

## JPlates 3.0 Template Method Syntax Summary



The Template for Success™

Object-Oriented Template Processing for Java

JPlates template methods are allowed wherever normal Java methods are allowed.

```
<template-method> ::=
    [ <modifiers> ] TEMPLATE <identifier> ( <parameter-list> )
    [ throws <exception-list> ]
    #[ <template-body-member>* #]

<template-body-member> ::=
    <template-statement>
|    #( <expression> )# // basic substitution
|    #( @ <expression> [ , { <argument-list> } ] )# // internationalization
|    #{ <template-body-member>* #} // template-block
|    <template-literal-text>

// Note: template-statements always start on a new line
<template-statement> ::=
    # <identifier> : // labeled statement
|    # <local-variable-declaration> ;
|    # <expression-statement> ; // method-call, assignment, etc.
|    # assert <expression> [ : <expression> ] ;
|    # break [ <identifier> ] ;
|    # continue [ <identifier> ] ;
|    # return; // template methods always void
|    # throw <expression> ;
|    # synchronized ( <expression> ) <template-block>
|    # for ( <same-as-Java> ) <template-block>
|    # while ( <expression> ) <template-block>
|    # do <template-block>
|        # while ( <expression> );
|    # try <template-block>
|        [ # catch ( <exception-decl> ) <template-block> ]*
|        [ # finally <template-block> ]
|    # if ( <expression> ) <template-block>
|        [ # else if ( <expression> ) <template-block> ]*
|        [ # else <template-block> ]
|    # switch ( <expression> ) #{
|        [ # ( case <constant-expression> : | default : )
|            <template-statement>* ]*
|        #}
|    # @indentLocal ( = | += | -= ) <int-expression> ;
|    # @indentGlobal ( = | += | -= ) <int-expression> ;
|    # @indentEnabled = <boolean-expression> ;
```

- JPlates comments start with ## and go to the end of the line. They do not contribute to the template output, and are translated into Java comments.
- Use #: to discard initial leading whitespace before literal text or substitutions.
- Inside literal text, “\#” is taken to be a literal “#” character in the template output.
- Leading whitespace before template statements, template blocks, and template comments does not contribute to template output.
- Leading whitespace before literal text and before substitutions is preserved, and initial whitespace is adjusted by indentation control.