

# Oto dokument



to dokument pokazujący potęgę Adoca



uważaj, nie będziesz taki sam po ujrzeniu tego dokumentu



warte uwagi: <https://docs.oracle.com/javase/7/docs/technotes/tools/windows/javadoc.html#see> linki działają



ważne - w części o URLach jest dokumentacja

## Asciidoctor

światny sposób na doktoryzowanie się w znaczkach Ascii. Ale można konwertować do HTMLa i PDFa i szeregu innych.

## Mark<sub>down</sub>

coś co konwertujemy do mark<sup>up</sup> - czyli znaczników HTMLa. Markdown, Ricka Grubera chyba dzieło.

### Przykład markdown

```
## Nagłówek z Markdown

'''
to jest blok kodu
'''

**Powyższe** *zostało* osiągnięte jak w Markdown.

Lista w Markdown:

1. punkt pierwszy
1. punkt drugi
1. punkt trzeci
1. punkt czwarty
1. punkt piąty
1. punkt szósty
```

## Spis Treści

Nagłówek z Markdown .....	2
Pokazówka .....	2
Bibliography / Links .....	3
Prawdziwa potęga z obsługą kodu! .....	4
Linie z numerami? .....	4

Wyróżnienia .....	4
Demo zmiennych .....	5
Proste formatowanie tekstu .....	5
URLe .....	6
Proste formatowanie tekstu .....	6
Listy .....	6
Zastosowania .....	6
Kod .....	7
Z tytułem i składnią .....	7
Blok kodu tak po prostu, czcionką mono .....	8
Bloki .....	8
Zagnieżdżone bloki .....	9
Inne? .....	10
Cytaty .....	10
Prezentacja .....	10
cytat .....	10
transition .....	11
transition .....	11
transition .....	11
tło .....	11
tło okładowe .....	12

## Nagłówek z Markdown


```
to jest blok kodu
```

**Powyższe zostało** osiągnięte jak w Markdown.

Lista w Markdown:

1. punkt pierwszy
2. punkt drugi
3. punkt trzeci
4. punkt czwarty
5. punkt piąty
6. punkt szósty

## Pokazówka

- 4  @Olayinka The answer depends, to some degree, on the question "what is a getter" -- there's a fuzzy line between "getter" and "method that returns something." Obviously, there are useful cases for returning an `Optional` from methods. But in real-world code (for better or worse), most getters (like, 99%) are just `return x`. In that case, returning an `Optional` from a getter means you are using `Optional` as your state representation -- and this is definitely not what was intended (and falls squarely in the "zealous overuse" category.) – Brian Goetz Jul 22 '16 at 18:18

## TERM

Definition of it, `inline code` works here too

Quote text accepts **bold**, *italics*, ~~--strikethrough--~~

— quote source

[This is a text that appears instead of an image if path or extension are wrong] | *pathTo.png*

A link if with http: <http://www.computerhope.com/unix/dmesg.htm>

*Title of a code snippet:*

```
Code snippet
Is between
Two lines of 4 dashes
Like so:
MCi_ADDR register valid
Processor context corrupt
MCA: Data CACHE Level-1 Data-Read Error
STATUS be00000000200135 MCGSTATUS 5
CPUID Vendor Intel Family 6 Model 58
RIP: _raw_spin_lock+0x12/0x50}
SOCKET 0 APIC 1 microcode 15
```

*A word of warning from the wise:*

The man page for `vmstat` describes "r" as "The number of processes waiting for run time", which is either incorrect or misleading (on recent Linux distributions it's reporting those threads that are waiting, and threads that are running on-CPU; it's just the wait threads in other OSes).

— Brendan Gregg, <http://www.brendangregg.com/USEmethod/use-rosetta.html>

## Bibliography / Links

Besides link already used and added earlier, few others caught my eye:



[Peter Lawrey's one-liner for poor-man's profiler... in Java code itself!](#)



[Pierre Hugues Charbonneau on Java Code Geeks presents thread dump analysis](#)



Bug report turned doc enhancement request, on SIGHUP handling by JVM



If you started process in the background and killed it's terminal, process might get killed. If it ain't, then  
to [redirect it's output is no easy task](#). Still, it's feasible.

## Prawdziwa potęga z obsługą kodu!

*to jest potęga!*

```
public int foo() {  
    int result = 1;  
    try {  
        return result;  
    } finally {  
        result++;  
    }  
}  
  
System.out.println("Result of foo(): " + foo());
```

## Linie z numerami?

*linie kodu mogą mieć atrybut inline lub table by kopiowanie działało z nimi lub bez*

```
1    public int foo() {  
2        int result = 1;  
3        try {  
4            return result;  
5        } finally {  
6            result++;  
7        }  
8    }  
9  
10   System.out.println("Result of foo(): " + foo());
```



domyślnie przy podświetlaczu składni pygments jest `:pygments-linenumbs-mode=table`, co daje opcje z lub bez, zależy co zaznaczysz



powyższy przykład ma też wybrane linie, które chce załączyć, ale można dać wybrane sekcje

## Wyróżnienia

Jeśli chcesz zwrócić czyjąś uwagę na konkretne rzeczy...

```
1 ③
2 class Test { ①
3
4     public static void main(String[] args) {
5         ②
6     }
7 }
```

- ① deklaracja klasy z domyślnym specyfikatorem dostępu
- ② tu kod, jaki będziemy wykonywać, obecnie apka nic nie robi, włączy się i się zamknie
- ③ tu Javadoc, dokumentacja techniczna mówiąca na co ta klasa jest i jak jej użyć

## Demo zmiennych



Zmienne są zadeklarowane w głównym pliku.

Agent {pseudonim} zimno rzekł:

> {dobrze}, nie {złe}

Agent 007 zimno rzekł:

wstrząśnięte, nie zmieszane

## Proste formatowanie tekstu

**pogrubienie**

**pogrubione też?**

*kursywa*

*kursywa*

-wykreślenie-

i trochę zwykłego tekstu

z pustymi liniami pomiędzy

by zobaczyć

jak to wygląda

Dobłą praktyką jest nazywanie wątków

A mutable reduction operation accumulates input elements into a mutable result container, such as a Collection or StringBuilder, as it processes the elements in the stream.

```
String concatenated = strings.reduce("", String::concat)
```

## URLe

<https://asciidoctor.org/docs/asciidoc-syntax-quick-reference/>  
Dokumentacja ADOCa

[Podsumowanie pakietu strumieni w JDK8 Oracle'a](#)

## Proste formatowanie tekstu

**pogrubienie**

*kursywa*

**wykreślenie**

i trochę zwykłego tekstu

z pustymi liniami pomiędzy

by zobaczyć  
jak to wygląda

Dobłą praktyką jest nazywanie wątków

## Listy

Bywają numerowane i punktowane. Robi się je przez . albo przez \*.

## Zastosowania

1. dokumentacja
2. maszynowe tworzenie HTMLa
  - a. statyczne strony
  - b. prezentacje

### 3. opis repozytoriów

- obsługiwane przez wiele platform
  - GitHub
  - GitLab
  - ale np. nie BitBucket, OIDP (sprawdźcie sobie)



listy można zagnieżdżać

1. punkt 1
2. punkt 2
3. punkt 3
  - a. punkt 3.1., mam nadzieję
    - i. dalsze zagnieżdżenie
4. punkt 4

Bez tej linii listy zostaną scalone, ona wprowadza nowy paragraf:

- oto
- lista
- jest

## Kod

Kod można wprowadzić pisząc czcionką mono, czyli o stałej szerokości, np. `String s = new String();`, nie mamy wtedy podświetlania składni ale możemy taki kod ująć w tekście.

Kod blokowo można osiągnąć Tabem:

```
blok czegokolwiek, pisany czcionką o stałej szerokości
kod też przejdzie
jest wieloliniowy
```

Albo ---- oznaczając tak początek i koniec bloku:

```
to też jest blok czcionką mono pisany
```

## Z tytułem i składnią

To **może** być kolorowane, jeśli czegoś do tego użyjemy. Adoc nie ma wbudowanego silnika do kolorowania składni.

częste polecenie hakera ;-)

```
cat /etc/passwd
```

koordynaty z POMa

```
<groupId>pl.lafk</groupId>
<artifactId>tomkat</artifactId>
<version>0.1</version>
<name atrybut="a">tomkat</name>
<description>Wiosna z buta</description>
<url>lafk.pl</url>
```

## Bloki kodu tak po prostu, czcionką mono

```
[tammo@lafk-T470 adoc.demo]$ asciidoctor plik.adoc [INSERT]
asciidoctor: WARNING: plik.adoc: line 18: section title out of sequence: expected
level 2, got level 3
[tammo@lafk-T470 adoc.demo]$ asciidoctor plik.adoc [INSERT]
[tammo@lafk-T470 adoc.demo]$ asciidoctor plik.adoc [INSERT]
[tammo@lafk-T470 adoc.demo]$ asciidoctor plik.adoc [INSERT]
[tammo@lafk-T470 adoc.demo]$
```

## Bloki



*obrazek czytany z netu jest groźny*

Nie tylko obrazek zresztą. By Twój adoc miał cokolwiek z netu musisz uruchomić go z parametrem `-a allow-uri-read`. Upewnij się, że to dobry pomysł, zwykle nie.

[Dan Allen - obrazek z netu] |

`<em>https://avatars1.githubusercontent.com/u/79351?s=88&u=d0e91997d1ba984f82a009663a26e5890420e425&v=4</em>`



Dan jest twórcą Asciidoktora.



```
Feedback_Diagra... Tytuł — Mozilla F... Q3AP_Pytania_W... Microsoft Teams ... Notatki - Notatki Learn - Chromium Signal
/home/tammo/tm
vim plik.adoc
plik.adoc ~/b/m/s/x.adoc X
1 :source-highlighter: pygments
2 :icons: font
3 :toc: preamble
4 :toc-title: Spis Treści
5 :author: Tomasz J. Borek
6 = Tytuł
7
8 Mark-down - coś co konwertujemy do mark-up - czyli znaczników H
  TMLa. Markdown, Ricka Grubera chyba dzieło.
9
10 == Bloki
11
12 image::https://avatars1.githubusercontent.com/u/79351?s=88&u=d0
  e91997d1ba984f82a009663a26e5890420e425&v=4[Dan Allen]
13
14 [IMPORTANT]
15 .Thread lifecycle
16 =====
17 *Enum: `java.lang.Thread.State`:*
18
19 NEW:: Thread state for a thread which has not yet started.
20
21 RUNNABLE:: Thread state for a runnable thread.
plik.adoc 11,0-1 Góra
"plik.adoc" 134L, 2401C zapisano
/home/tammo/tmp/adoc.demo
-q, --quiet          suppress warnings (default: false)
--trace             include backtrace information on errors (default: false)
-v, --verbose        enable verbose mode (default: false)
-t, --timings        enable timings mode (default: false)
-h, --help [TOPIC]  print the help message
                   show the command usage if TOPIC is not specified (or not recognized)
                   dump the Asciidoctor man page (in troff/groff format) if TOPIC is manpage
-V, --version        display the version and runtime environment (or -v if no other flags or argument
s)
[tammo@lafk-T470 adoc.demo]$ asciidoctor -a allow-uri-read plik.adoc [INSERT]
[tammo@lafk-T470 adoc.demo]$ [INSERT]
```

Figure 1. A teraz z dysku

[Tego obrazka nie ma - zamiast wyświetla się tekst w nawiasach kwadratowych] |  
nieMaTakiego.png

## Zagnieżdżone bloki



*Thread lifecycle*

Enum: `java.lang.Thread.State`:

## NEW

Thread state for a thread which has not yet started.

## RUNNABLE

Thread state for a runnable thread.



*Nested blocks are OK if more = signs are in*

`_RUNNABLE_` means OK, right?

`Thread.getState()` gives you that.

## Inne?



blokami są też bloki kodu, cytaty i nagłówki (sekcje) dokumentu.

## Cytaty

cytat tak po prostu w bloku

cytat

wieloliniowy

cytat

wieloliniowy

— Tomek powiedział, że jego tata powiedział

Cytat po prostu

## Prezentacja

testy `reveal.js`

## cytat

This is one of my favorite books on programming, although I burned through it in about 3 or 4 hours, and didn't learn a huge amount from it. The point is that it was fun to read, it made a lot of sense, and it was helpful in reinforcing the habits I knew were good ones.

The primary focus of the book is on how to be an effective and efficient programmer, and how to have fun practicing your craft. So, for instance, they'll recommend that you learn how to touch-type with all ten of your fingers. And that you choose a powerful extensible editor and learn how to write extensions for it. And that you use a programming language that supports metaprogramming, so you can code your way out of holes without using hundreds of thousands of lines of code.

## **transition**

how is it? zoom

## **transition**

how is it? slide

## **transition**

how is it? fade-in

## **tlo**

# Who wants to become a manager?



tło okładowe

# Who wants to become a manager?

