

# tinytemplate

Jesper Öqvist

Department of Computer Science  
Lund University

May 6, 2014

# Goals

- ▶ Simple syntax
- ▶ A minimal feature set
- ▶ Thorough test suite
- ▶ Intelligent indentation
  - ▶ Customizable output indentation
  - ▶ Each line of an expanded string should be properly indented

# Goals

- ▶ Simple syntax
- ▶ A minimal feature set
- ▶ Thorough test suite
- ▶ Intelligent indentation
  - ▶ Customizable output indentation
  - ▶ Each line of an expanded string should be properly indented

# Features

- ▶ Multiple templates per template file
- ▶ Variable expansion
- ▶ Attribute expansion
- ▶ List concatenation
- ▶ Conditional expansion
- ▶ Subtemplate expansion

# Template Declarations

- ▶ A template file includes a number of template declarations
- ▶ Each template is declared with one or more names
- ▶ Template bodies are enclosed in `[ [ and ] ]`
- ▶ The `=` is optional

Examples:

```
template1 = [ [ ] ]
```

```
template2 = [ [xy] ]
```

```
X = Y = [ [$x] ]
```

# Context

- ▶ All template expansions happen with a *context* object
- ▶ The context is used to lookup variables and attributes
- ▶ SimpleContext is the standard context - it stores variables in a map and evaluates attributes by invoking methods on an object

# Variable Expansion

- ▶ Variable references have two forms:
  - ▶ `$varname`
  - ▶ `$(varname)`
- ▶ Variable names can include periods, but remember to use parenthesis: `$(var.name)`
- ▶ Parenthesis are also needed when the reference is in the middle of a word: `$(var)_name`
- ▶ **A Java identifier is parsed if no parenthesis are present.**
- ▶ There are a couple reserved keywords that can not be used for variables - **if, else, endif, cat, include**

# Attribute Expansion

- ▶ A hash sign is used to reference an attribute:
  - ▶ #name
  - ▶ #(name)
- ▶ Attributes are Java methods invoked on the context object (SimpleContext)
- ▶ The attribute method must take no arguments and have non-void return type
- ▶ The expansion of an attribute is the return value of the attribute method converted to a string



# List Concatenation

- ▶ Concatenates a Iterable list of objects using an optional separator string
- ▶ Examples (without & with separator):
  - ▶ `$cat(#list)`  
Expands to "123"
  - ▶ `$cat(#list, ", ")`  
Expands to "1, 2, 3"

# Conditional Expansion

- ▶ Examples:

- ▶ `$if(a) hello $endif`

- Expands to "hello" if the variable a expands to "true"

- ▶ `$if(a) hello $else goodbye $endif`

- Expands to "hello" if a is "true", otherwise "goodbye"

- ▶ `$if(#a) hello $endif`

- If the **attribute** a evaluates to "true" the statement expands to "hello"

# Subtemplate Expansion

- ▶ Templates can be expanded inside each other (recursion will result in a stack overflow!):
  - ▶ `$include(a)`  
Expands the template named “a” in the current template. Indentation works as expected – if the statement is indented the expansion will be indented the same additional amount.

# JastAdd Integration

```
// Test.jrag:
syn String A.name() = "myName";

// Test.java:
tt = templateFromFile("Test.tt");
context = new SimpleContext(tt, new A());
context.expand("my.template", out);

// Test.tt:
my.template = [[public String #name;]]

// output:
public String myName;
```