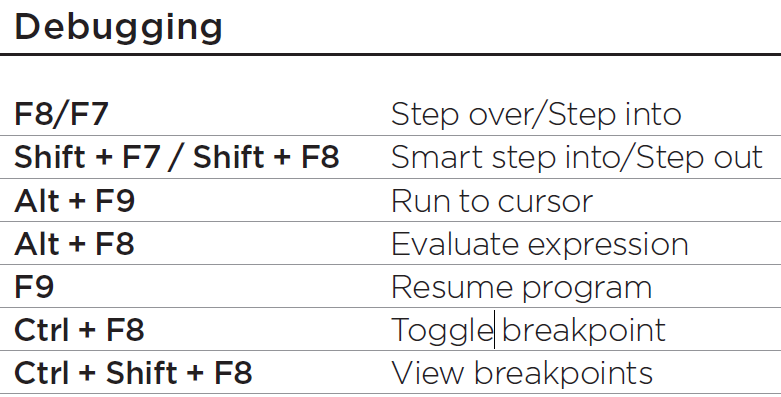
调试快捷键-Debugging



应当熟练使用的是：

**F7、F8、Shift+F7、Shift+F8、Alt+Shift+F7、Ctrl+Shift+F8**

**F9、Alt+F9**

**Ctrl+Shift+F8**

**Ctrl + F8、Alt+F8**

详细介绍：

F7、F8都是下一步，F7是进入方法内Step Into，F8不进入，Step Over，直接跳过。

## F7：Step Into，下一步，进入到代码<相当于Eclipse的F5>;

如果当前行断点是一个方法，则进入当前方法体内。

## F8：Step Over，下一步，跳到下一步<相当于Eclipse的F6>;

如果当前行断点是一个方法，则**不进入当前方法体内**。

## Shift+F7：Smart Step Into；

F7与Shift+F7的区别：

F7只能进入到该项目中的方法中；而Shift+F7：可以进入JDK提供的源码中。

如sb.append(headTemp.val).append("-->");

F7不会进入，而Shift+F7可以智能进入。

System.out.println(commonNode.val);

同样，F7无法进入，而Shift+F7可以进入。

## Shift+F8：Smart Step Out；

当通过F7或Shift+F7进入了方法内部之后，可以通过Shift+F8直接退出原来的方法。

## <Alt+Shift+F7>： 不常用，放在最后。

## Ctrl+Shift+F8：设定断点进入条件；

如设置一个for循环，i从0到10；只需要在i=9的时候，单步执行循环体，此时设置改断点的**Condition：i==9**。

## F9：Resume program，恢复程序连续运行，直至停在下一个断点上，也就是跳到下一断点BreakPoint；

## Atl+F9：Run To Cursor 运行到光标处；

## Ctrl+F8：添加或取消断点

设置光标所在行为断点，如果当前已经是断点则去掉断点；

## Alt+F8：Evaluate，输入表达式查看结果Expression—>Result；

选中对象，弹出**可输入计算表达式Expression调试框**，查看该输入内容的调试结果Result。

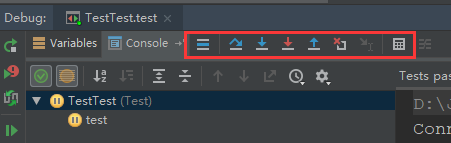
## Alt+Shift+F7：Force Step Into，强制进入代码，如果当前行断点是一个方法，则进入当前方法体内，如果方法体还有方法，则会进入该内嵌的方法中，依此循环进入。

Toggle breakpoint：**开启或关闭断点Ctrl + F8**

**toggle**英 [ˈtɒgl] 美 [ˈtɑ:gl] n. 棒形纽扣;套索扣;转换键;切换键 v. 切换

resume英 [rɪ'zju:m] 美 [rɪ'zju:m]

v. 继续;重新开始;恢复职位 n. 简历;摘要



 ：Show Execution Point (Alt+F10);

：**Step Over(F8)**;

：Step Into(F7);

：Force Step Into Alt+Shift+F7；

：Step Out(Shift+F8)；<使用频率高>

：Drop Frame；

：Evaluate Expression(Alt+F8)。



<https://blog.csdn.net/gezilan/article/details/80177769>