

创建package流程

1. 来到workspace的src目录下

```
1 | cd first_ws/src
```

2. 通过catkin创建package

```
1 | catkin_create_pkg demo_cpp roscpp rospy rosmmsg
```

!!!tip

`catkin_create_pkg` 是创建package的命令。运行以上命令，会新建好package的目录，并且在目录中创建 `CMakeLists.txt`，`package.xml`，`src`，`include` 等文件和目录

```
1 | 第一个参数`demo_cpp`是指创建的package名称，可以根据自己的实际需求去设定。
2 |
3 | 后面的参数`roscpp`，`rospy`，`rosmmsg`是指当前创建的这个package需要提供哪些环境依赖。
4 |
5 | `roscpp`是对c++的一种依赖，有了它就可以用c++开发ros程序。
6 |
7 | `rospy`是对python的一种依赖，有了它就可以用python开发ros程序。
8 |
9 | `rosmmsg`是Node间通讯的消息协议依赖，有了它就可以让节点间进行消息通讯。
10 |
11 | **注意：通常在开发过程中，这三个依赖都是必备的。**
```

使用Clion开发package

1. 启动clion

开启命令行工具，来到工作空间目录下，设置开发环境。

```
1 | cd first_ws
2 | source devel/setup.bash
```

!!!note

此操作非常重要。 `devel`目录中的`setup.bash`是用于开发环境中，方便找到开发依赖的。

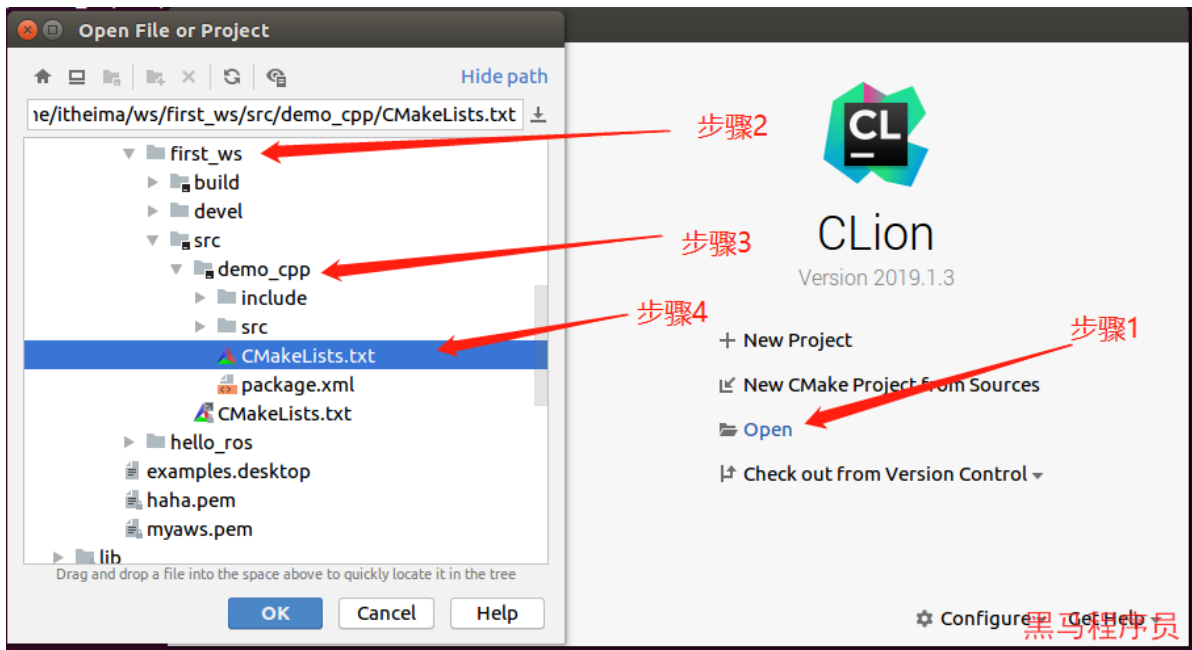
来到clion的安装目录下，通过命令启动clion

```
1 | cd ~/clion/bin
2 | ./clion.sh
```

!!!note

笔者的clion安装目录在`~/clion`，大家根据实际情况，来到自己clion的安装目录，并且进入到 `bin` 目录，因为启动文件在 `bin` 目录下。

2. 使用clion打开package



!!!note

clion启动后，首先点击open，

- 1 然后找到工作空间，在工作空间的src中找到要打开的package，
- 2
- 3 在package中找到`CMakeLists.txt`，选中双击，
- 4
- 5 此时点击`open as project`就可以打开package做开发了。