

# AsciiDoctor

Dokumentation schreiben  
kann Spass machen!



# Sebastian Hempel

IT-Consulting Hempel  
@ithempel



# Welche Dokumente schreiben Entwickler?

**PROGRAM**

Form No. 400

0001 J.V. KEOGH

330434 FINCHLEY MANOR MILL.

EB A level COMPUTER SCIENCE  
Course work.



Warum macht das  
Schreiben  
keinen Spass?





Was muss anders  
werden?





# AsciiDoctor





# AsciiDoc

leightweight markup language

erste Veröffentlichung im November 2002

# AsciiDoctor



Implementierung von AsciiDoc in Ruby  
erste Veröffentlichung im Januar 2013



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

# direkte Erstellung von HTML5



# Ausgabe im DocBook Format



Dan Allen  
@mojavelinux



# Wer nutzt AsciiDoc(tor)



# Formatierung

```
*fett*  
_kursiv_  
`nicht proportional`
```

# Listen

```
* Element  
** nächstes Ebene  
* weiteres Element
```



# Checklisten

- [ ] noch nicht erledigt
- [\*] erledigt

# Aufzählung

- . erstes Element
- . zweites Element
- .. erstes Unterelement

# Labeled Lists

Label:: Beschreibung  
noch ein Label:: eine weitere Beschreibung

# Zitate

[quote, Bill Gates, Microsoft]

640 KB are enough for everyone.



# Literal Block

```
....  
Jetzt wird alles  
  wie angegeben  
-  ausgegeben.  
....
```

# Code Blöcke

```
----  
public class EnterpriseAbstractFactory {  
    public doSomething(int howLong) {  
        Thread.sleep(howLong);  
    }  
}  
----
```

# Callouts

```
----  
public class EnterpriseAbstractFactory { <1>  
    public doSomething(int howLong) {  
        Thread.sleep(howLong); <2>  
    }  
}
```

----  
<1> to shoort

<2> busy waiting please

# Codeformatierung

```
[source,java]
----
public class EnterpriseAbstractFactory {

    public doSomething(int howLong) {
        Thread.sleep(howLong);
    }
}
----
```



# Codeformatierung ohne Zeilenumbruch

```
[source,java,options="nowrap"]
----
public class EnterpriseAbstractFactory {

    public doSomething(int howLong) {
        if (checkSomethingThatLeadsToAVeryLongLine() == WE_EXPECT_EXACTLY_THIS) {
            Thread.sleep(howLong);
        }
    }
}
----
```

# Tabellen

```
|===  
| Kopfzeile | mit zweiter Spalte  
  
| JDK | 8  
| Java EE | 7  
|===
```

# Tabellen aus CSV

```
[format="csv",options="header"]  
|===  
Operation System,Software,Version  
Linux,JDK,8  
NoArch,WildFly,8.1  
|===
```

# Einbinden von Bildern

`image::filename.png[A Picture,200,100]`

# Festlegung des Bilderverzeichnisses

```
:imagesdir: ./img
```



# Inhaltsverzeichnis

```
:toc:  
:toclevels: 3  
:toc-title: Inhaltsverzeichnis  
:toc-placement!:  
:sectanchors:  
:numbered:
```

```
toc::[]
```

# Integration in Build-Tool

***maven***

 **gradle**

# Integration des Plugins

```
...  
<plugins>  
  <plugin>  
    <groupId>org.asciidoctor</groupId>  
    <artifactId>asciidoctor-maven-plugin</artifactId>  
    <version>0.1.4</version>  
    ...  
  </plugin>  
</plugins>  
...
```

# Konfiguration des Plugins

```
<plugin>
  ...

  <configuration>
    <sourcedirectory>${basedir}/src/main/doc</sourcedirectory>
    <outputdirectory>${basedir}/target/docs</outputdirectory>
    <backend>html</backend>
    <doctype>book</doctype>
  </configuration>
  ...
</plugin>
...
```

# Aufruf des Plugins

```
...  
<plugin>  
  ...  
  <executions>  
    <execution>  
      <id>output-html</id>  
      <phase>generate-resources</phase>  
      <goals>  
        <goal>process-asciidoc</goal>  
      </goals>  
    </execution>  
  </executions>  
  ...  
</plugin>  
...
```

# Integration in JavaDoc

AsciiDoc statt HTML



# Integration in JavaDoc-Erstellung

```
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-javadoc-plugin</artifactId>
  <version>2.9</version>
  <configuration>
    <source>1.7
    <doclet>org.asciidoctor.Asciidoctor</doclet>
    <docletartifact>
      <groupId>org.asciidoctor</groupId>
      <artifactId>asciidoctor</artifactId>
      <version>0.1.4</version>
    </docletartifact>
  </configuration>
</plugin>
```

# Kommentar mit AsciiDoc

```
/** = Example Class
 *
 * This ist an example class.
 *
 * * This is a List
 *
 * This is bold or italic.
 */
public class Example {

    private String attribute;

    /**
     * Get some attribute.
     *
     * null:: The value might be null.
     * other:: The name of the attribute.
     */
}
```

# AsciiDoctor Plugins



# Beispiel AsciiDoctor-Diagram

- Graphviz
- PlantUML
- Dita

# Installation

```
gem install asciidoctor-diagram
```



# Integration

```
#!/usr/bin/ruby

require 'asciidoctor'
require 'asciidoctor-diagram/plantuml'

Asciidoctor.render_file('sample.adoc',
  :in_place => true, :safe => 'unsafe')
```



# Integration ab 1.5.x

Vorraussetzung: Java (JAVA\_HOME)

```
asciidoctor -r asciidoctor-diagram sample.adoc
```

# Beispiel

```
["plantuml", "asciidoctor-diagram-classes", "png"]  
----  
interface BlockProcessor  
class DiagramBlock  
class DitaaBlock  
class PlantUmlBlock  
class GraphvizBlock  
  
BlockProcessor <|-- DiagramBlock  
DiagramBlock <|-- DitaaBlock  
DiagramBlock <|-- PlantUmlBlock  
DiagramBlock <|-- GraphvizBlock  
----
```

# Bildnachweis

- (1) Sebastian Hempel / Sebastian Hempel / CC-BY SA
- (2) document folders / John Keogh / CC-BY NC
- (3) frustration / Sybren Stüvel / CC-BY NC
- (4) There are years that ask questions and years that answer. /  
theunquietlibrarian / CC-BY NC
- (5) Day 9 / Jay Reed / CC-BY SA
- (8) Dan Allen / Dan Allen
- (18) puzzle / Olga Berrios / CC-BY

# Creative Commons CC-BY SA

