

# Eclipse 代码模板

## 使用代码模板

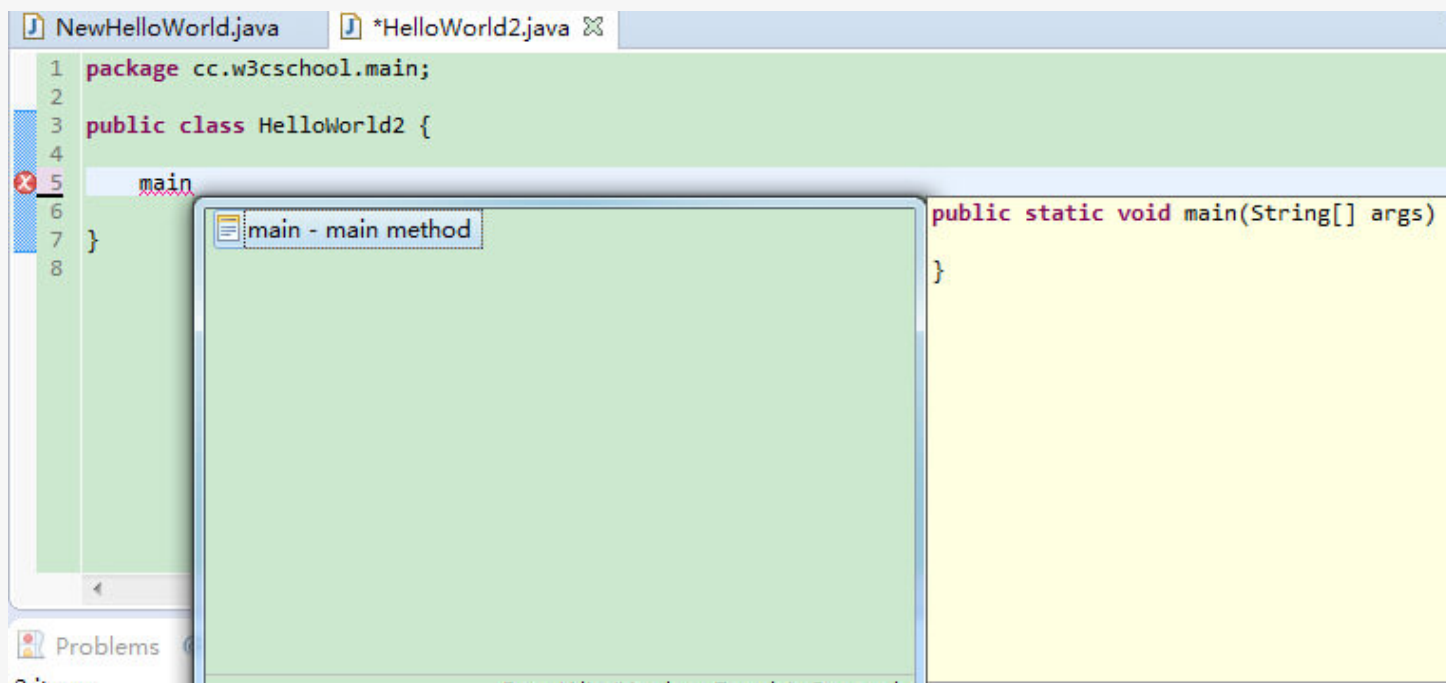
Eclipse 提供了通过定义和使用代码模板来提高工作效率与代码可预测性的能力。

我们在开发 Java 程序过程中经常需要编写 main 方法：

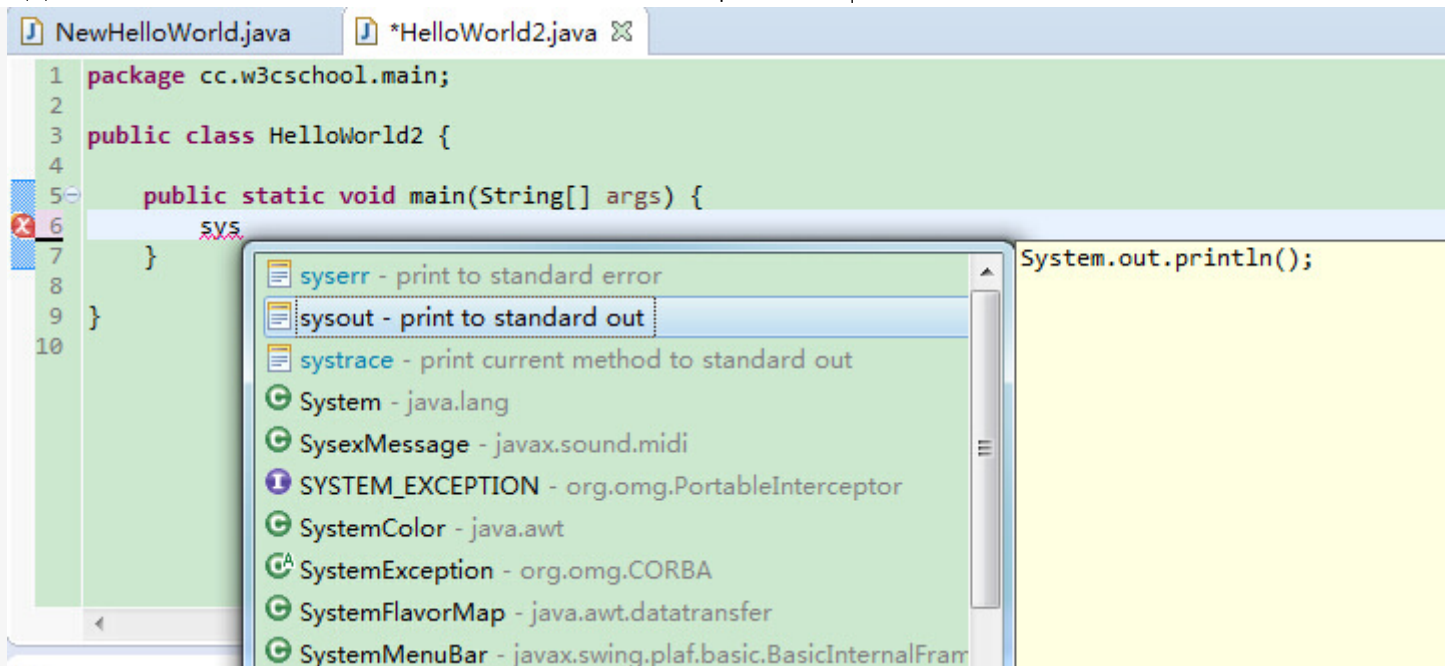
```
public static void main(String[] args) {  
  
}
```

如果我们一个字母一个字母去编写，将是一个重复而又毫无意义的事情，这是我们就可以使用 Eclipse 代码模板来快速完成这些工作。

我们只需在类体中键入main，然后使用Eclipse的代码提示快捷键(默认为Alt+/, 回车后，就可以看到Eclipse自动帮我们完成了main函数的完整定义：

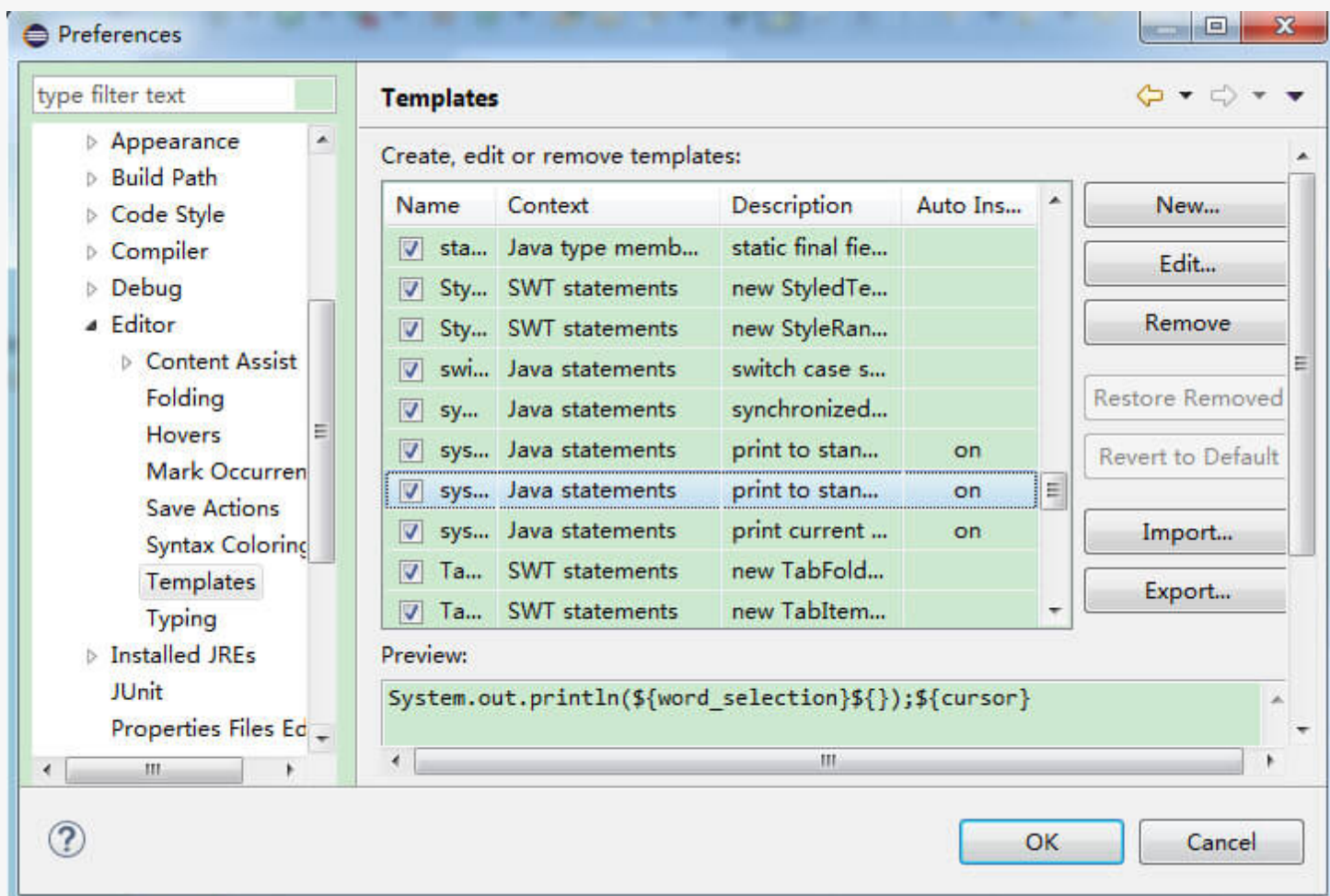


如果我们要使用 System.out.println()，我们只需要输入 syso 然后按下 Alt+/ 即可：

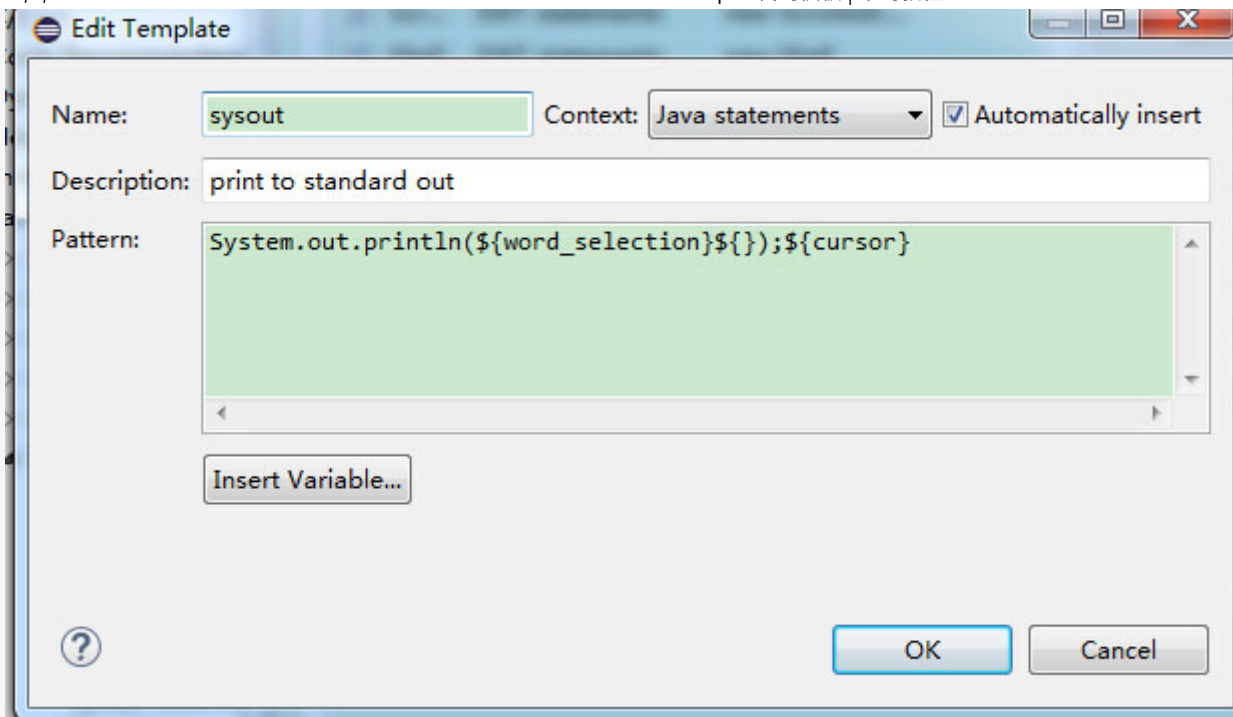


## 自定义代码模板

Eclipse 还提供了非常多的代码模板，我们可以通过 Windows->Preferences->Java->Editor->Templates (你可以在搜索框中输入Templates查找)看到所有已定义的代码模板列表。



我们在弹窗口选中 sysout 模板并点击右侧Edit，显示如下：



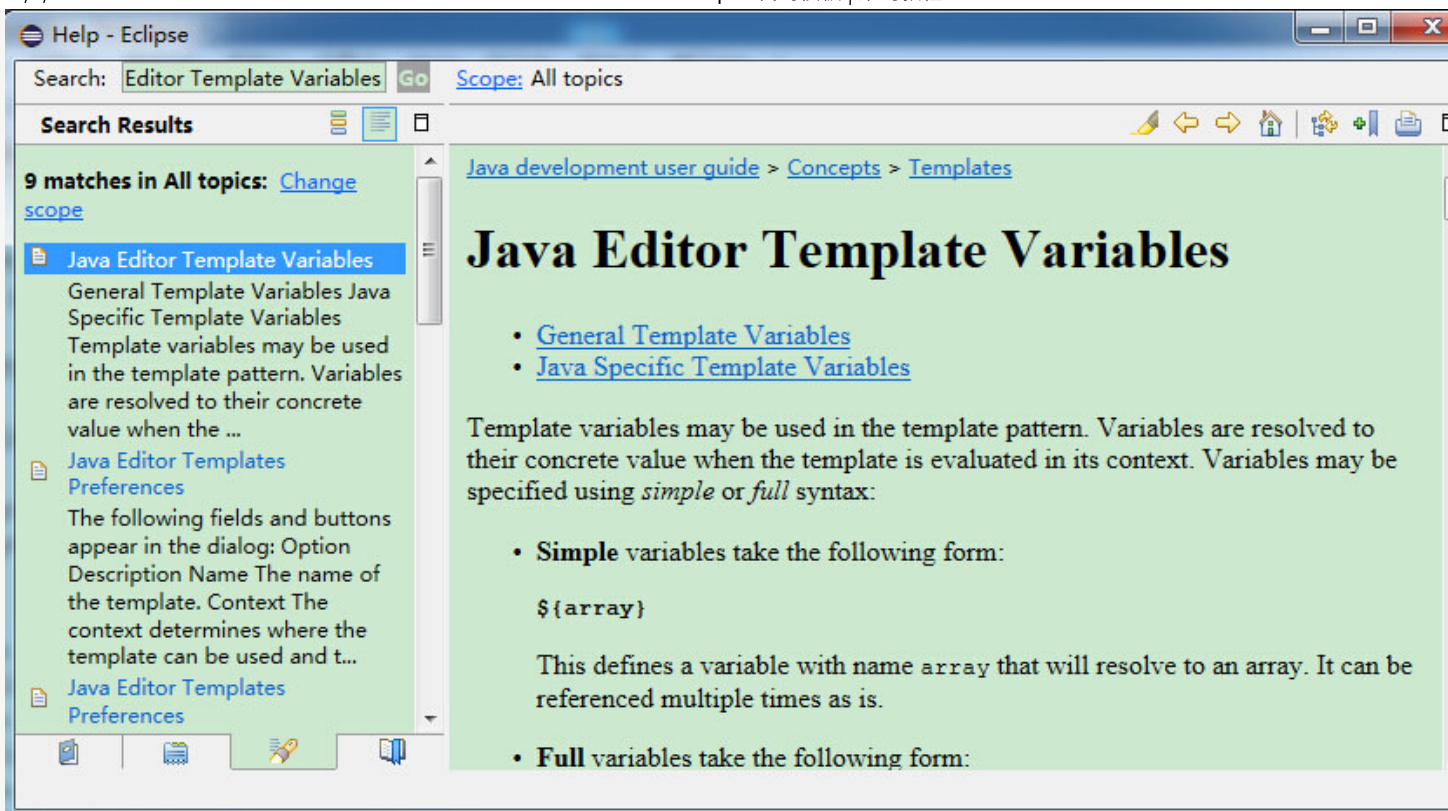
编辑面板是核心关注对象，因为一切东西都在这里配置。先来熟悉下这个面板中关键的五项分别是什么。

- **Name**：名称，其实就是以后可以用到的代码缩写
- **Context**：模板上下文，指定该代码模板在什么地方才能生效，对于Java至少包含这么四个：
  - Java type members，模板对应的代码是类成员，psvm模板严格来说应该选择这个
  - Java statements，模板对应的代码是语句块
  - Java，最通用的，只要是Java代码就行
  - Java doc，顾名思义了
- **模板变量**：eclipse已经预置了一些模板变量（点Insert Variables可以看到所有预置变量），如：
  - `${cursor}`是表示光标
  - `${date}`表示当前日期字符串
  - `${time}`表示当前时间字符串
  - `${line_selection}`让当前行被选中
  - `${word_selection}`让当前单词被选中

当然我们也可以定义自己的模板变量，比如我定义一个 `${myTemplateVariable}`，那么对应代码显示的就是 myTemplateVariable。

- **Pattern**：代码模板对应的模式，按照你希望代码的格式逐个输入即可

更多自定义代码模板的内容你可以通过点击 Help 菜单中的 Help Contents 选项，在弹出的对话框的搜索栏上输入 "Java Editor Template Variables" 选择 Java Editor Template Variables 查看具体的文档描述：



The screenshot shows the Eclipse Help window with the search bar containing 'Editor Template Variables'. The search results list 9 matches in All topics. The first result, 'Java Editor Template Variables', is selected and expanded, showing a list of sub-topics: 'General Template Variables', 'Java Specific Template Variables', and 'Template variables may be used in the template pattern. Variables are resolved to their concrete value when the ...'. The main content area displays the 'Java Editor Template Variables' page, which includes a breadcrumb trail 'Java development user guide > Concepts > Templates', a title 'Java Editor Template Variables', and a list of sub-topics: 'General Template Variables' and 'Java Specific Template Variables'. The page also contains a paragraph explaining that template variables may be used in the template pattern and are resolved to their concrete value when the template is evaluated in its context. Variables may be specified using *simple* or *full* syntax. The page lists two types of variables: 'Simple variables take the following form:' and 'Full variables take the following form:'. The 'Simple variables' section shows the syntax `${array}` and explains that it defines a variable with name `array` that will resolve to an array. It can be referenced multiple times as is.

Search:  Go Scope: All topics

Search Results

9 matches in All topics: [Change scope](#)

- Java Editor Template Variables
  - General Template Variables
  - Java Specific Template Variables
  - Template variables may be used in the template pattern. Variables are resolved to their concrete value when the ...
- Java Editor Templates Preferences
  - The following fields and buttons appear in the dialog: Option Description Name The name of the template. Context The context determines where the template can be used and t...
- Java Editor Templates Preferences

Java development user guide > Concepts > Templates

## Java Editor Template Variables

- [General Template Variables](#)
- [Java Specific Template Variables](#)

Template variables may be used in the template pattern. Variables are resolved to their concrete value when the template is evaluated in its context. Variables may be specified using *simple* or *full* syntax:

- Simple** variables take the following form:  
`${array}`  
This defines a variable with name `array` that will resolve to an array. It can be referenced multiple times as is.
- Full** variables take the following form:

[← Eclipse 安装插件](#)[Eclipse 快捷键 →](#)[点我分享笔记](#)