user profile icons. Below the navigation bar, the repository information is shown: BSI-Business-Systems-Integration-AG / org.eclipse.scout.docs, with options to Unwatch (22), Star (0), and Fork (2). The main content area shows the file path org.eclipse.scout.docs / docs / adoc / HelloWorldFrontend.adoc and a button to cancel the edit. The editor window has tabs for 'Edit file' and 'Preview changes', along with settings for Spaces (2) and Soft wrap. The code editor shows a document with line numbers 17 through 32. Line 19 is highlighted, showing the text: 'We will add widgets to the client's desktop form that will later display the "Hello World!" message.' The code includes several paragraphs of text and a code block for an image placeholder: `<<img-new_field_context_menu>>`. The document ends with instructions on how to use the form field wizard.

This repository Search

Pull requests Issues Gist

BSI-Business-Systems-Integration-AG / org.eclipse.scout.docs
forked from eclipse/org.eclipse.scout.docs

Unwatch 22 Star 0 Fork 2

org.eclipse.scout.docs / docs / adoc / HelloWorldFrontend.adoc or cancel

Edit file Preview changes Spaces 2 Soft wrap

```
17
18 The project creation wizard has created a Scout client that contains an empty desktop form.
19 We will add widgets to the client's desktop form that will later display the "Hello World!" message.
20
21 To add any widgets to the desktop form, navigate to the [element]_DesktopForm_ in the Scout Explorer.
22 For this, click on the orange client node in the Scout Explorer view.
23 Then, expand the [element]_Forms_ folder by clicking on the small triangle icon, and further expand the
   [element]_DesktopForm_.
24 As a result, the [element]_MainBox_ element becomes visible below the desktop form as shown in
   <<img-new_field_context_menu>>.
25 With a click of the right mouse button over the [element]_MainBox_, the available context menus are displayed.
26 To start the form field wizard select the menu:New Form Field ...[] menu.
27
28 [[img-new_field_context_menu, Figure New Form Field Menu]]
29 .Using the menu:New Form Field ...[] menu to start the form field wizard provided by the Scout SDK.
30 image::{imgsdire}/sdk_new_field_wizard_menu.png[]
31
32 In the first step of the form field wizard shown on in <<img-helloworld_groupboxfield>> choose [java]+GroupBox+ as the
   form field type and click on the btn:[Next] button.
```

GitHub Preview

The screenshot shows a GitHub repository page for `BSI-Business-Systems-Integration-AG / org.eclipse.scout.docs`. The repository is forked from `eclipse/org.eclipse.scout.docs`. The page displays the file `HelloWorldFrontend.adoc` in the `docs / adoc` directory. The preview mode is active, showing the content of `_initDoc.adoc`. The text in the preview is:
The project creation `stepwizard` has created a Scout client that `displays` contains an empty desktop form. We will `now` add widgets to the client's desktop form that will later display the "Hello World!" message.
To add any widgets to the desktop form, navigate to the `DesktopForm` in the Scout Explorer. For this, click on the orange client node in the Scout Explorer view. Then, expand the `Forms` folder by clicking on the small triangle icon, and further expand the `DesktopForm`. As a result, the `MainBox` element becomes visible below the desktop form as shown in [Figure New Form Field Menu](#). With a click of the right mouse button over the `MainBox`, the available context menus are displayed. To start the form field wizard select the `New Form Field ...` menu.
At the bottom of the preview, a snippet of an IDE interface is visible, showing the `Scout Explorer` and `Package Explorer` views.

Thanks

/@j2r2b

/+JeremieBresson

/jmini

jeremie.bresson@bsi-software.com