

IDEA 快捷键汇总

作者: [bigdaddy](#)

原文链接: <https://ld246.com/article/1505014474441>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

智能提示

有过实际编程经验的读者都应该知道，任何一款 IDEA，是否好用和受欢迎的第一要素就是其智能提示的准确性和快速性。所以，我们先看看 IntelliJ IDEA 中的代码智能提示功能。

`Space` 就是指空格键

代码重构

有着实际编程经验的读者肯定或多或少地重构过别人或自己的旧代码，特别是在阅读了《重构改善既有代码的设计》之后，有着代码洁癖的读者肯定有经常重构旧代码的需求。实际上，在 Eclipse 中已经提供了很多强大和智能的重构功能。但是，IntelliJ IDEA 后来居上，提供了更多、更智能的重构功能。智能提示结合重构功能使得 IntelliJ IDEA 下的 TDD 开发变得十分简单和流畅。

<tr>
<td>重命名</td>
<td><code>Shift</code> + <code>F6</code> </td>
</tr>

<tr>
<td>提取变量</td>
<td><code>Ctrl</code> + <code>Alt</code> + <code>V</code> </td>
</tr>

</tbody>
</table>

<h4 id="代码生成">代码生成</h4>

<p>在日常的编程实践中，笔者经常需要编写一些重复且常有的代码片段。实际上，在 Eclipse 和 IntelliJ IDEA 中都提供了非常便捷的代码模板来减少开发工程师的重复操作。此类快捷键的使用频率很高</p>

<blockquote>

<p><code>Tab</code> 是代码模板的触发按钮，生成 boilerplate 样板代码片段。</p>

</blockquote>

<table>
<thead>

<tr>
<th>功能</th>
<th>快捷键</th>
</tr>

</thead>

<tbody>

<tr>
<td>for 循环</td>
<td><code>fori</code> + <code>Tab</code> </td>
</tr>

<tr>
<td>System.out.println 语句</td>
<td><code>sout</code> + <code>Tab</code> </td>
</tr>

<tr>
<td>main 方法</td>
<td><code>psvm</code> + <code>Tab</code> </td>
</tr>

<tr>
<td>查询所有代码模板</td>
<td><code>Ctrl</code> + <code>J</code> </td>
</tr>

<tr>
<td>自动生成构造函数、toString 函数、getter/setter、重写父类方法等</td>
<td><code>Alt</code> + <code>Insert</code> </td>
</tr>

</tbody>
</table>

<p>后缀自动补全 (Postfix Completion) : 示例如下</p>

<blockquote>

user <code>.for</code> + <code>Tab</code> => for(User user : users)

user.getBirthday()<code>.var</code> + <code>Tab</code> => Date birthday = user.
etBirthday()

</blockquote>

<h4 id="高效编辑">高效编辑</h4>

<p>在代码的编辑过程过程中，IntelliJ IDEA 参考 vim、Emacs 等，提供了一套非常智能、高效的编辑快捷键。</p>

<table>

<thead>

<tr>

<th>功能</th>

<th>快捷键</th>

</tr>

</thead>

<tbody>

<tr>

<td>根据语言的语法特性来扩展或收缩光标所选范围</td>

<td><code>Ctrl</code>+<code>W</code> 和 <code>Ctrl</code>+<code>Shift</code>+<code>W</code></td>

</tr>

<tr>

<td>以 字符 为单位进行前后移动</td>

<td><code>Left</code> 和 <code>Right</code></td>

</tr>

<tr>

<td>以 单词 为单位进行前后移动</td>

<td><code>Ctrl</code>+<code>Left</code> 和 <code>Ctrl</code>+<code>Right</code></td>

</tr>

<tr>

<td>以 代码块 为单位进行前后移动</td>

<td><code>Ctrl</code>+<code>/</code> 和 <code>Ctrl</code>+<code>Shift</code>+<code>/</code></td>

</tr>

<tr>

<td>删除当前行</td>

<td><code>Ctrl</code>+<code>Y</code></td>

</tr>

<tr>

<td>复制当前行并插入在下面一行</td>

<td><code>Ctrl</code>+<code>D</code></td>

</tr>

</tbody>

</table>

<h4 id="查找打开">查找打开</h4>

<p>对于 Java 程序员而言，Eclipse 中的查找和打开资源功能是一个经常使用和便利的功能。当然，IntelliJ IDEA 也提供了同样的便利性，而且还进行了智能加强。</p>

<table>

<thead>

<tr>

<th>功能</th>

<th>快捷键</th>

</tr>

</thead>

<tbody>

<tr>

<td>打开类、文件等资源</td>

<code>Ctrl</code> + <code>N</code> 和 <code>Ctrl</code> + <code>Shift</code> + <code>N</code>
全局搜索 (Serarch Every Where)
<code>Shift</code> + <code>Shift</code>
打开当前类的继承层次窗口
<code>Ctrl</code> + <code>H</code>
在类的继承层次窗口进行跳转
<code>Ctrl</code> + <code>B</code> 和 <code>Ctrl</code> + <code>Alt</code> + <code>B</code>
查看当前类的所有方法
<code>Ctrl</code> + <code>F12</code>
查找类或者方法的使用
<code>Alt</code> + <code>F7</code>
当前窗口中进行文本查找
<code>Ctrl</code> + <code>F</code>
全工程中进行文本查找
<code>Ctrl</code> + <code>Shift</code> + <code>F</code>
在查找的所有匹配处间进行移动
<code>F3</code> 和 <code>Shift</code> + <code>F3</code>

基础功能

实际上，除了上述的这些专题快捷键，IntelliJ IDEA 还提供了非常多的基础功能快捷键。如果能够熟练地掌握这些快捷键的用法，那么基本可以做到双手在 95% 的时间内不需要离开键盘，开发率势必大大地提高。

功能	快捷键
查询所有的 IntelliJ IDEA 命令和对应快捷键	<code>Ctrl</code> + <code>Shift</code> + <code>A</code>

<tr>	<td>自动新建类、文件、文件夹等资源</td>
<td><code>Alt</code> + <code>Insert</code></td>	</tr>
<tr>	<td>优化 import 列表</td>
<td><code>Ctrl</code> + <code>Alt</code> + <code>O</code></td>	</tr>
<tr>	<td>格式化代码</td>
<td><code>Ctrl</code> + <code>Alt</code> + <code>L</code></td>	</tr>
<tr>	<td>聚焦到编辑窗口中</td>
<td><code>ESC</code></td>	</tr>
<tr>	<td>聚焦到工具窗口中</td>
<td><code>Alt</code> + <code>NUM</code></td>	</tr>
<tr>	<td>在编辑窗口的标签页间进行切换</td>
<td><code>Ctrl</code> + <code>Tab</code></td>	</tr>
<tr>	<td>打开最近访问过或者编辑过的文件</td>
<td><code>Ctrl</code> + <code>E</code> 和 <code>Ctrl</code> + <code>Shift</code> + <code>E</code></td>	</tr>
<tr>	<td>创建单元测试用例</td>
<td><code>Ctrl</code> + <code>Alt</code> + <code>T</code></td>	</tr>
<tr>	<td>开始运行程序</td>
<td><code>Alt</code> + <code>Shift</code> + <code>F10</code></td>	</tr>
<tr>	<td>开始调试程序</td>
<td><code>Shift</code> + <code>F9</code></td>	</tr>
<tr>	<td>停止程序</td>
<td><code>Ctrl</code> + <code>F2</code></td>	</tr>
<tr>	<td>调试程序时</td>
<td><code>F7</code>/<code>F8</code>/<code>F9</code> 对应 Step into、Step over、ontinue</td>	</tr>

</tbody>

</table>

<p>最终，我们根据所有快捷键的使用频率和功能性，总结出一个 IntelliJ IDEA 的十强榜单。</p>

```

<table>
<thead>
<tr>
<th>功能</th>
<th>快捷键</th>
</tr>
</thead>
<tbody>
<tr>
<td>智能补全</td>
<td><code>Ctrl</code> + <code>Shift</code> + <code>Space</code> </td>
</tr>
<tr>
<td>智能修复</td>
<td><code>Alt</code> + <code>Enter</code> </td>
</tr>
<tr>
<td>重构一切</td>
<td><code>Ctrl</code> + <code>Shift</code> + <code>Alt</code> + <code>T</code> </t
>
</tr>
<tr>
<td>万能插入</td>
<td><code>Alt</code> + <code>Insert</code> </td>
</tr>
<tr>
<td>自动完成</td>
<td><code>Ctrl</code> + <code>Shift</code> + <code>Enter</code> </td>
</tr>
<tr>
<td>全局搜索</td>
<td><code>Shift</code> + <code>Shift</code> </td>
</tr>
<tr>
<td>命令查找</td>
<td><code>Ctrl</code> + <code>Shift</code> + <code>A</code> </td>
</tr>
<tr>
<td>模板触发</td>
<td><code>Template/Postfix</code> + <code>Tab</code> </td>
</tr>
<tr>
<td>智能选取</td>
<td><code>Ctrl</code> + <code>W</code> </td>
</tr>
<tr>
<td>切换标签</td>
<td><code>Ctrl</code> + <code>Tab</code> </td>
</tr>
</tbody>
</table>

```