

Atom 快速入门

第一章 简介

Atom有啥用

有太多的文本编辑器可以选择，为什么要花时间学习使用Atom？

编辑器诸如Sublime、TextMate非常便捷，可惜扩展性有限。另一方面，Emacs和Vim提供了极大的灵活性，但并不合适，只能针对特殊目的脚本语言做定制。

Atom的核心

原生Web

JavaScript，支持C++

Web Tech: The Fun Parts

开源文本编辑器

<https://github.com/atom>

使用Atom

我们已经介绍了Atom的基础，接下来准备学习如何使用。

本章将学习如何查找和安装新packages以便增加新的功能，如何查找和安装新主题，如何应更高级的方式操作文本，如何按照你的想法定制编辑器，如何使用Git做版本控制。

在本章结束时，你应该已经创建好一个环境，可以随时更改、增加和移动文本，就像master一样。

Atom Packages

首先介绍Atom的包系统。Atom只是一个非常基本的功能核心，集成了许多有用的包来增加新特性。

事实上，有90多个包构成了Atom最基本的功能。例如，启动时看到的欢迎屏幕，拼写检查，主题以及模糊搜索，都是单独维护的包，并且使用相同的API供你访问。

这意味着packages变得非常强大，可以改变任何东西，从整个界面的视觉效果，甚至到核心功能的基本操作。

安装一个新package，需要打开**Settings**视图的**Install**选项卡，使用`Ctrl+I`，在Install Packages下面输入包名进行搜索。列出的packages已经发布到<http://atom.io/packages>（Atom包的官方注册地址）。

点击Install按钮，下载并安装，编辑器即可获得包所提供的功能。

Package 设置

若package已安装好，它会出现**Settings**视图下的**Packages**选项卡下。

点击Settings，可以对相应的包进行设置。

如果新版本的package发布了，可以在Updates选项卡里对它升级。

Atom 主题

在**Settings**视图里还可以安装新主题。包括UI和语法主题，可以在**Install**那里搜索主题，就如搜索新packages一样。

命令行

还可以在命令行使用 **apm** 来安装包和主题。

在控制台输入如下命令检查是否已安装 **apm**

```
C:\> apm help install
```

若没有看到相应安装信息，可以参阅 [安装Atom](#) 如何安装 **apm** 命令。

安装packages可以使用 `apm install` 命令

- `apm install <package_name>` 安装最新版本
- `apm install <package_name>@<package_version>` 安装指定版本

例如 `apm install emmet@0.1.5` 安装 **Emmet** 包。

使用 `apm search` 来搜索新包。

使用 `apm view` 查看特定包的信息。

移动光标

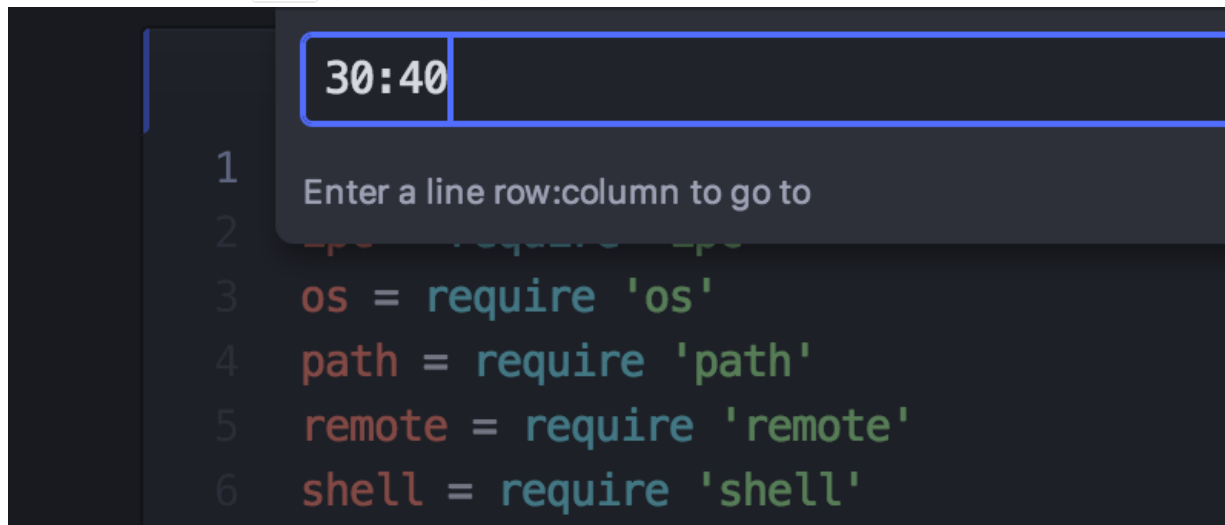
尽管使用鼠标和方向键在Atom里移动非常容易，但使用一些快捷键可以让你手不离键盘而且导航得更快。

Atom支持所有标准Windows光标移动键组合。使用方向键上下左右移动一个字符。

除了单字符移动，还有下列移动快捷键：

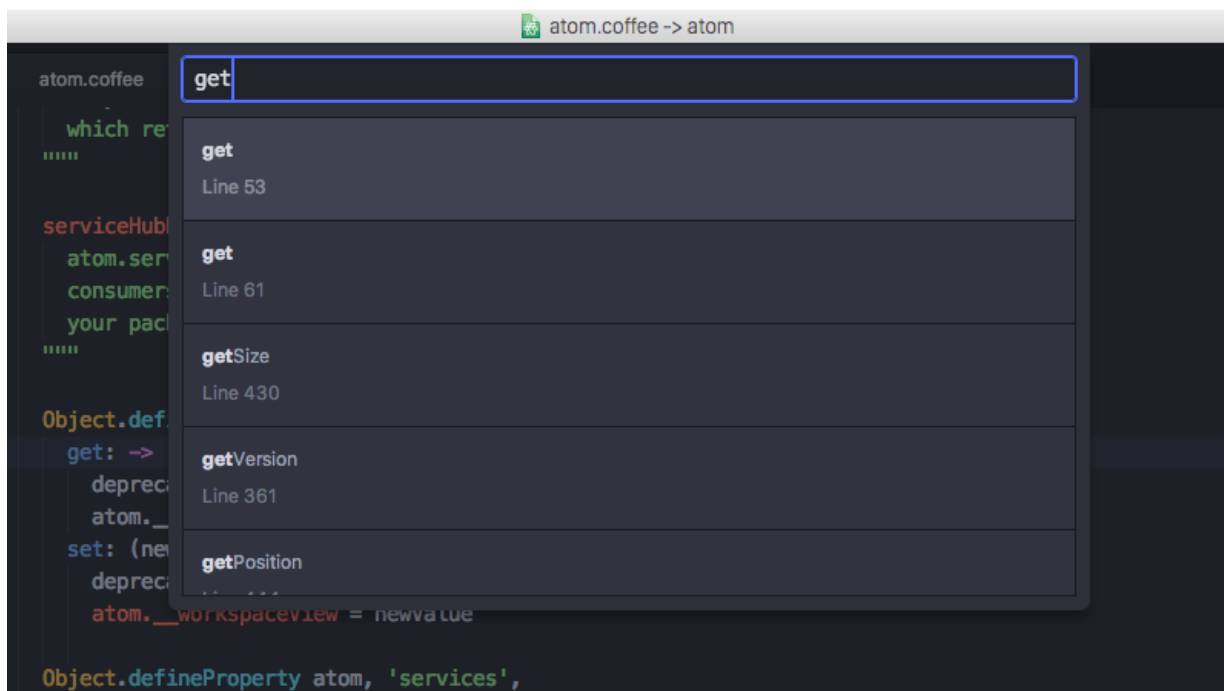
- `Ctrl+Left` - 移到单词开头
- `Ctrl+Right` - 移到单词末尾
- `Home` - 移到当前行的开头
- `End` - 移到当前行的末尾
- `Ctrl+Home` - 移到文件的顶部
- `Ctrl+End` - 移到文件的底部

移动到指定行或列，使用 `Ctrl+G` 弹出一个输入框输入行号，使用 `row:column` 语法跳到指定字符。



按标志导航

在**Symbols**视图中，按 `Ctrl+R` 可以跳到一个标志例如方法定义。它会打开一个标志列表（当前文件），按 `Ctrl+T` 可以模糊匹配。



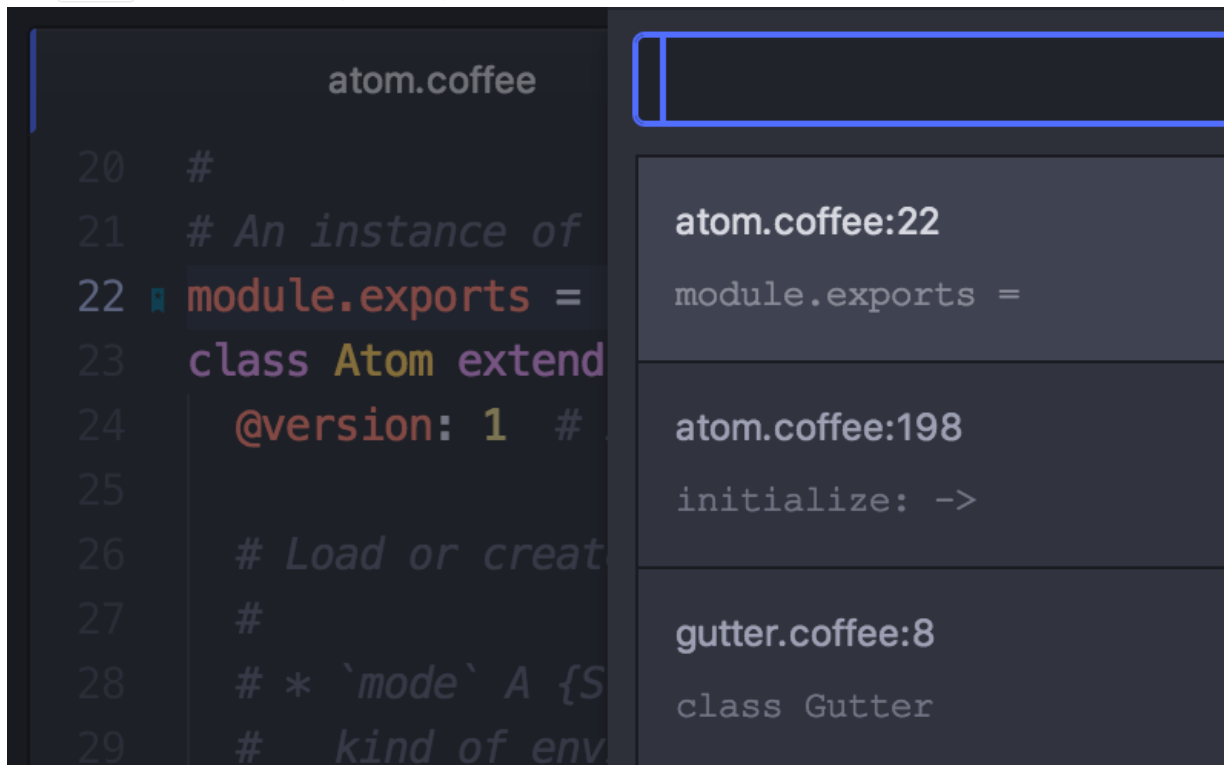
书签

Atom可以对项目中指定行添加书签，从而可以快速跳回到这些行。

按下 `Alt+Ctrl+F2`，Atom会给当前行添加标签，在行的左侧会看到一个书签的标记（如图）。

按下 `F2`，Atom会跳到当前文件中的下一个书签。使用 `Shift+F2` 则可以在书签之间来回挑战。

按下 `Ctrl+F2` 可以列出项目的所有书签，过滤并跳到指定位置。



PS：书签功能通过 `bookmarks` 包实现。

选择文本

文本选择有许多操作方式，例如范围选择、缩进和搜索，标记文本如引用和括住。

选择映射许多移动命令。其实与移动命令的键位相同，只是假如了 `Shift` 键。

- `Shift+up` - 上选
- `Shift+Down` - 下选
- `Shift+Left` - 选中前一个字符
- `Shift+Right` - 选中下一个字符
- `Ctrl+Shift+Left` - 选到单词开头
- `Ctrl+Shift+Right` - 选到单词结尾
- `Shift+End` - 选到行尾
- `Shift+Home` - 选到行首
- `Ctrl+Shift+Home` - 选到文件首部
- `Ctrl+Shift+End` - 选到文件尾部

除了光标移动选中命令，有些命令可以选择一块内容。

- `Ctrl+A` - 选择整个文件内容
- `Ctrl+L` - 选择整行

编辑和删除文本

现在让我们来修改文本。当然我们可以顺序插入字符，但也有其他方式删除和操作文本。

基本操作

下面是一些基本的文本操作快捷键，从移动文本行到复制文本行。

- `Ctrl+J` - 把下一行串连到当前行尾
- `Ctrl+Up/Down` - 上移或下移当前行
- `Ctrl+Shift+D` - 复制当前行
- `Ctrl+K` `Ctrl+U` - 当前单词转大写
- `Ctrl+K` `Ctrl+L` - 当前单词转小写

Atom内置了段落排版，在给定长度自动换行的功能。按下 `Alt+Ctrl+Q` ,可以重新排版选择的内容（行长度不超过 `editor.preferredLineLength` ），若未选择中，当前段落会重新布局。

删除和剪切

快捷键如下：

- `Ctrl+Shift+K` - 删除当前行
- `Ctrl+Backspace` - 删除到单词开头
- `Ctrl+Delete` - 删除到单词末尾

多光标选择

Atom支持多光标，在操作文本列表时非常有用。

- `Ctrl+Click` - 在点击位置插入光标
- `Alt+Ctrl+Up/Down` - 在当前光标的上面或下面插入光标
- `Ctrl+D` - 选择下一个与当前选中单词相同的文本内容
- `Alt+F3` - 选中所有与当前选中单词相同的文本内容

使用这些命令，可以在文档的多个地方放置光标，一次性高效地执行相同的命令。



这种方式对许多重复任务非常方便，例如重命名变量名、修改部分文本的格式。该操作几乎可以与任何插件、命令一起使用。

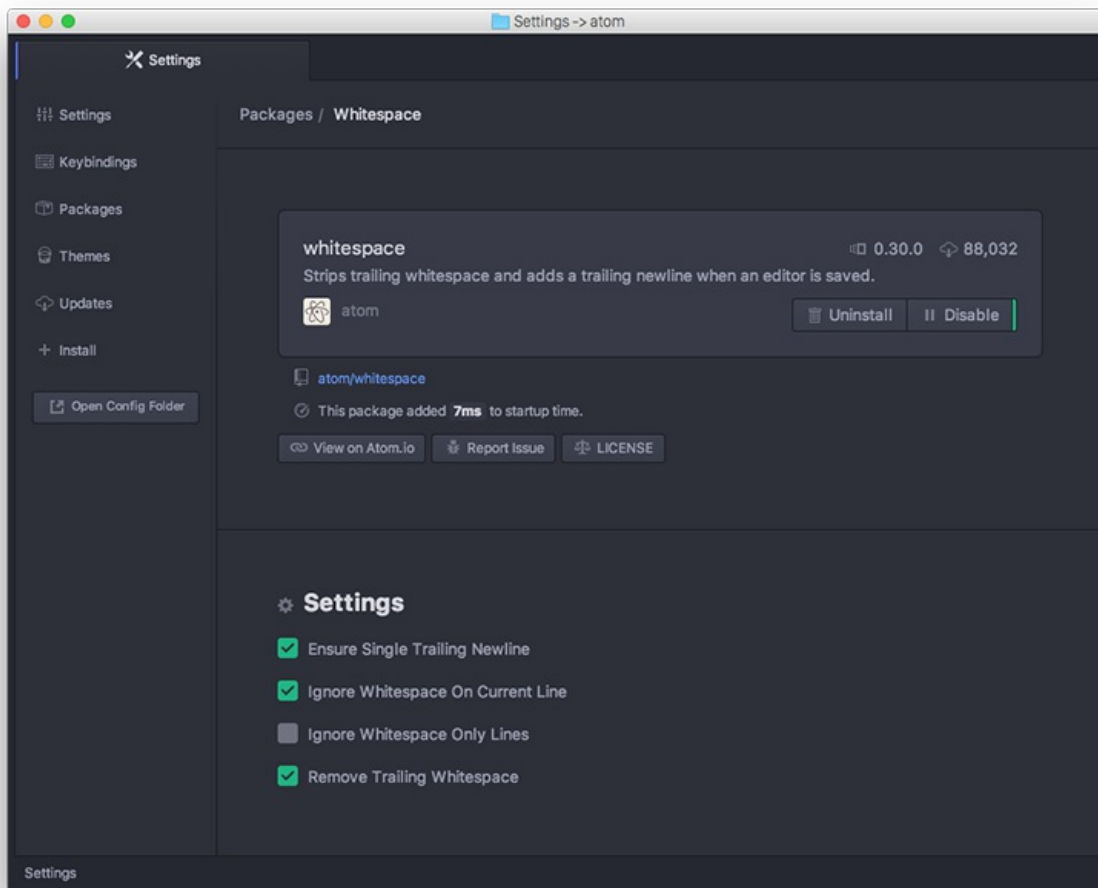
按住 `Ctrl` 键就可以同时选择多个文本区域啦。

空白字符

Atom有多个命令可以管理文档中的空白字符。例如前导空格与制表符互转。

倘若文档中含有混合的空白字符，这些命令有助于文件的规范化。空白符命令没有快捷键。

空白字符命令通过 `atom/whitespace` 包实现。空白符命令的设置和管理位于 `whitespace` 包页面。



“移除尾部空白字符”选项是默认的。也就是说，每次你报错Atom打开的文件时，他会剔除文件的所有尾部空白字符。默认情况下，Atom会保证文本文件有一个尾部换行。这些选项都可以禁用。

括号

Atom的括号处理智能易用。

它默认高亮常见括号对。同时会自动完成成对的标点符号。如果选中一段内容，输入开括号或引号，则自动添加关闭括号或引号。

其他一些与括号有关的命令：

- `Ctrl+M` - 跳到第一个与光标相邻括号匹配的地方，若无相邻括号，跳到最近的关闭括号处。
- `Alt+Ctrl+M` - 选中当前括号内所有的内容
- `Alt+Ctrl+.` - 闭合当前XML/HTML标签

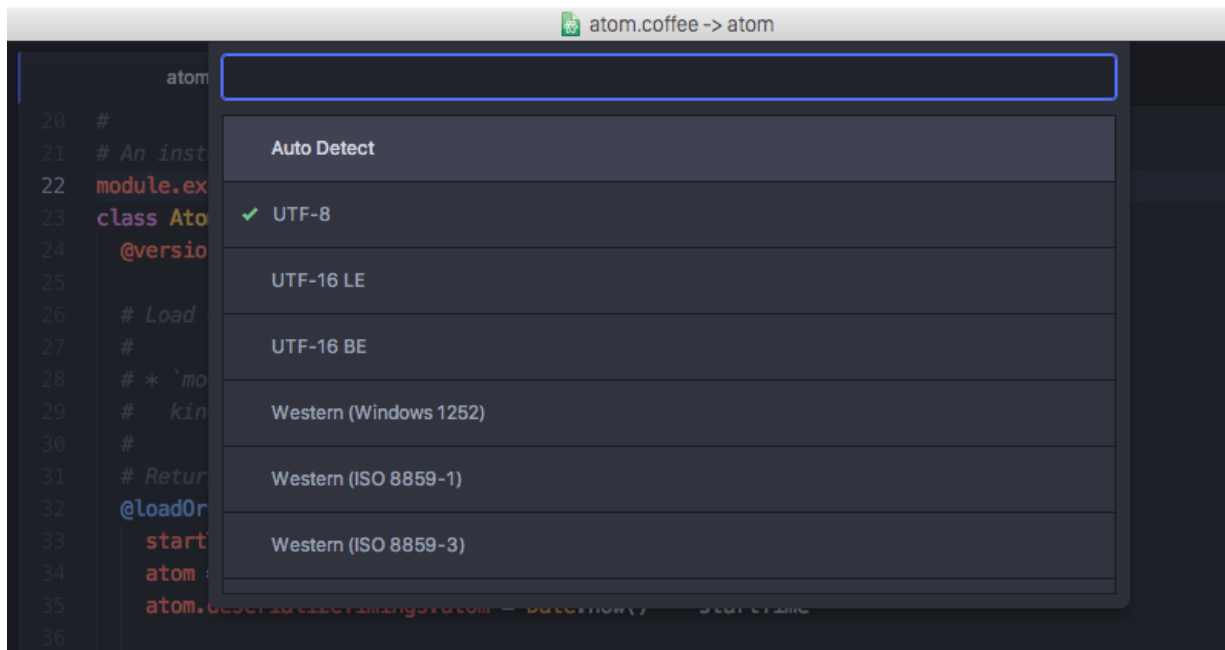
括号功能实现在 `bracket-matcher` 包。可以到该包下配置括号的相关行为。

Encoding

Atom提供了最基本的文件编码支持，用来处理非UTF-8编码文件。

- `Ctrl+Shift+U` - 打开文件编码切换菜单

打开文件时，Atom自动检测文件编码。若无法识别，则默认编码为 **UTF-8**，这也是所有新建文件的默认编码。



PS: 编码选择器的实现为 [encoding-selector](#) 包。

查找替换

快速且简单：

- `Ctrl+F` - 在buffer中查找
- `Ctrl+Shift+F` - 搜索整个项目

PS: 查找替换的实现为 [find-and-replace](#) 包。

代码片段

代码段是一种强大的方式用来快速生成通用代码语法。

其想法是，输入几个字符如 `habtm` 然后按下 `Tab` 键，它会自动扩展成 `has_and_belongs_to_many`

许多核心和社区包提供了自带的代码片段。例如 `language-html` 包支持HTML语法高亮，提供了大量的代码段来创建各种HTML标签。在Atom中创建一个HTML文件，输入 `html` 按下Tab键，它会扩展成：

```
<html>
  <head>
    <title></title>
  </head>
  <body>

  </body>
</html>
```

许多代码段有好几个输入焦点，通过Tab键可以来回切换。

按下 `Alt+Shift+S` 可以查看当前文件类型支持的所有代码片段。

自定义代码片段

在 `%USERPROFILE%\atom` 目录下有个叫 `snippets.cson` 的文本文件，可以添加定制代码段，Atom在启动时会加载它们。可以选择 **File > Snippets** 菜单打开该文件。

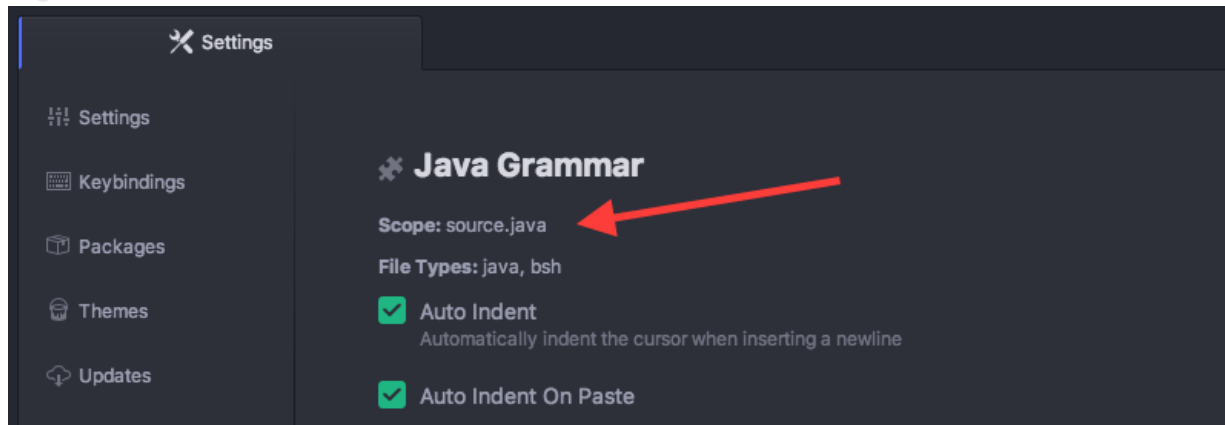
代码段格式

基本的代码片段格式看起来如下：

```
'source.js':  
  'console.log':  
    'prefix': 'log'  
    'body': 'console.log(${1:"crash"});$2'
```

最左侧的keys是选择器，即代码段在哪里激活。最简单的方式，打开需要添加代码段的语言的相应包，查看 **Scope** 信息。

例如，想为Java文件添加代码片段，在 **Settings** 视图查看 **language-java** 包，看到 **Scope** 为 *source.java*。那么顶级代码段key，需要在其前面加上 `.` 字符（类似CSS的class选择器）。



第二个级别的keys是片段名称。用于代码段菜单描述该代码段。可以任意命名。

在片段名下是 `prefix` 用来触发代码片段，`body` 表示当触发代码段时插入的内容。

每个带数字的 `$` 是Tab占位符。一旦代码段被触发，按Tab键可以在占位符间来回切换。

编号相同的占位符会创建多个光标。

上述的例子添加一个 `log` 片段到JavaScript文件，它会扩展成：

```
console.log("crash");
```

字符串 `"crash"` 初始被选中，按下Tab键，光标会置于 `;` 之后。

注意

☐ TODO <http://flight-manual.atom.io/using-atom/sections/snippets/>