canoo
[delivering end-user happiness]

Make your Asciidoctor Groovy

Stephan Classen

About the speaker

Stephan Classen

- » TDD enthusiast
- » Loves Open Source
- » Hates repetitive jobs

About the speaker

Stephan Classen

- » TDD enthusiast
- » Loves Open Source
- » Hates repetitive jobs
- » A little paranoid

About the speaker

stephan.classen@canoo.com

github.com/sclassen/greach2016

The Screenshot Extension

- » Original idea and Proof of concept by François-Xavier Thoorens
- » Use GEB to control the browser and take screenshots of your application

Terminology

» AsciiDoc

– A mature, plain-text writing format for authoring documentation, books and more.

» Asciidoctor

– A fast text processor and publishing toolchain for converting AsciiDoc files.

Advertising
advertisement
internet
radio
selling
customer
trademark
commercial
online
mobile
media
banner
brand
marketing
newspapers
billboards
concept
ads

Little Formatting mostly Content

== Parameters

=== screenshot

The _screenshot_ block macro has the syntax `screenshot::<url>[<parameters>]` and supports the following parameters (note: the URL is not needed when the browser was navigated to the page using a _geb_ block or other means)

name:: an optional unique file name (will be generated otherwise), the screenshot will be called `<name>.png`. This is the first positional parameter.
frame:: Try awesomeness with `iphone5`, `nexus5` or `browser`. This is the second positional parameter.
dimension:: size of the screenshot in the format `<width>x<height>` for instance 800x600. Also the values of 'frame' are supported.
selector:: the CSS-like dom selector to screenshot. Only this will be part of the image. For instance `div #login_window`.

Note: it is not allowed to set both 'frame' and 'dimension' or 'frame' and 'selector' for a screenshot.
Note: `screenshot::http://google.com[google]` and `screenshot::http://google.com[name=google]` are equivalent.

=== geb

The _geb_ block supports the following parameters:

dimension:: size of the screenshot in the format `<width>x<height>` for instance 800x600. Also the values of 'frame' are supported. This is the first positional parameter.

Little Formatting mostly Content

== Parameters

=== screenshot

The _screenshot_ block macro has the syntax `screenshot::<url>[<parameters>]` and supports the following parameters (note: the URL is not needed when the browser was navigated to the page using a _geb_ block or other means)

name:: an optional unique file name (will be generated otherwise), the screenshot will be called `<name>.png`. This is the first positional parameter.
frame:: Try awesomeness with `iphone5`, `nexus5` or `browser`. This is the second positional parameter.
dimension:: size of the screenshot in the format `<width>x<height>` for instance 800x600. Also the values of 'frame' are supported.
selector:: the CSS-like dom selector to screenshot. Only this will be part of the image. For instance `div #login_window`.

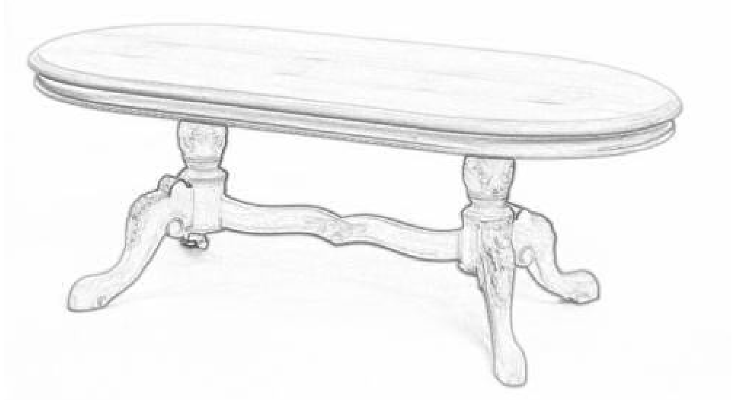
Note: it is not allowed to set both 'frame' and 'dimension' or 'frame' and 'selector' for a screenshot.
Note: `screenshot::http://google.com[google]` and `screenshot::http://google.com[name=google]` are equivalent.

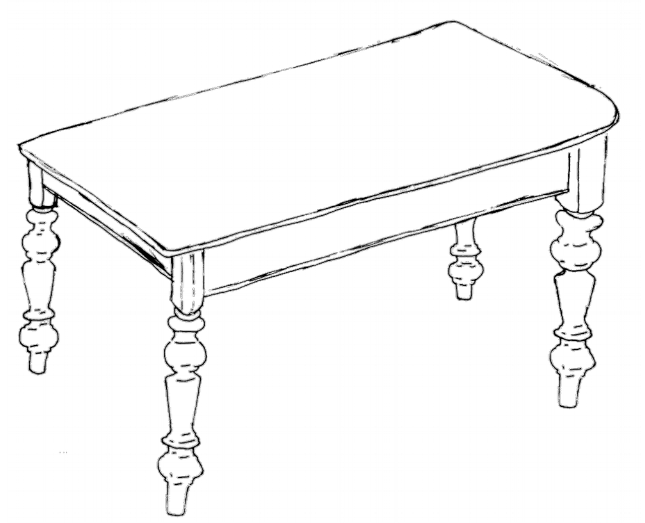
=== geb

The _geb_ block supports the following parameters:

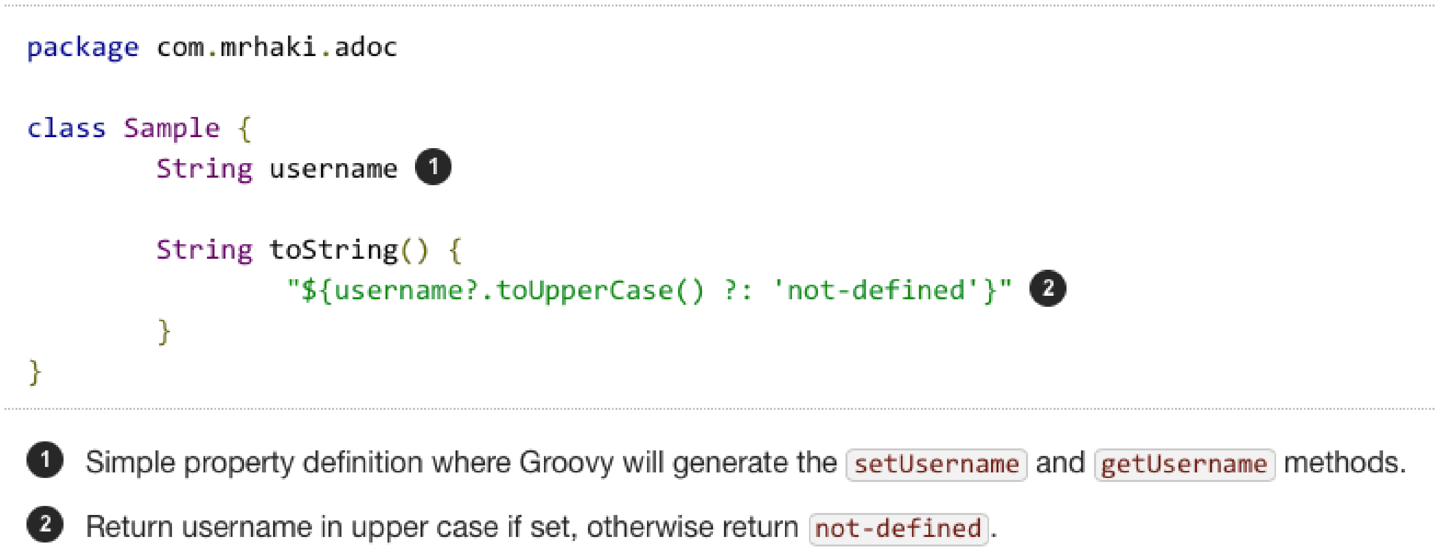
dimension:: size of the screenshot in the format `<width>x<height>` for instance 800x600. Also the values of 'frame' are supported. This is the first positional parameter.

Tables





Syntax Highlighting and Callouts



```
package com.mrhaki.adoc

class Sample {
        String username ❶

        String toString() {
                "${username?.toUpperCase() ?: 'not-defined'}" ❷
        }
}
```

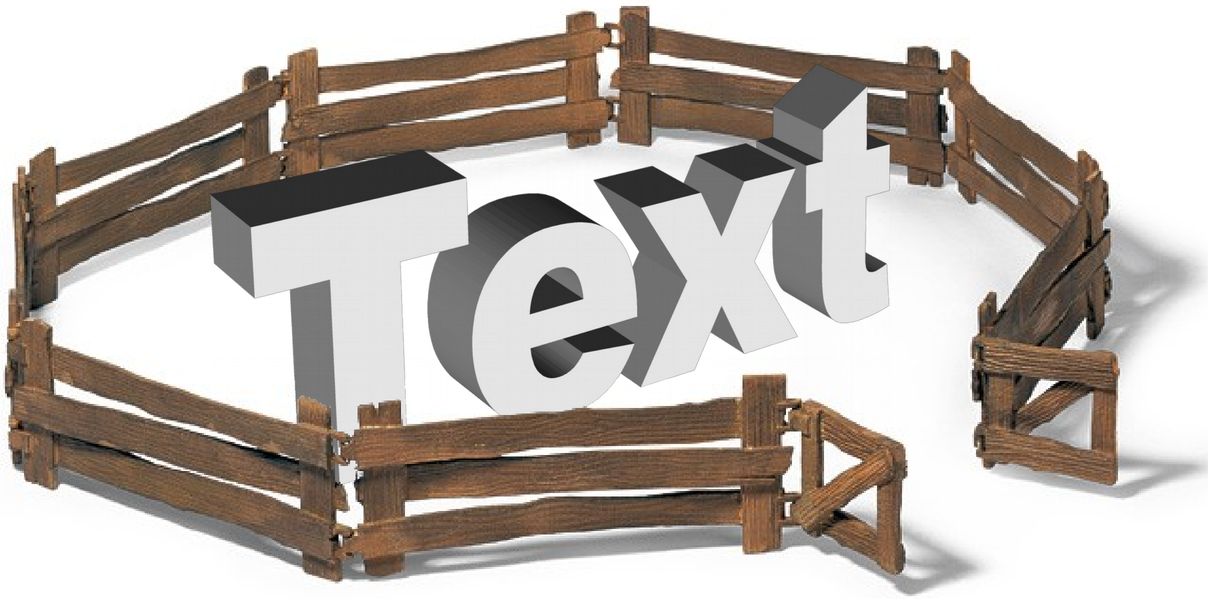
❶ Simple property definition where Groovy will generate the `setUsername` and `getUsername` methods.

❷ Return username in upper case if set, otherwise return `not-defined`.

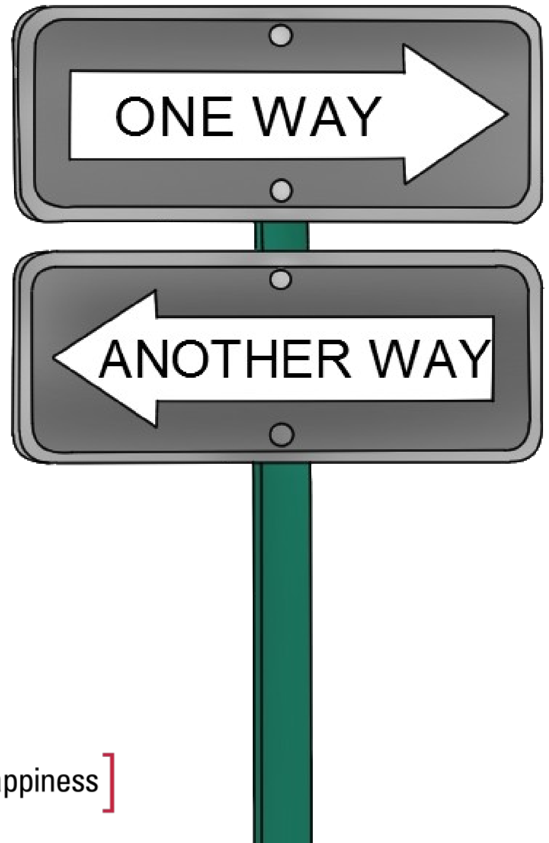
Includes

Attributes (Variables)

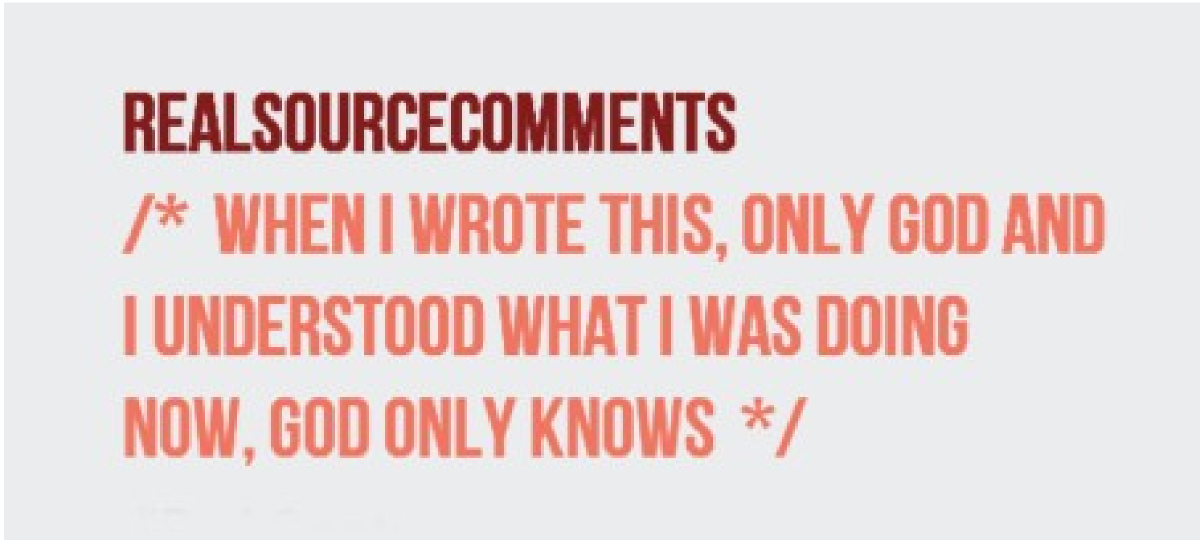
Fenced Content



Conditionals



Comments



REALSOURCECOMMENTS

/* WHEN I WROTE THIS, ONLY GOD AND I UNDERSTOOD WHAT I WAS DOING NOW, GOD ONLY KNOWS */

Output Formats (Backends)

HTML 5

DOCBOOK 5

PDF

ePUB

$\LaTeX$

Runtimes

Build Tools

GRUNT

Gradle

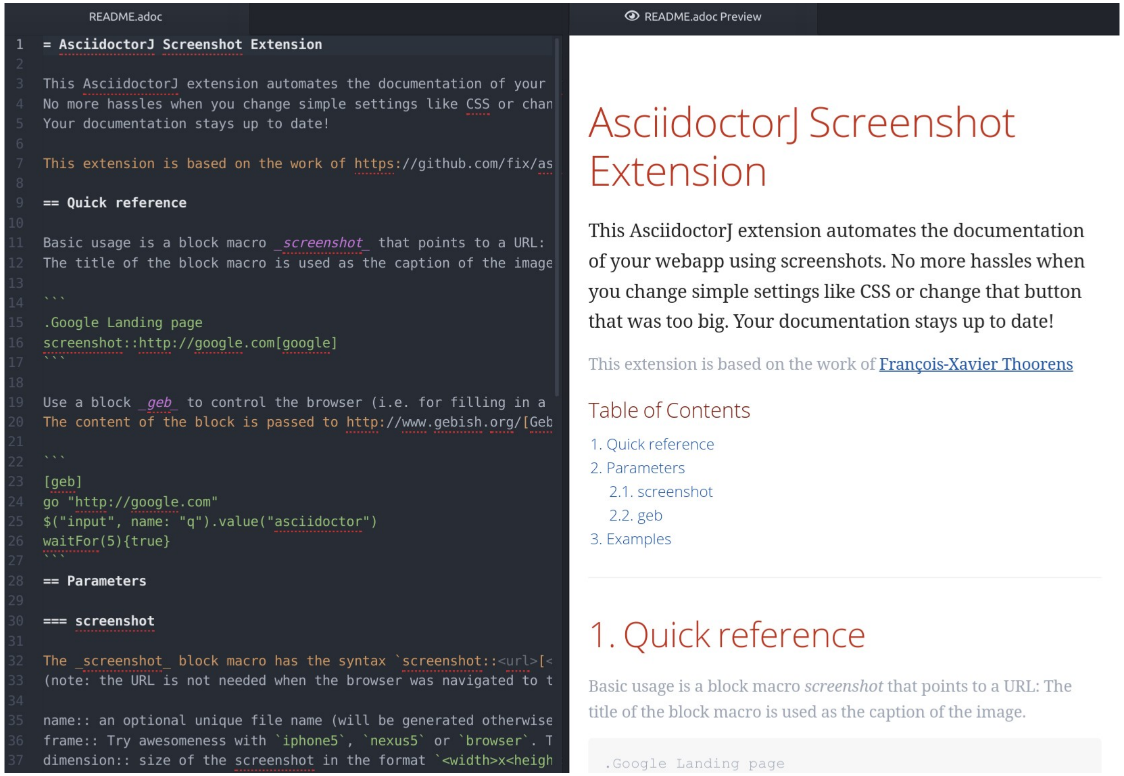
<APACHE ANT>

Gulp

maven

Extensions

Instant Preview



README.adoc

~~~~
= AsciidoctorJ Screenshot Extension

This AsciidoctorJ extension automates the documentation of your
No more hassles when you change simple settings like CSS or chan
Your documentation stays up to date!

This extension is based on the work of https://github.com/fix/as

== Quick reference

Basic usage is a block macro _screenshot_ that points to a URL:
The title of the block macro is used as the caption of the image

```
.Google Landing page
screenshot::http://google.com[google]
```


Use a block _geb_ to control the browser (i.e. for filling in a
The content of the block is passed to http://www.gebish.org/[Geb

```
[geb]
go "http://google.com"
$("input", name: "q").value("asciidoctor")
waitFor(5){true}
```
== Parameters

=== screenshot

The _screenshot_ block macro has the syntax `screenshot::<url>[<
(note: the URL is not needed when the browser was navigated to t

name:: an optional unique file name (will be generated otherwise
frame:: Try awesomeness with `iphone5`, `nexus5` or `browser`. T
dimension:: size of the screenshot in the format `<width>x<heigh
~~~~

README.adoc Preview

AsciidoctorJ Screenshot Extension

This AsciidoctorJ extension automates the documentation of your webapp using screenshots. No more hassles when you change simple settings like CSS or change that button that was too big. Your documentation stays up to date!

This extension is based on the work of François-Xavier Thoorens

Table of Contents

1. Quick reference
2. Parameters
   2.1. screenshot
   2.2. geb
3. Examples

1. Quick reference

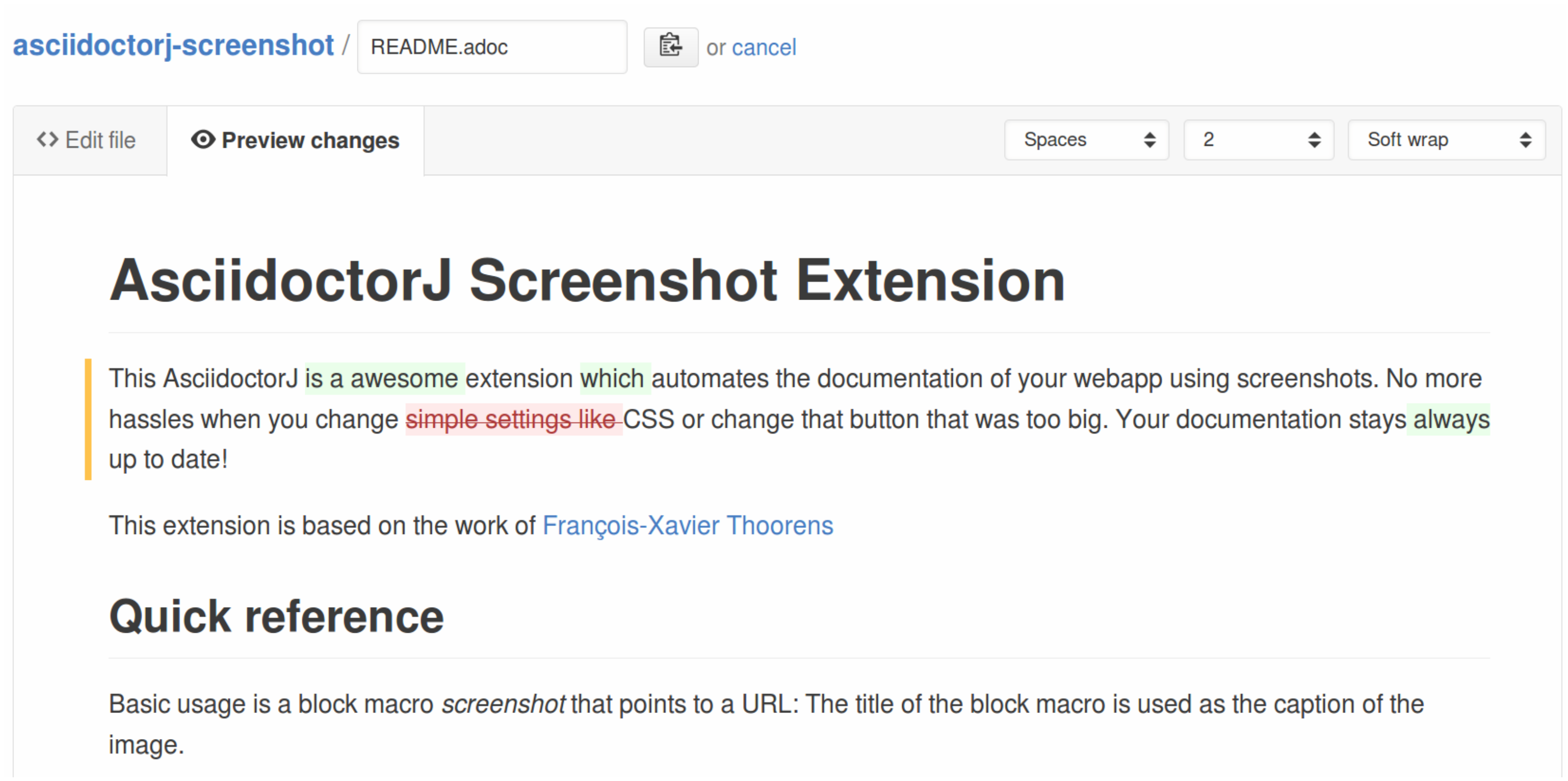
Basic usage is a block macro *screenshot* that points to a URL: The title of the block macro is used as the caption of the image.

.Google Landing page

[delivering end-user happiness]

canoo

Github Diff Preview



asciidoctorj-screenshot / README.adoc

AsciidoctorJ Screenshot Extension

This AsciidoctorJ is a awesome extension which automates the documentation of your webapp using screenshots. No more hassles when you change ~~simple settings like~~ CSS or change that button that was too big. Your documentation stays always up to date!

This extension is based on the work of François-Xavier Thoorens

Quick reference

Basic usage is a block macro *screenshot* that points to a URL: The title of the block macro is used as the caption of the image.

Who uses Asciidoctor

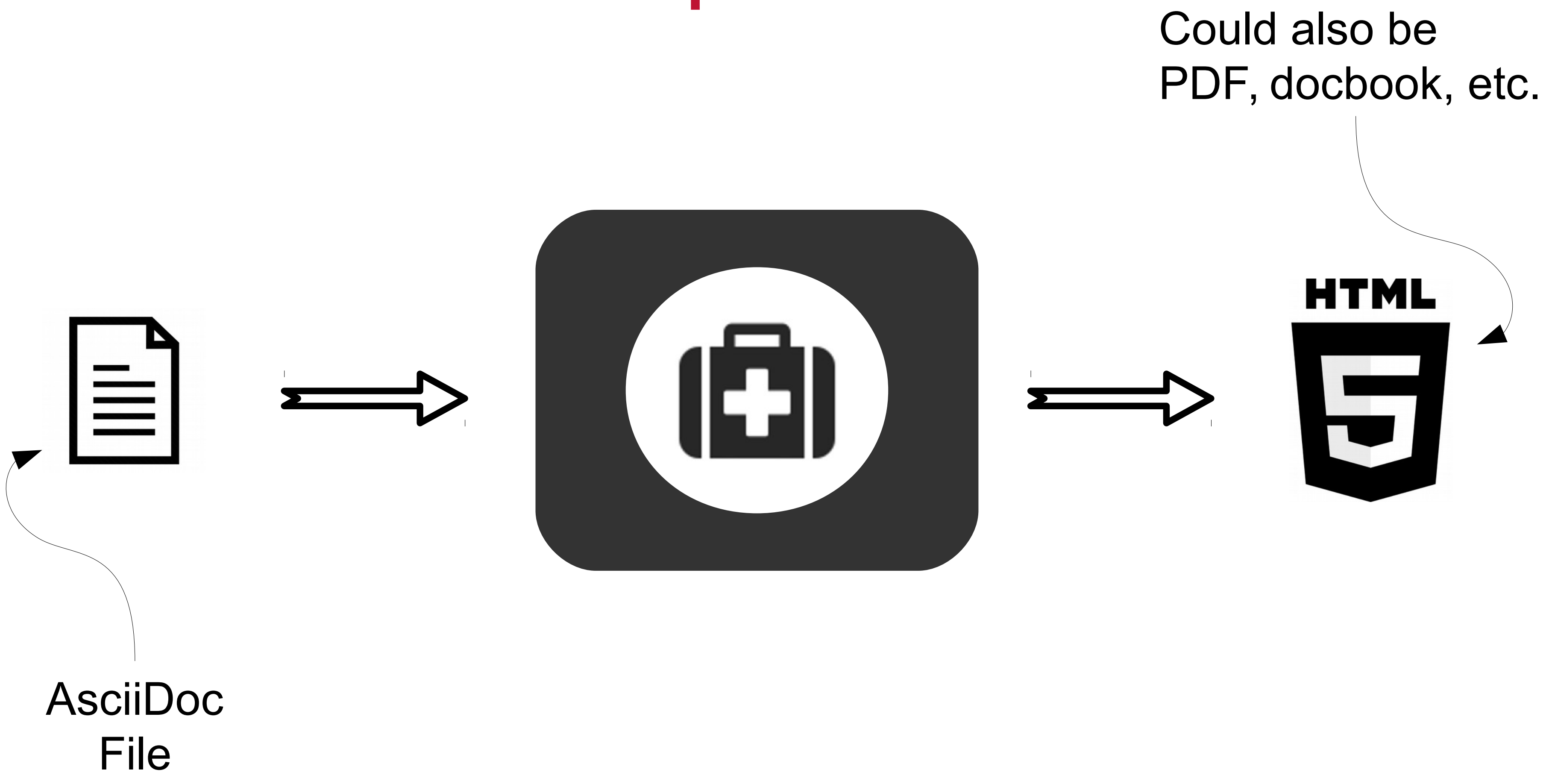
O'REILLY®

Gradle

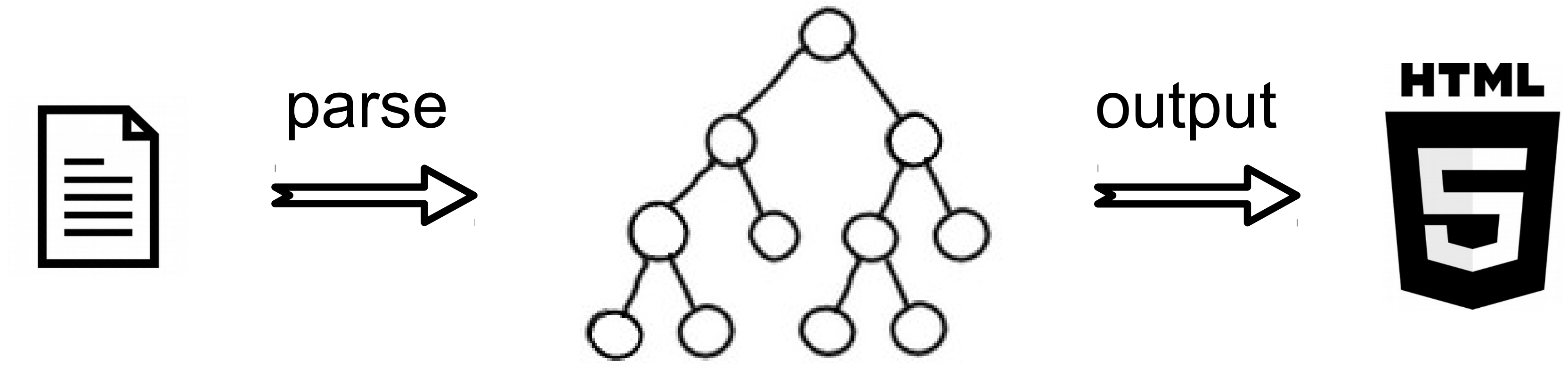
git

spring

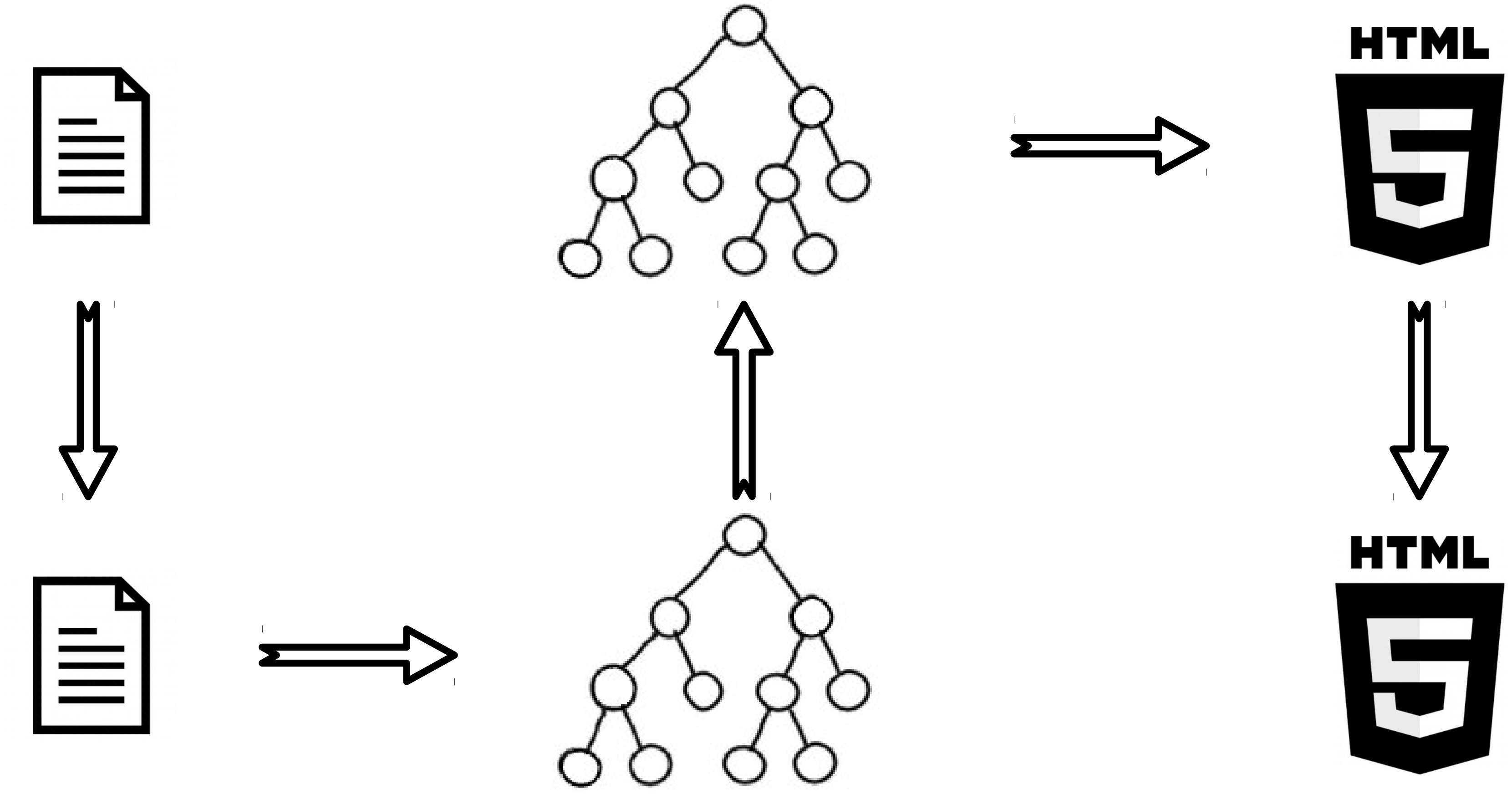
Asciidoctor conversion process



Asciidoctor conversion process

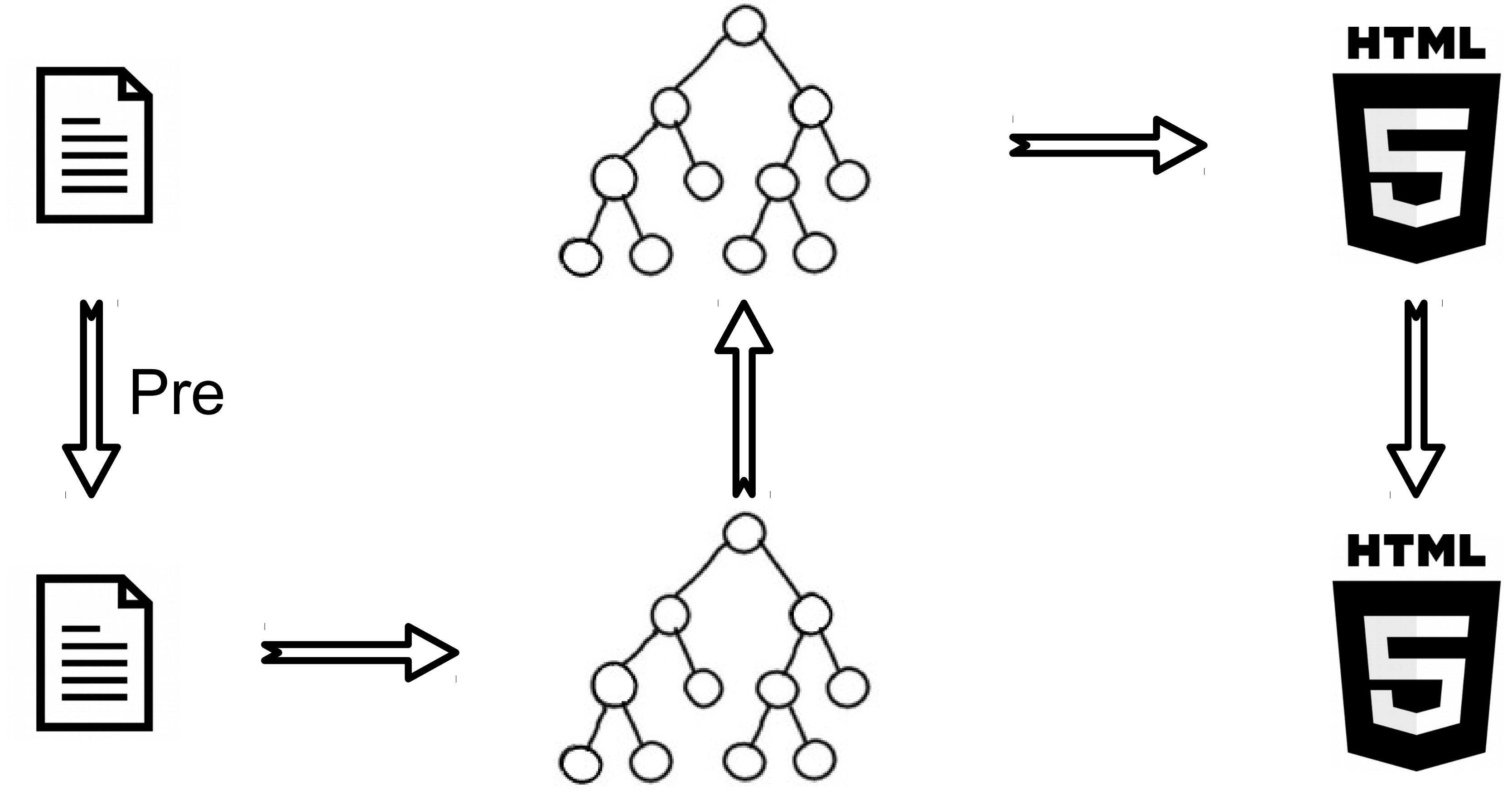


Extending the conversion process

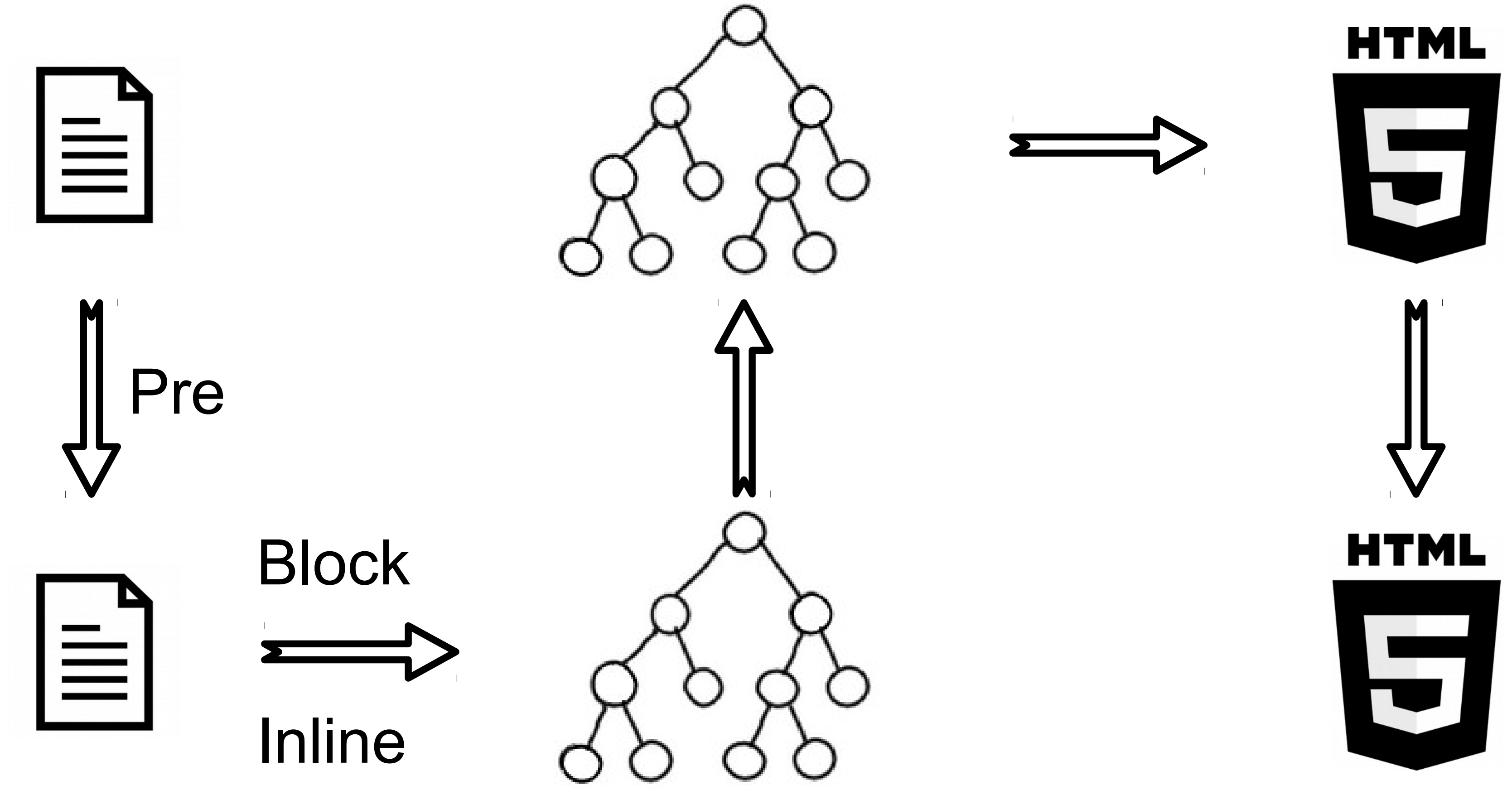


canoo

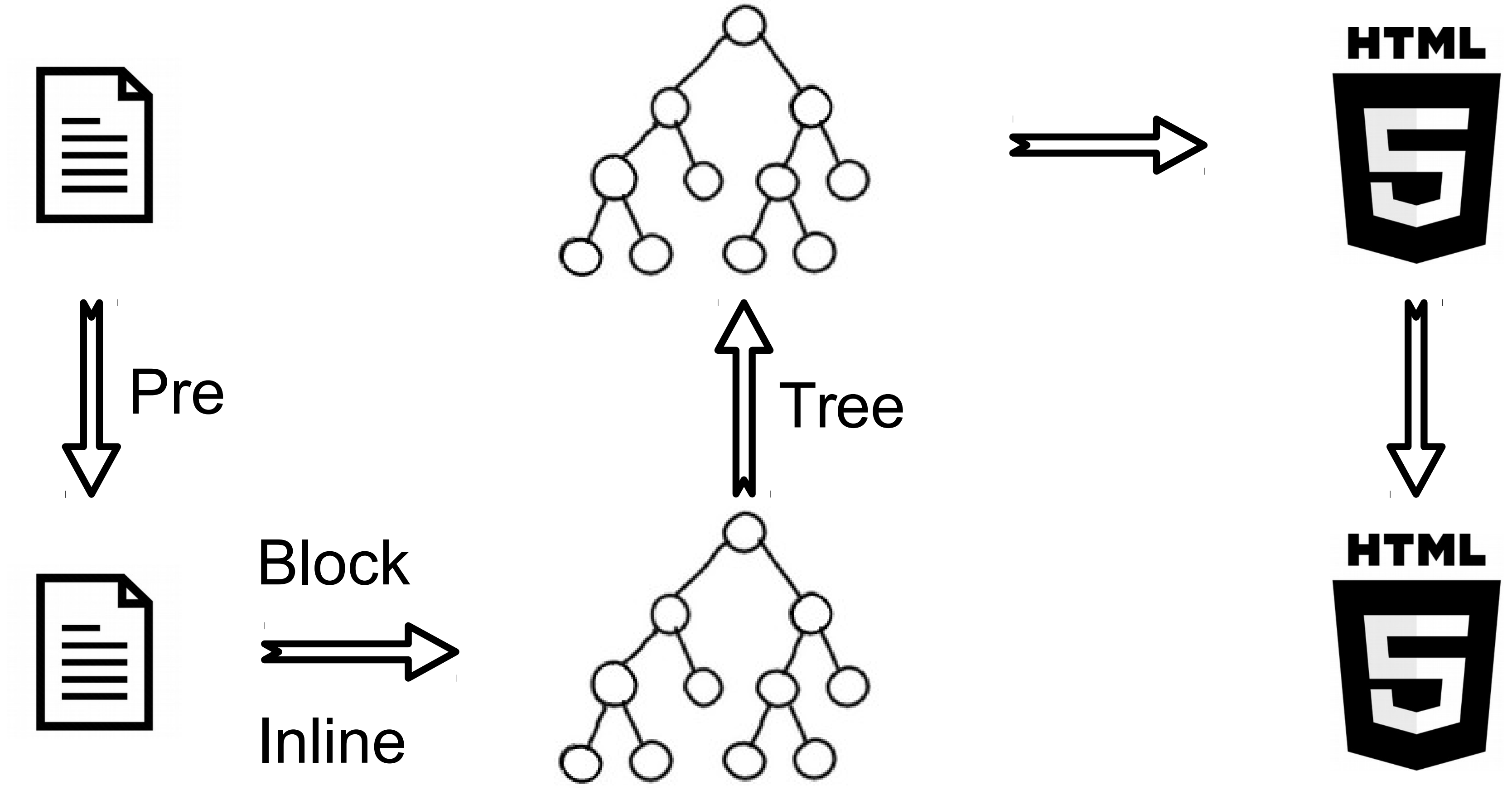
Extending the conversion process



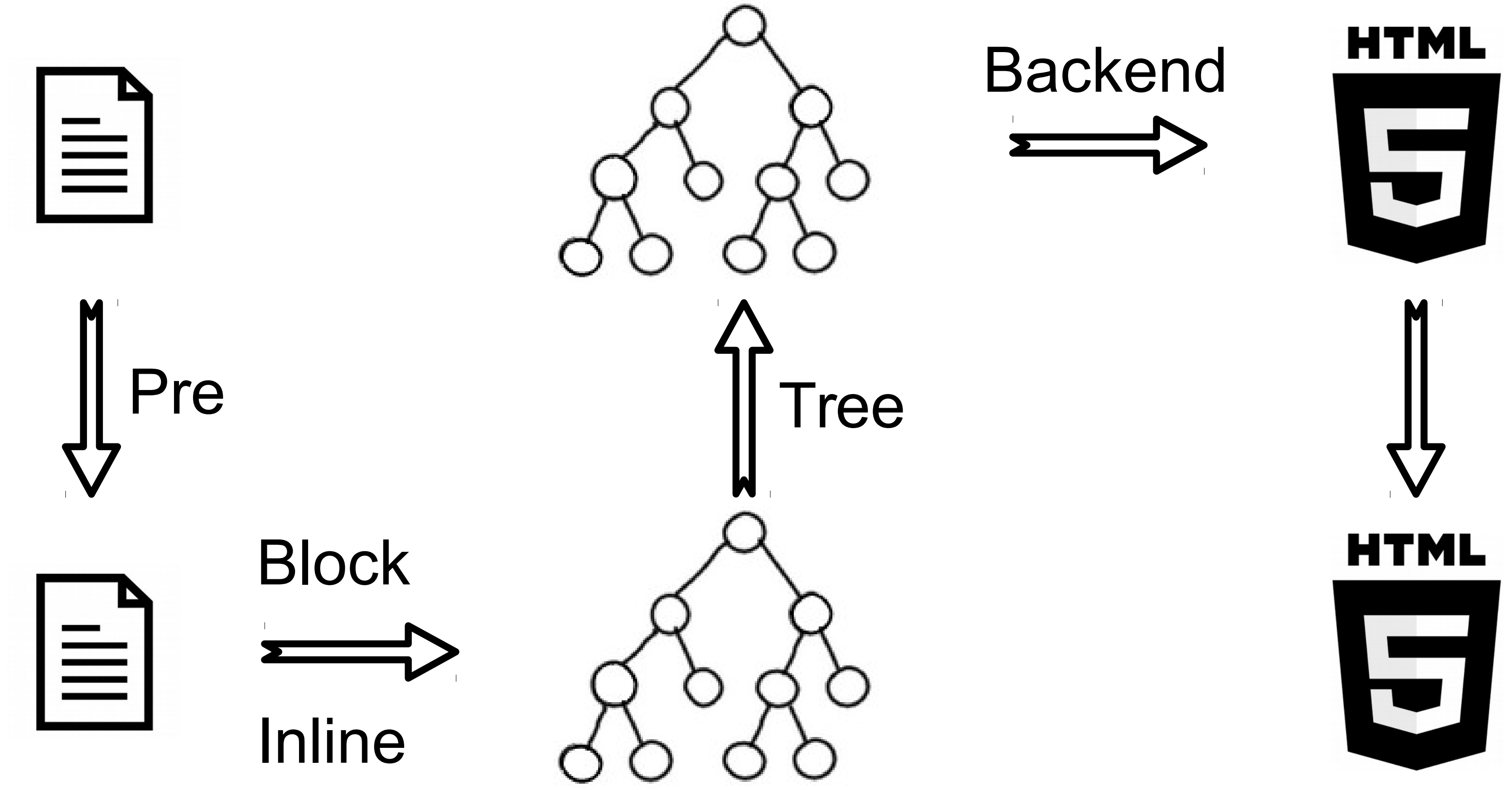
Extending the conversion process



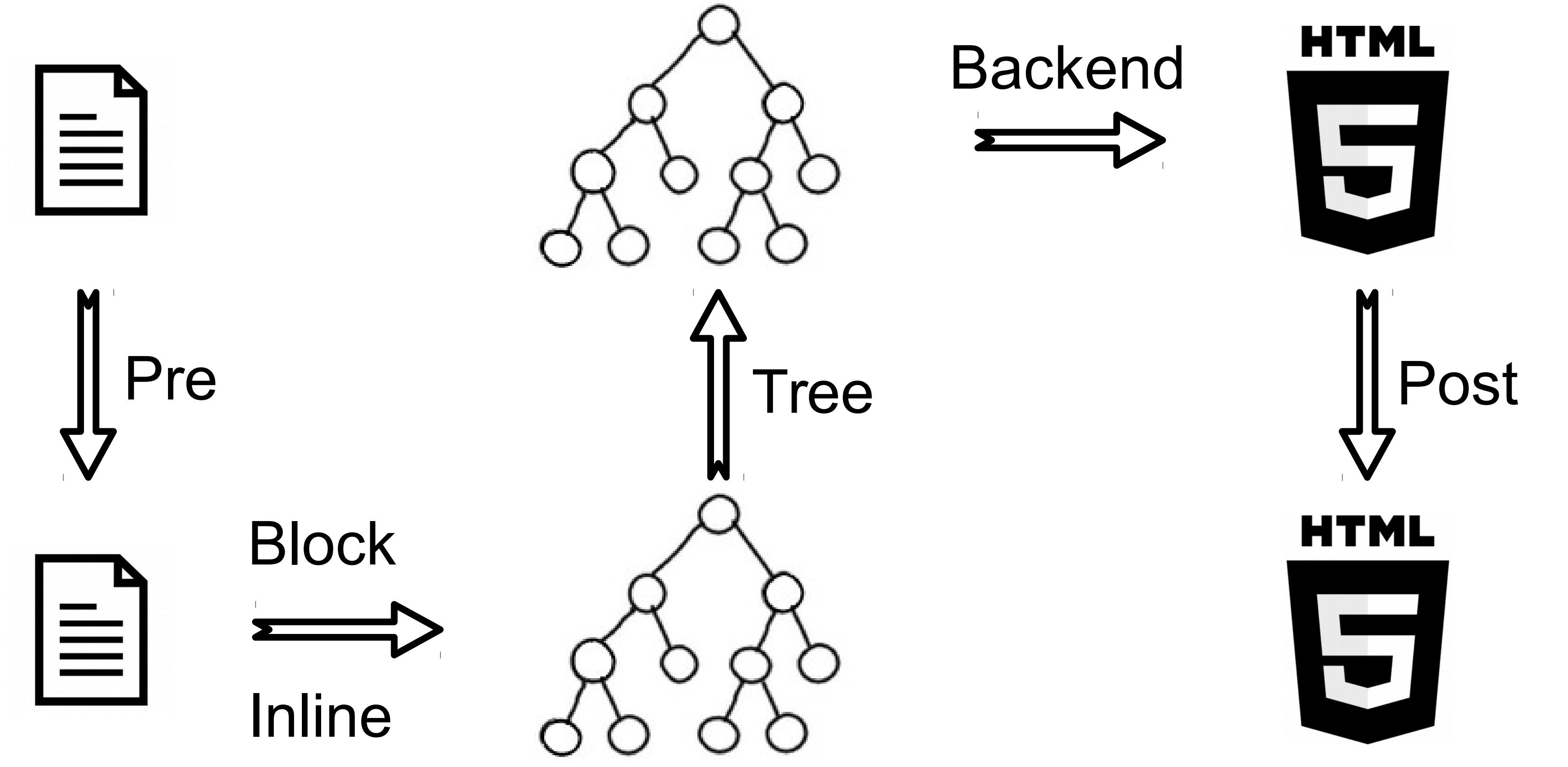
Extending the conversion process



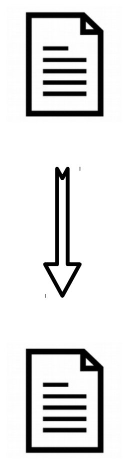
Extending the conversion process



Extending the conversion process



PreProcessor

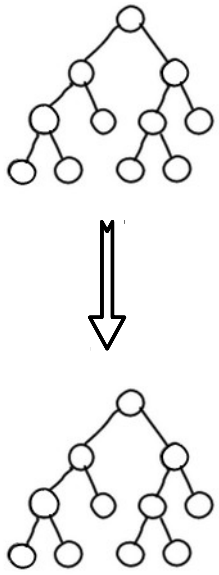


» Process raw AsciiDoc file

» Can change everything

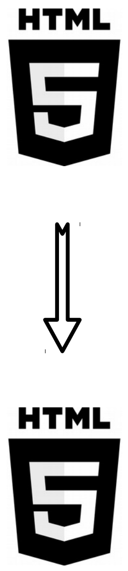
» Result will be passed to parser

TreeProcessor



- » Processes the entire tree
- » Can change every node in the tree
- » Result will be sent to the backend

PostProcessor

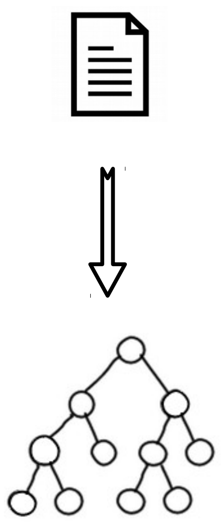


» Processes the output of Asciidoctor

» Can change everything

» Result is the final outcome

BlockProcessor

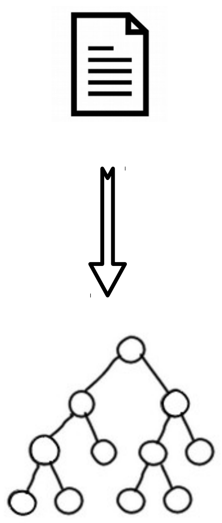


» Processes a single block

» Receives the content of the block

» Creates a node to add to the AST

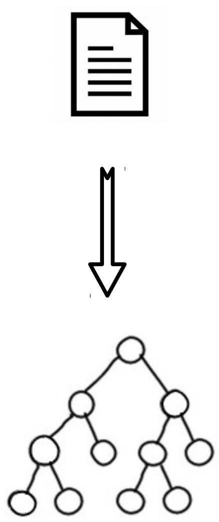
BlockMacroProcessor



» Processes an empty block

» Creates a node to add to the AST

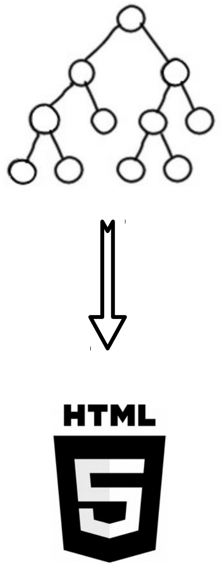
InlineMacroProcessor



» Processes a substring of a block

» Creates a replacement for the substring

Backend



» Convert the tree into the output format

The goal

» BlockMacro which takes screenshot

» Block to control the browser

Simplified Screenshot Extension

```
== Capture Test

[geb]
....
go "file://example.html"
$("input").value("Greach 2016")
....

screenshot::"file://example.html"[]
```

3 Steps

- » Macro which includes a static image
- » Macro taking screenshot
- » Geb block to control browser

Extensions in Gradle

build.gradle

```
asciidoctor {
    extensions {
        inlinemacro(name: "issue") {
            parent, issueNo, attributes ->
                options = [
                        "type": ":link",
                        "target": "${issueTracker}/${issueNo}".toString()
                ]
                createInline(parent, "anchor", target, attributes, options).render()
        }
    }
}
```

AsciidoctorJ 1.6.x

» Annotations for extensions

» Enums for attributes and configurations

» Convenience methods

Go and write your documentation with Asciidoctor

- http://asciidoctor.org/
- http://discuss.asciidoctor.org/
- http://mrhaki.blogspot.ch/