

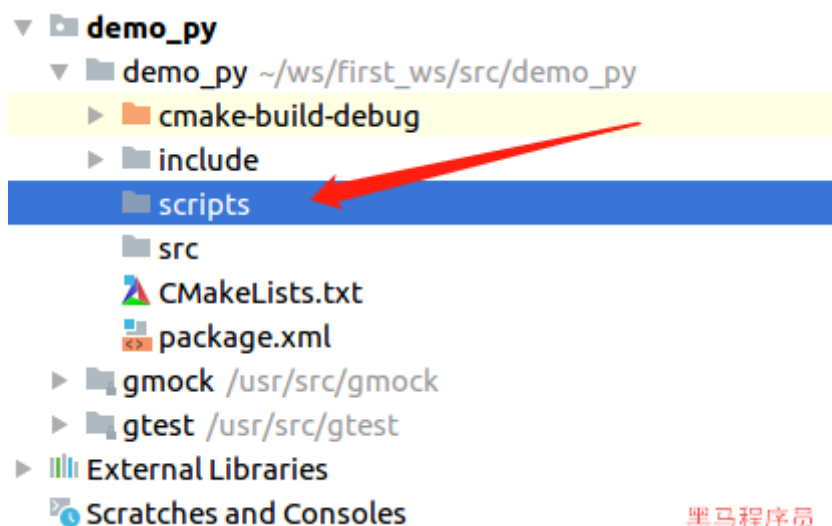
Clion的Python环境

为了演示第一个程序，首先我们按照前面的例子，在workspace中创建一个demo_py的package，我们基于这个demo_py进行讲解开发。

使用clion打开demo_py。

1. 创建scripts目录

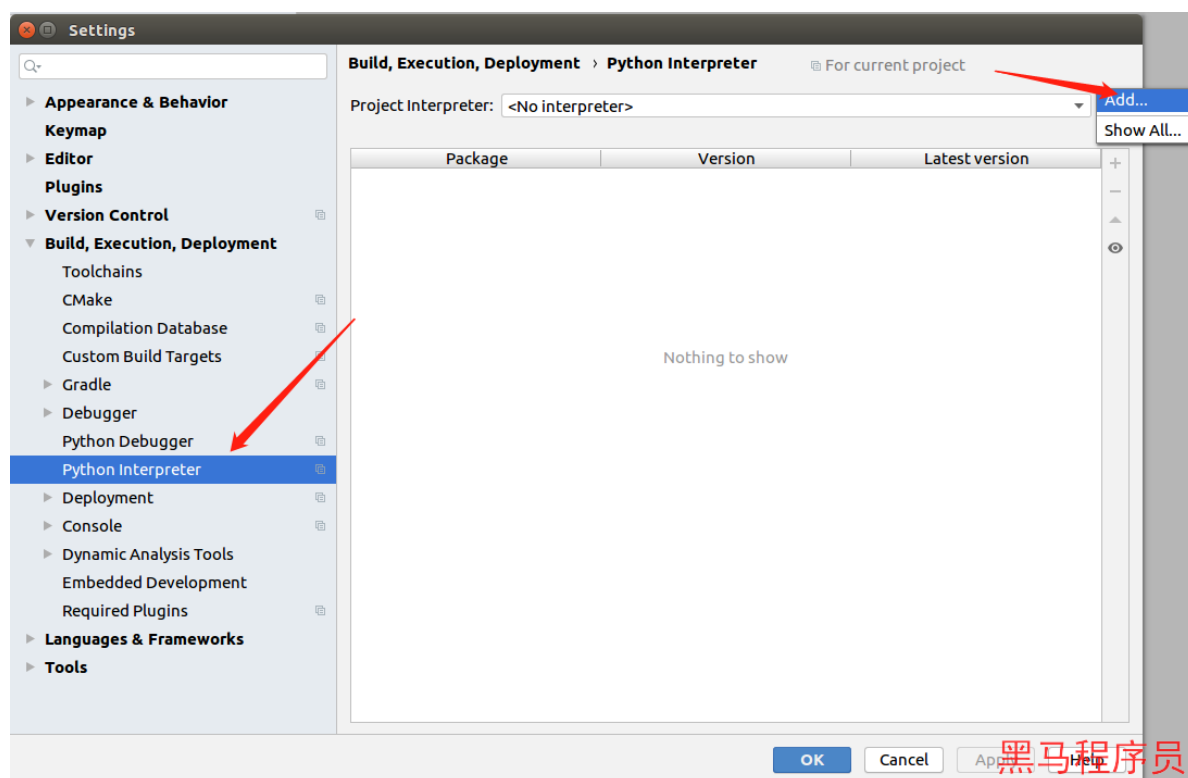
在demo_py的目录中创建scripts目录，用于写python代码。



黑马程序员

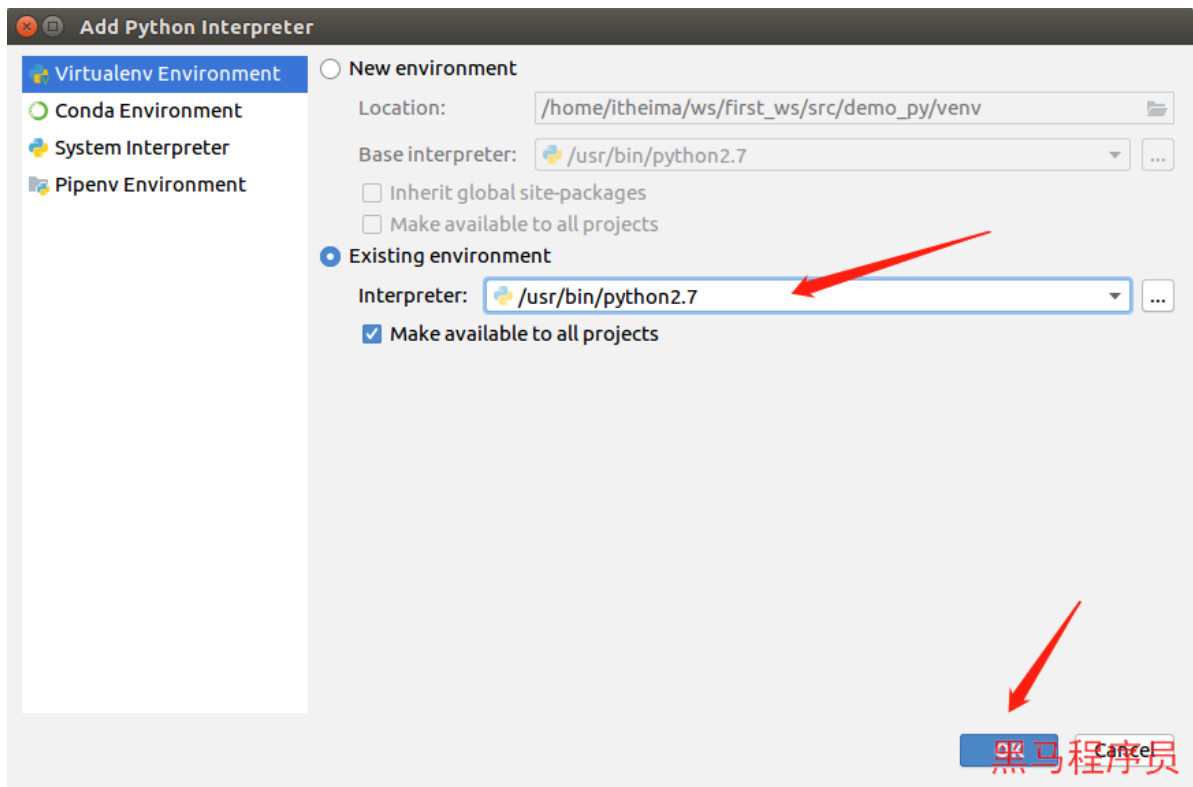
2. 配置Python环境

打开clion的setting，来到Build, Execution, Deployment的Python Interpreter下，点击设置按钮，点击添加。



黑马程序员

打开python环境设置按钮，添加python环境



!!!tip

特别要注意的是，目前ROS Melodic版本还不支持python3，要等到ROS N版才会支持。

1 | 因此，我们选择环境的时候**选择python2.x版本**。

编写Python代码

1. 代码编写

在scripts目录下新建 `hello.py` 文件，代码如下：

```
1 | #!/usr/bin/env python
2 | # coding:utf-8
3 | import rospy
4 |
5 | if __name__ == '__main__':
6 |     # 创建节点
7 |     rospy.init_node("pyhello")
8 |     print("hello ros python")
```

!!!note

`#!/usr/bin/env python` 让系统知道当前是可执行的python脚本，避免系统当作shell脚本执行。

1 | `# coding:utf-8`解决编码问题

2. 可执行权限修改

将 `hello.py` 文件做成可执行文件

```
1 | sudo chmod +x hello.py
```

!!!note

默认编写的 `py` 文件是不具备执行权限的。

```
1 | 我们需要手动讲文件权限修改为可执行。
```

调试运行代码

1. 项目编译

启动命令行，来到工作空间目录下，编译项目

```
1 | cd first_ws
2 | catkin_make
```

!!!tip

编译的过程，会将新建的package进行编译。

```
1 | 在编写`python`版本程序时，这一操作可以省略。
2 |
3 | 但是建议养成良好习惯，还是进行编译。
```

2. 启动ROS Master

打开新的命令行，启动ROS master

```
1 | roscore
```

!!!tip

ROS节点启动前，必须启动ros master

3. 运行节点

启动编译好的node节点，有**两种方式**可以运行。

1. 通过clion工具直接运行

```
1  #!/usr/bin/env python
2  import rospy
3
4  if __name__ == '__main__':
5
6      rospy.init_node("hello")
7      print("hello ros python")
```

黑马程序员

2. 通过命令行运行

1. 打开新的命令行，来到工作空间目录下，source一下开发环境

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

!!!tip

做这一步的操作是为了让roslaunch 命令找到刚刚编译好的node

2. 通过命令运行编译好的node

```
1  roslaunch demo_py hello.py
```

!!!tip

roslaunch 命令后跟第一个参数是package的名称，第二个参数是节点名称