Intellij Idea使用技巧、快捷键

提示：如果快捷键不起作用，可能是跟输入法有冲突，取消输入法的快捷键即可。

提高输入效率的要点：能用键盘操作的，绝不用鼠标~

## 常用快捷键：

Ctrl+Shift+A (action)：可以查找所有Intellij的命令，并且每个命令后面还有其快捷键。它是查找学习快捷键的工具。

shift+F10 ：运行(初次运行得用shift+ctrl+F10)

shift+F9：调试

ctrl+shift+F9：重新编译(recomplie)。如果没有新增文件，不用每次都重新调试或者重启，只需要recomplie就可以调试/运行最新的代码了。

alt+enter：自动修复，包括导入类、导入包等

双shift：查找所有类和方法(特别好用！)

ctrl + E ：查找Recent Files，也就是最近的文件

ctrl + N ：查找类

ctrl+shift+F ： 全局搜索内容

ctrl + alt + <- ： 回到上一步

ctrl + alt + -> ： 回到下一步

ctrl + W：选你所想，连续按会有其他效果。非常好用~

ctrl+shift+enter：补齐行尾的符号，比如括号或者分号，经常用来迅速补充当前语句行尾的分号，或者if()后面的括号。

ctrl+G ：定位某一行

ctrl+z ：撤消

ctrl+z+shift ： 恢复

/** ： 再按回车，生成注释

ctrl+F12：显示类的大纲，包括方法、变量等结构。

变量名.null ： 快速生成 if (变量==null){}

变量名.notnull ： 快速生成 if (变量!=null){}

变量名.for ： 快速生成for循环

变量名.var ： 快速声明变量。

比如输入一个字符串,"userName".var，就可以直接生成 String userName = "userName";

也适合于其他的类。比如项目中有一个类Developer，输入 new Developer().var，直接生成 Developer developer = new Developer();

## 智能提示

Ctrl+Shift+Space：智能提示
F2：快速移动到warnning或error的代码

Alt+Enter：快速修复（自动填充对象类型、转换类型）
Ctrl+Shift+Enter：自动补全末尾的字符（括号、花括号）

ctrl+space ： 根据提示自动补全(与输入法的切换快捷键有冲突，建议改成类似eclipse的快捷键 alt+/)

自动提示忽略大小写：File-->Setting-->搜索框输入跳转到Code Completion-->选中All letters-->去掉Match case（匹配大小写）

## 类和文件

ctrl+Tab ： 可以在IDEA界面上的各个模块之间切换。

ctrl+F12：显示类的大纲，包括方法、变量等结构。

ctrl+Alt+O ： 格式化import列表 ，去除类里面无用的import

ctrl+shift+Alt+U： 查看继承关系

点击代码行数旁边的箭头，可以查看相关类。箭头向下表示查看子类。

ctrl+N：快速查找类

ctrl+H ：查看类的层次结构，可以查到父类、子类等

alt+Insert ：生成getter和setter，还能生成constructor

alt + 左箭头 ： 跳转到左边的文件

alt+右箭头：跳转到右边的文件

ctrl+shift+T：创建单元测试

ctrl+G：跳到某一行

ctrl+Home：跳到当前类的第一行

ctrl+End：调到当前类的最后一行

F4：点击项目包，按F4可以打开project structure，可以设置Jdk、依赖和target输出

选中项目文件，按住右箭头，可以快速展开目录。

Alt+数字：切换窗口，常用的有1-项目结构，3-搜索结果，4/5-运行调试。

Ctrl+Tab：切换标签页

方法

ctrl+alt+H ： 查看在何处调用了该方法

ctrl+I ： 生成接口的方法

ctrl+O ： 覆盖基类的方法。可以快速重写方法。

alt + 上箭头/下箭头：跳转到当前类的上/下一个方法间

ctrl+P ：查看方法参数提示
ctrl+Q：可以看到当前方法的声明
Ctrl+”+/-”，当前方法展开、折叠
ctrl+鼠标点击方法名：查看何处调用了该方法。
ctrl+B：进入光标所在方法(变量)定义的地方或返回该方法(变量)被使用的地方（代替Ctrl+鼠标点击方法进入方式，避免了手指在键盘和鼠标之间切换，非常好用的快捷键）

## 编辑

ctrl+Y ：删除当前行

ctrl+alt+L：使代码对齐

ctrl+shift+V：查找并使用最近复制过的内容，进行粘贴

tab+shift ：反向缩进

ctrl + W：选你所想，连续按会有其他效果。非常好用~

ctrl + shift +W：取消所选。

ctrl + [ 或者ctrl + ] ： 移动到前/后代码块的括号处

ctrl+shift+[ ：选中括号内的代码块。

ctrl+shift+u： 大小写的快速切换。(也可以在"Edit"导航栏下点击"Toggle Case"进行切换)
Ctrl+Backspace：按单词删除

Ctrl+←(→) ：移动到一个词的开始(结尾)

Ctrl+Shift+←(→) ：从后到前（从前到后）选中一个词

Shift+Enter：插入一行

home：跳转到行首

end：跳转到行尾

shift+home：快速选中当前位置到行首的内容（其他编辑器也适用）

shift+end：快速选中当前位置到行尾的内容（其他编辑器也适用）

ctrl+home：快速到顶部

ctrl+end：快速到底部

## 查找和替换

ctrl + E ：查找Recent Files(非常好用)

ctrl+shift+E： 最近编辑的文件以及具体内容！(居然还有这种功能)

ctrl+shift+F ： 全局搜索内容

ctrl+F：在本文件中查找，之后可以通过F3向下查找，通过shift+F3向上查找。

ctrl+F3：先选中某个变量或者具体值，然后按ctrl+F3,就可以对该内容进行搜索。可以通过F3向下查找，通过shift+F3向上查找。

ctrl+N ：查找类

ctrl+shift+N ：查找文件

双击shift ： 在所有文件中查找对应类和方法

ctrl+R ： 替换
ctrl+shift+R ：所有文件选择性替换

在项目文件夹/弹窗/侧栏查找：点击选中项目文件夹(或者是其他的弹窗/侧栏)，直接输入字母，就可以搜索相关的内容了，非常好用~

Todo：找到界面最下方的工具栏，选择Todo，如果想在当前文件查找，可以选择Current File

## 重构

shift+F6 ： 重构

ctrl+alt+M ： 提炼方法

ctrl+alt+N 提炼方法的逆操作
ctrl+alt+C 快速提取出常量（Constant）
ctrl+alt+V快速提取出变量（Variable）
ctrl+alt+F 快速提取出成员变量（Filed Variable）
ctrl+shift+F6 重构变量的类型

ctrl+shift+alt+T 重构一切

## 生成代码(代码智能补全)

psvm ： 再按tab，打出main函数
sout ： 再按tab，打出System.out.println();

iter ： 生成foreach循环

psfs ：  public static final String
fori ： 再按tab，打出for循环

ifn ： 快速生成 if( ==null)

变量名.null ： 快速生成 if (变量==null){}

变量名.notnull ： 快速生成 if (变量!=null){}

变量名.for ：快速生成for循环

alt+Enter：先写赋值后面的变量或者方法，再按alt+Enter，可以快速声明变量。

比如 输入方法名： test(); 然后按alt+Enter，选择"Introduce local variable"，可以根据类型生成 String name=test();

自定义代码生成：File ---> Settings ---> Editor ---> Live Templates，点击 + 号，自定义代码生成快捷键。

更多的智能代码生成见：https://www.cnblogs.com/javastack/p/11230120.html

## 其他

代码版本管理VCS

1.ctrl+T ： 更新代码

2.ctrl+k ：提交代码

3.从版本仓库导出项目代码 VCS-->CheckOut From Version Control

4.点击工具栏的"Compare With"，可以对比修改的内容，按F7迅速跳转下一处修改的内容。

5.点击工具栏的"Show History"，可以查看历史

6.点击工具栏的"Revert"，可以进行回退，撤销掉修改的内容。

代码检查分析

1.右键选择analyze，点击Inspect Code。

2.选中方法或变量，右键选择analyze，选择analyze data from here，analyze data to here。

## 单元测试/覆盖率测试

1.ctrl+shift+T：创建单元测试

2.选中类或者文件，点击RunWithCoverage，进行覆盖率的测试。

## 调试技巧：

1.左键点击行首，设置断点

2.shift+F9  开始调试

ctrl+shift+F9：重新编译。如果没有新增文件，不用每次都重新调试或者重启，只需要recomplie就可以调试/运行最新的代码了。

3.F7:step into 单步执行，遇到方法就进入方法

F8:step over 单步执行，遇到方法不会进入，直接跳过

F9:resume program 跳到下一个断点

alt+shift+F7:force step into 强制进入下一步

shift+F8:step out 单步执行，如果在方法内就跳出方法，如果在方法外就跳到下一个断点

alt+F9:run to cursor 跳转到光标处

alt+F8:evaluate expression 计算表达式

drop frame:回退线程栈。

mute breakpoints: 暂时屏蔽断点。

show execution points:显示正在运行的断点。

## 自定义快捷键：

自定义快捷键的方法： File--Setting--KeyMap

1.搜索Line End，将End快捷键修改为`，然后搜索Line Start，将Home快捷键修改为Shift+`。

这是因为Home和End的键位在另一侧，用起来很麻烦。

## 浏览器快捷键：

1.alt + tab ：可以从Idea切换到浏览器。

2.ctrl+ tab ：切换到浏览器的下一个标签页。

涉及到标签页的切换，大多可以用ctrl+ tab。比如NotePad++里面也可以用。

3.ctrl + shift +tab ：切换到浏览器的上一个标签页。

4.ctrl + 数字：可以直接切换到第几个标签页。

5.tab ：跳转到浏览器当前页面的下一处可编辑处。

要从Idea切换到浏览器进行浏览或者搜索，可以使用上面的1到5快捷键进行操作。

## 常用设置：

1.Idea设置字体大小:

file -> setting -> editor -> colors&font -> save as并自定义 -> font -> 修改size大小

2.Idea改变背景颜色：

file -> setting -> editor -> colors&font -> general -> text -> default text ->勾选Background

3.project相当于Eclipse中的workspace，module相当于Eclipse里真正的project。

4.Idea复制错误提示：选中出错的代码，在窗口最下方可以看到错误提示，右键 "copy"即可

5.Idea显示侧栏、工具栏：View-->勾选Toolbar、Tool Buttons

6.设置当前类的编码:点击右下角的GBK或者UTF-8，自由切换

设置当前项目的编码:File--Setting--Editor--File Encodings--Project Encoding。

7.添加jar包: File--Project Structure--Libraries，点击"+"即可.

8.autoscroll from source 点击源码，自动跳转到对应的文件

autoscroll to source 点击文件，自动跳转到对应的源码

9.artifact部署：

普通的ssm项目，需要设置artifact，还要手动设置configuration中的tomcat及context(上下文)。

如果是springBoot项目，则不需要设置。

https://blog.csdn.net/u014042066/article/details/75299002

10.Editor--->General---->Appearance---->Caret blinking取消勾选，表示鼠标光标不跳动。

11.设置类的注释模板：

https://www.cnblogs.com/expiator/p/10057543.html

## 必备插件

JRebel for IntelliJ：热部署，修改代码后不用重新启动项目就能看到效果。

Alibaba Java Coding Guidelines ：阿里巴巴代码规范，自动检查代码是否规范。

.ignore：git提交时过滤掉不需要提交的文件

Lombok：不再写get/set方法，还能快速的实现builder模式

Codehelper.generator：生成代码

Free mybatis plugin：方便在dao层和mapper之间跳转

Rainbow Brackets：不同颜色的代码块

JUnitGenerator V2.0：单元测试必备。

FindBugs：找bug。

Background image Plus：背景图片

MyBatis Log Plugin：把mybatis的sql日志替换成原生的sql

Maven Helper：找出jar包冲突。

RestfulToolkit：快速查找Url对应的方法。