

Publisher创建流程

1. 设置环境和编码

```
1  #!/usr/bin/env python
2  #coding:utf-8
```

!!!warning

在编程过程中，必须加上以上两行代码。

```
1  `#!/usr/bin/env python` 用来表示当前为python脚本，如果不加，系统会默认为bash脚本
2
3  `#coding:utf-8` 提供中文支持
```

2. 创建节点

```
1  rospy.init_node(nodeName)
```

3. 创建发布者

```
1  publisher = rospy.Publisher(topicName, String, queue_size=1000)
```

!!!tip

第一个参数为topic名称

```
1  第二个参数为发布的消息类型
2
3  第三个参数为topic中消息队列最多的数量。
```

4. 定期发布消息

```
1  rate = rospy.Rate(10)
2  count = 0
3  while not rospy.is_shutdown():
4      # 发布消息
5      publisher.publish("hello %d" % count)
6      rate.sleep()
7      count += 1;
```

完整的示例代码

```
1  #!/usr/bin/env python
2  #coding:utf-8
3
4  import rospy
5  from std_msgs.msg import String
6
```

```
7  if __name__ == '__main__':
8      nodeName = "pypublisher"
9      topicName = "pytopic"
10     # 初始化节点
11     rospy.init_node(nodeName)
12
13     # 创建发布者
14     publisher = rospy.Publisher(topicName, String, queue_size=1000)
15
16     rate = rospy.Rate(10)
17     count = 0
18     while not rospy.is_shutdown():
19         # 发布消息
20         publisher.publish("hello %d" % count)
21         rate.sleep()
22         count += 1;
```

调试发布者

调试Publisher主要是查看是否有发送数据，也就是提供一个订阅的调试工具。ROS提供了命令行工具和图形化工具进行调试。

1. 通过rostopic工具进行调试

查看所有的主题

```
1 | rostopic list
```

打印主题所发布的信息

```
1 | rostopic echo pytopic
```

2. 通过rqt_topic工具进行调试

通过命令启动rqt_topic工具

```
1 | roslaunch rqt_topic rqt_topic
```

选中要调试的主题

Topic Monitor

Topic	Type	Bandwidth	Hz	Value
<input checked="" type="checkbox"/> /pytopic data	std_msgs/String string	151.43B/s	10.00	'hello 3808 '
<input type="checkbox"/> /rosout	rosgraph_msgs/Log			not monitored
<input type="checkbox"/> /rosout_agg	rosgraph_msgs/Log			not monitored

黑马程序员