
pyDemo Documentation

发布 *1.0*

wins

2016年03月03日

1	准备工具篇	3
1.1	PyQt学习资料	3
1.2	准备工作介绍	4
1.3	Eric配置	5
1.4	PyCharm配置	6
1.5	Sphinx写笔记	10
2	Python基础学习	15
2.1	Python 基础知识	15
3	PyQt初阶	17
3.1	PyQt初阶	17
4	相关代码Demo	19
4.1	src package	19
5	索引和目录	21
	Python 模块索引	23

准备以PyQt入门

1.1 PyQt学习资料

1.1.1 参考官网

- PyQt5 tutorial 英文 <http://zetcode.com/gui/pyqt5/>
- PyQt reference 英文 <http://pyqt.sourceforge.net/Docs/PyQt5/index.html>
- Python: <https://www.python.org/>
- Anaconda: <http://continuum.io/downloads>
- Pycharm: <https://www.jetbrains.com/pycharm/>
- Zeal / Dash: <https://zealdocs.org/>

1.1.2 中文教程

- 安装的PyQt5下面的例子: windows下安装目录下: PyQt5examples
- 本地安装zeal, 然后下载python和qt5的api参考手册 (qt5基于c++的, pyqt接口类似)
- 《Qt 学习之路 2》, c++版本的QT5的教程, 非常详细 <http://www.devbean.net/2012/08/qt-study-road-2-catalog/>
<https://github.com/joinAero/qt-study-road-2>
- PyQt5 GUI入门教程 百度搜索一下, 共计67页, 一些简单例子, 先做一下
- oschina上的pyqt的帖子 <http://www.oschina.net/question/tag/pyqt>
- 鱼c论坛的帖子 (PyQt5学习与交流): <http://bbs.fishc.com/forum.php?mod=collection&action=view&ctid=153>
<http://bbs.fishc.com/forum.php?mod=viewthread&tid=59816>
- Python菜鸟教程全目录, 里面有一大章介绍pyqt的 <http://blog.csdn.net/a359680405/article/details/42486689>
- PyQt5应用与实践 <http://www.cnblogs.com/skynet/p/4229556.html>
- 宁静的天空 python <http://www.cnblogs.com/ribavnu/category/612519.html>
- qt论坛 <http://bbs.qter.org/forum.php?mod=viewthread&tid=187&extra=page%3D1>
- Qt Model/View 的简单说明 <http://qimo601.iteye.com/blog/1534309>
- pyqt学习笔记 http://blog.sina.com.cn/s/articlelist_1283603943_1_1.html

1.1.3 源码项目

- pyqt的财务软件 <http://www.cnblogs.com/coderzh/archive/2009/06/28/1512654.html>
<https://github.com/coderzh/pygrid>
- 用python的交易员（与pyqt无关，但我的目标是做成他这样的）
<https://www.zhihu.com/people/traderusingpython/posts>
<https://github.com/vnpy/vnpy>
- 网易云音乐山寨版本 <https://github.com/HuberTRoy/NetEase>
- pyqt的烧写器的版本. <https://github.com/djstava/PyQt5Fastboot>
- 桌面便签 <http://my.oschina.net/zjuysw/blog/318352?fromerr=CVKV1k1Q>
<https://github.com/zjuysw/memo.git>
- 12306 pyqt5抢票 <https://github.com/lovesnow/ticket>
- pyqt写的聊天客户端 <http://my.oschina.net/fants/blog/144407?fromerr=9PkAdgoF>
<http://git.oschina.net/fantsme/oschina-for-pc>
- QMusic <http://dragonjdf.github.io/QMusic/>
<https://github.com/dragonjdf/QMusic>
- baoboa <https://github.com/baoboa/pyqt5>
- 分页查询的例子 <https://github.com/newdebug/TableViewPaging>

1.2 准备工作介绍

1.2.1 开发环境搭建

作为一个初学者，第一步就是搭建python的开发环境，因为我是要用pyqt包写界面的，在网上看了写文章，都是用Eric 5开发的. 所以本文主要说一下搭建环境的步骤：须先安装Python，再安装PyQt，再安装Eric或PyCharm

- Python 3.4 : 最原始的方式就是到Python官网 <https://www.python.org/downloads/> 下载python3.4，但实际上学习Python有很多依赖的软件包，后续一个个安装比较麻烦，本人是新手，看人推荐是安装的Anaconda， <https://www.continuum.io/downloads> 里面集成了很多常用的软件包，如numpy等，不用自己一个个装。
- PyQt5: 目前网上很多PyQt的例子还是用PyQt4写的，不过本人喜欢新东西，安装的是PyQt5，这个直接到官网上下载就可以了。PyQt提供了一个QT设计师的程序，可以用来图形化的设计界面UI，但生成的.ui文件要通过pyuic命令转换成.py代码，不过可以使用Eric或PyCharm开发IDE辅助转换就很方便了。PyQt安装会自动安装到andconda的Anaconda3Libsite-packagesPyQt5下
- Eric5 :到官网 <http://eric-ide.python-projects.org/> 下载最Eric5安装包，Eric是免费的，代码提示较弱。

参见：

具体见Eric章节

- PyCharm5 :到官网 <http://eric-ide.python-projects.org/> 下载最Eric5安装包，PyCharm是收费的，功能强大。

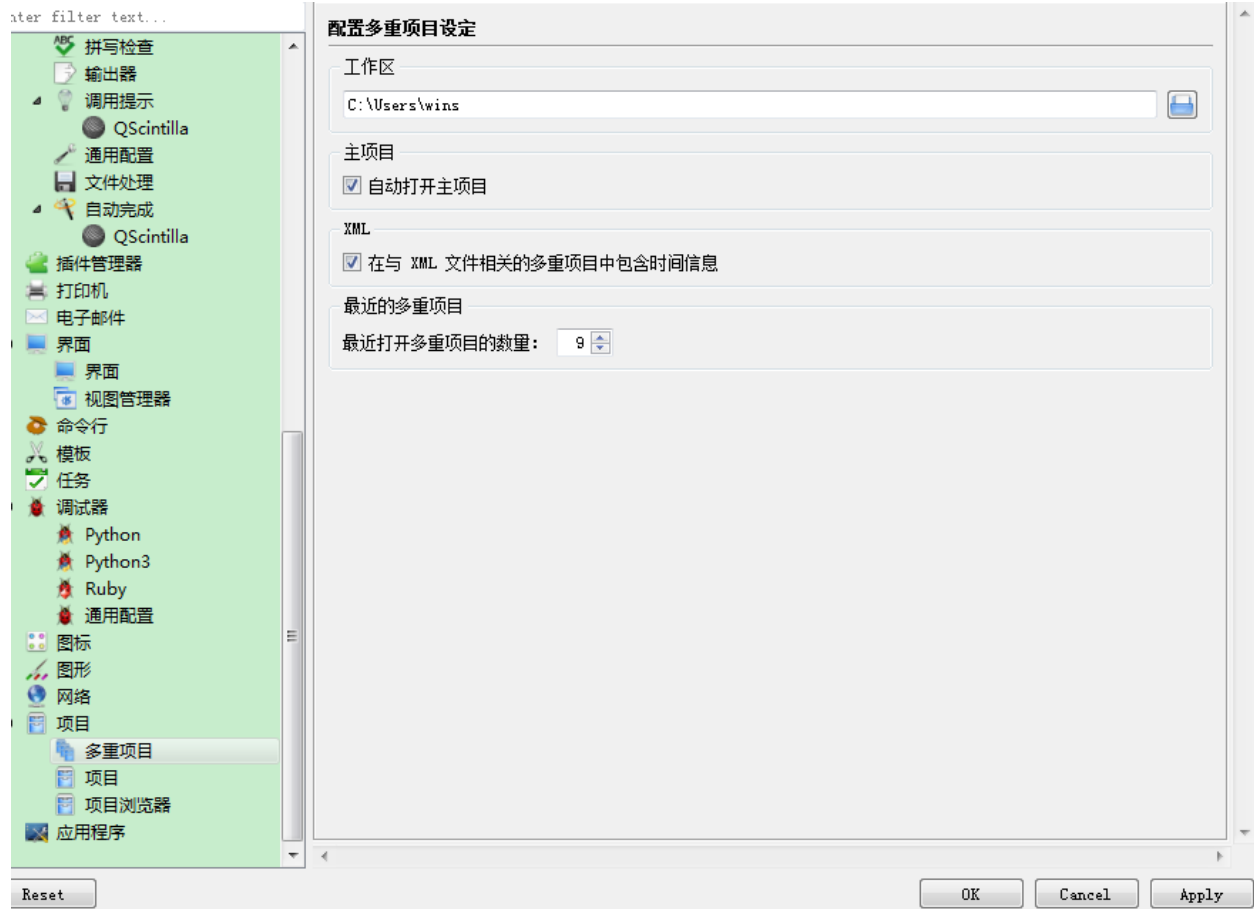
参见：
具体见PyCharm章节

1.2.2 相关模块包概要

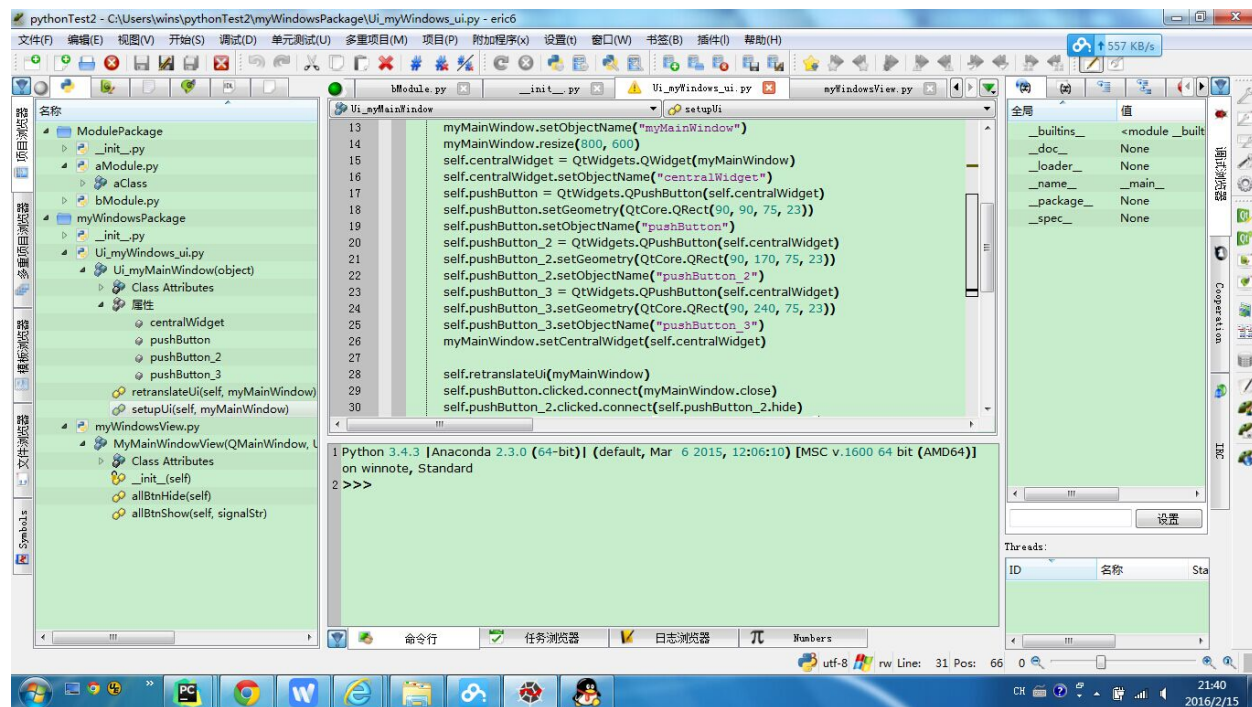
1.3 Eric配置

作为一个初学者，第一步就是搭建python的开发环境，因为我是要用pyqt包写界面的，在网上看了写文章，都是用Eric 5开发的，那么最初使用的就是Eric，Eric配置没有特别需要说明的

Eric有中文的语言包，可以下载语言包。运行eric6.pyw不显示console窗口，运行eric6.py会显示console窗口



Eric5运行的界面如下：



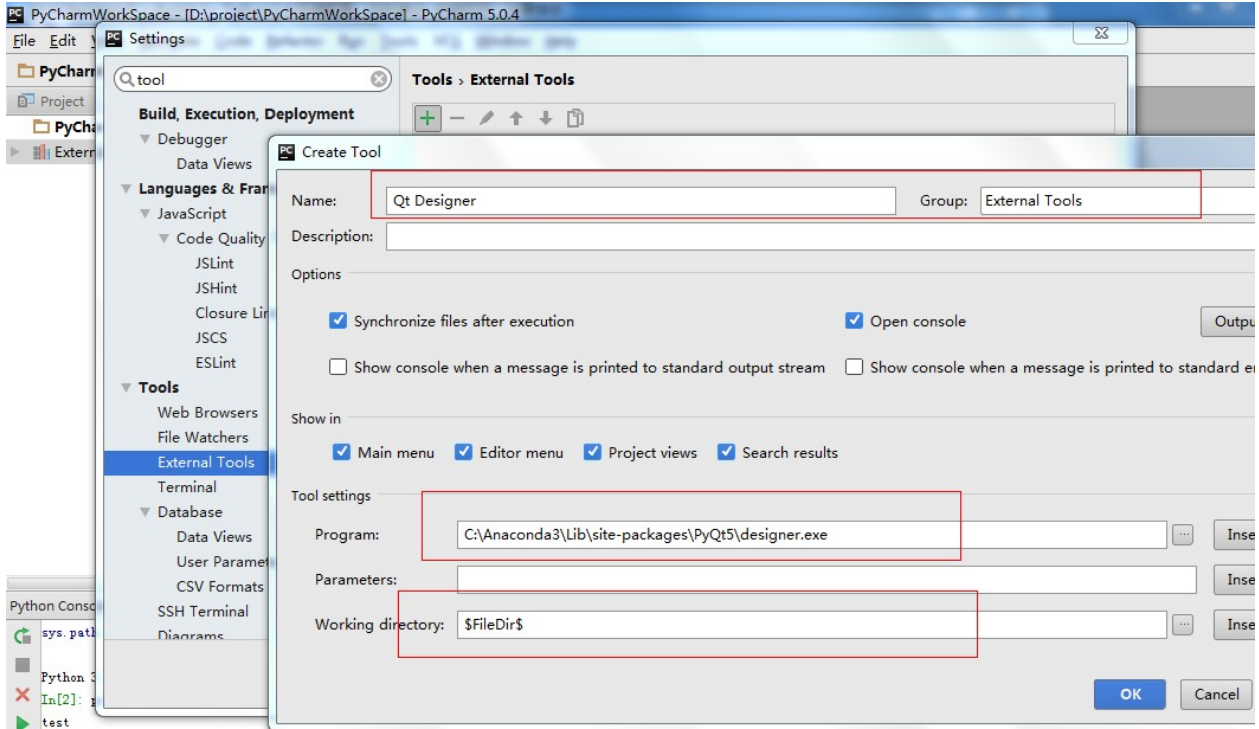
1.4 PyCharm配置

1.4.1 PyCharm介绍

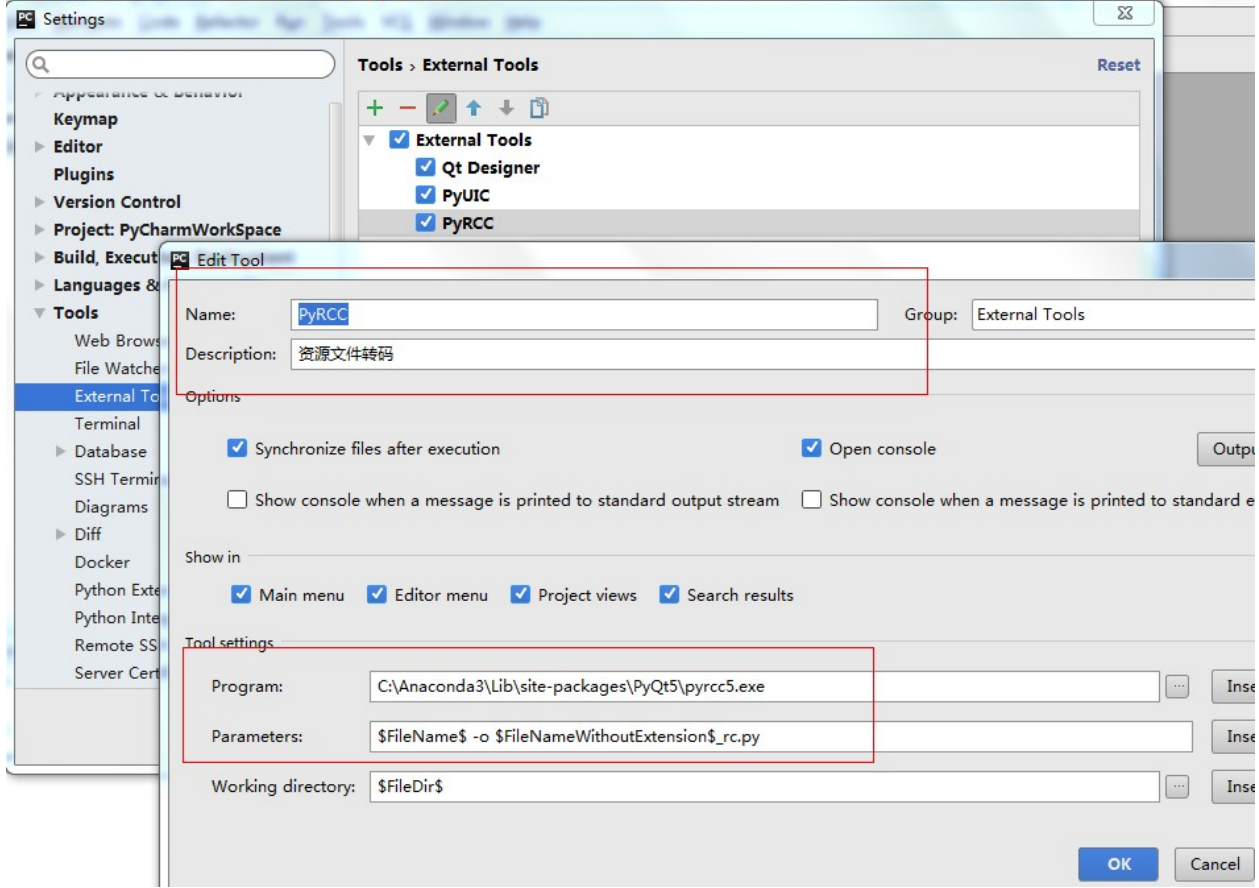
1.4.2 PyCharm配置

最初的几个PyQt例子都是用Eric编写的，但是Eric的代码提示功能不够强大，对于我这样的新手而言确实有点麻烦，看到很多人都推荐用PyCharm，用起来真心强大，推荐新手都从PyCharm开始学习，完善的代码提示能少走很多弯路。PyCharm调用Qt Designer是通过配置Extern tools实现的。本文配置了以下几个Extern Tools。

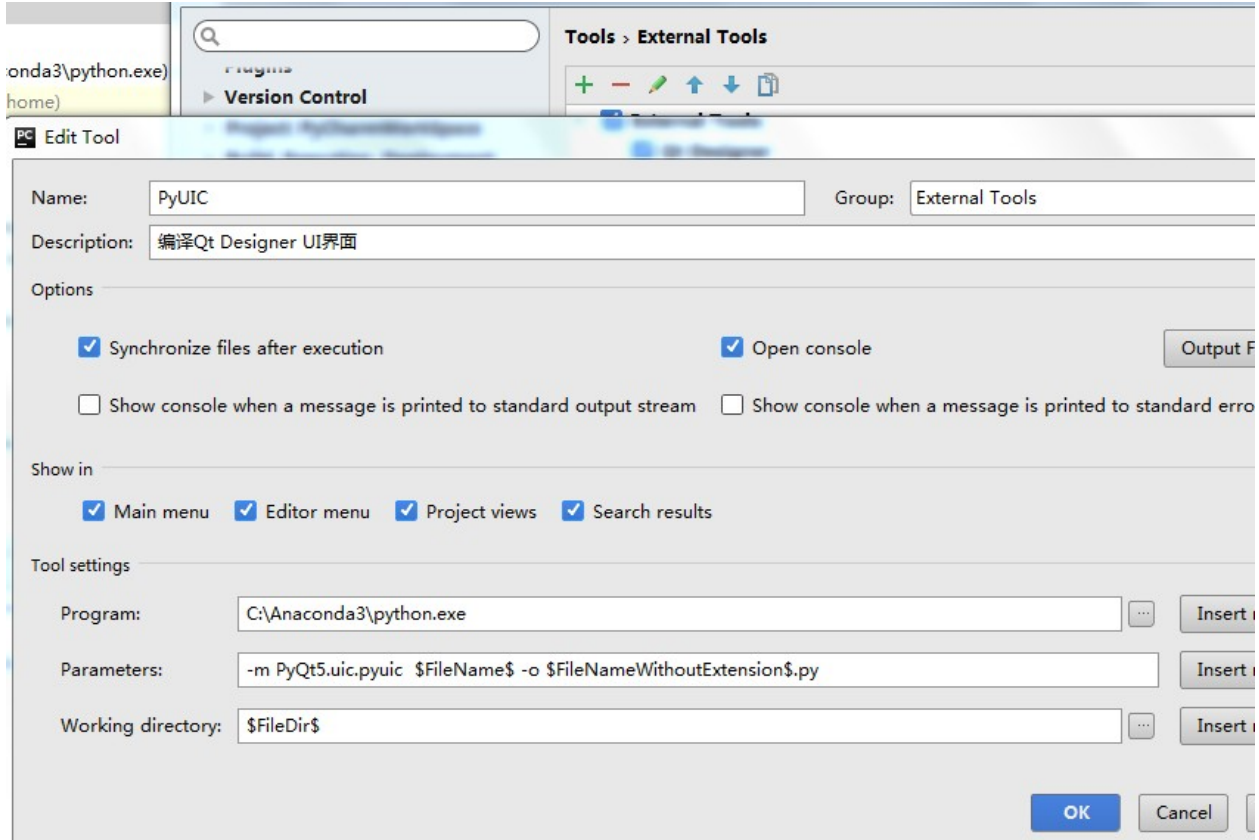
配置Qt Designer



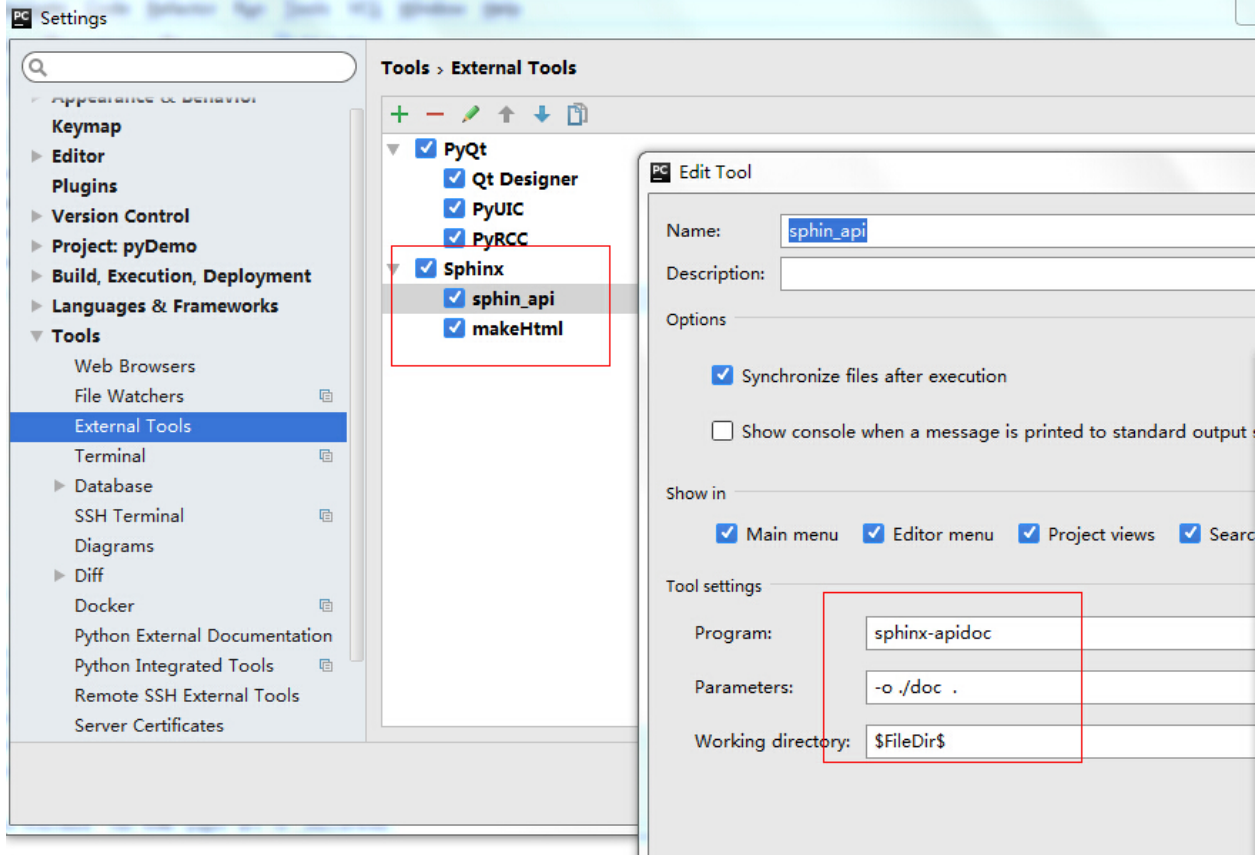
配置Pyqt 资源转换工具



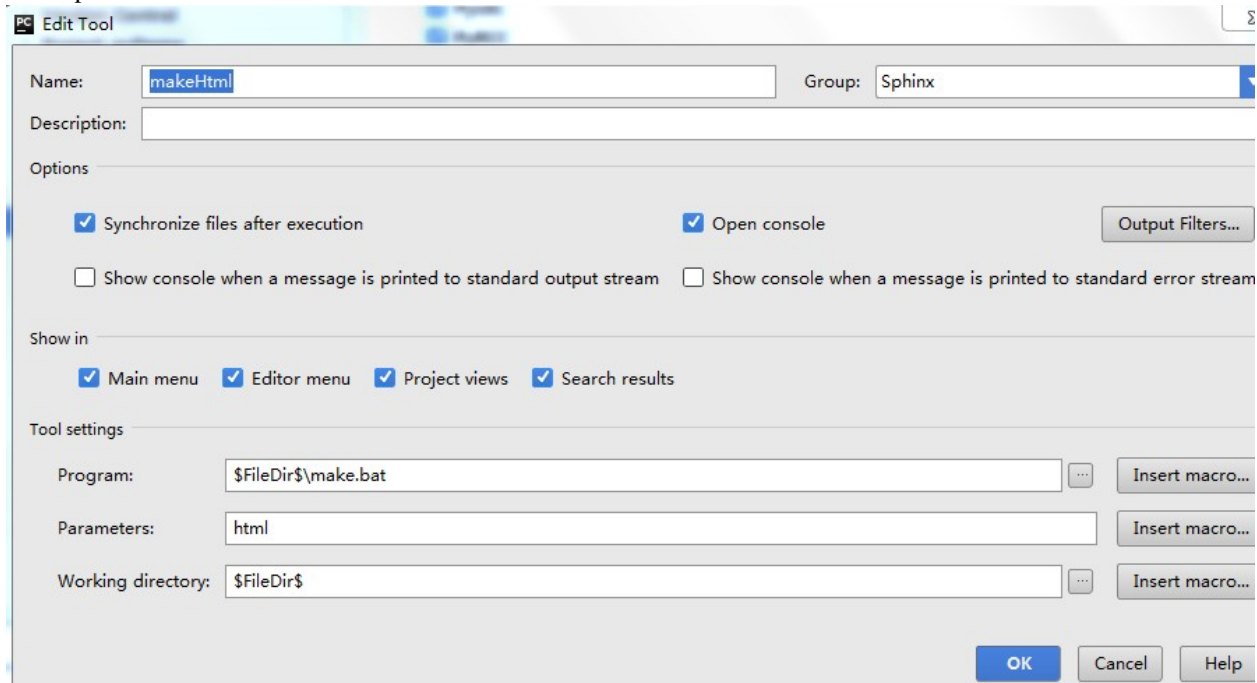
配置pyqt转换*.ui文件到*.py



配置Sphinx生成api接口



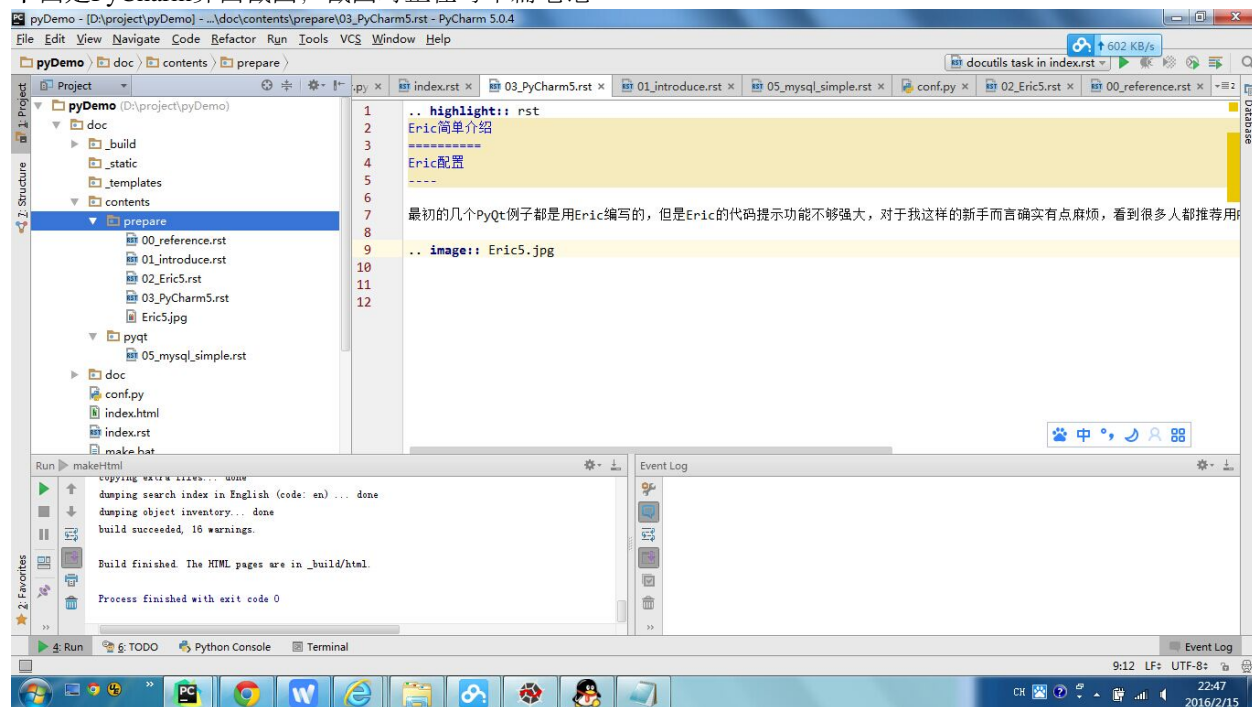
配置Sphinx 生成html文件



PyCharm快捷键:



下面是PyCharm界面截图，截图时正在写本篇笔记



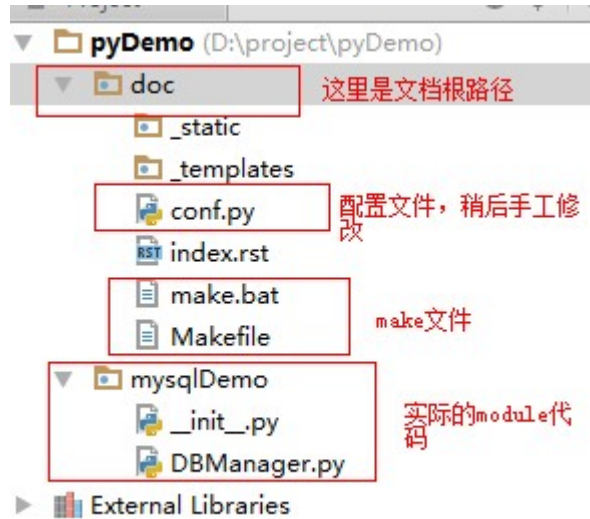
1.5 Sphinx写笔记

Sphinx写笔记体现在两个方面

- 能生成python项目里面代码的API文档，使用 **sphinx-apidoc**
- 能将手工编写的*.rst文件转换成html网页， **make html**

1.5.1 Sphinx配置

- 安装 Sphinx，使用 **\$ pip3 install Sphinx**
- 为项目生成文档，使用 **sphinx-quickstart**，本文中以 `d:/project/pyDemo/doc` 作为根目录 这里生成的时候要注意，向导要求指定的根目录我喜欢在项目目录下建一个doc文档，专门放sphinx的相关文件。向导运行完毕后，我建立的文档结构如下



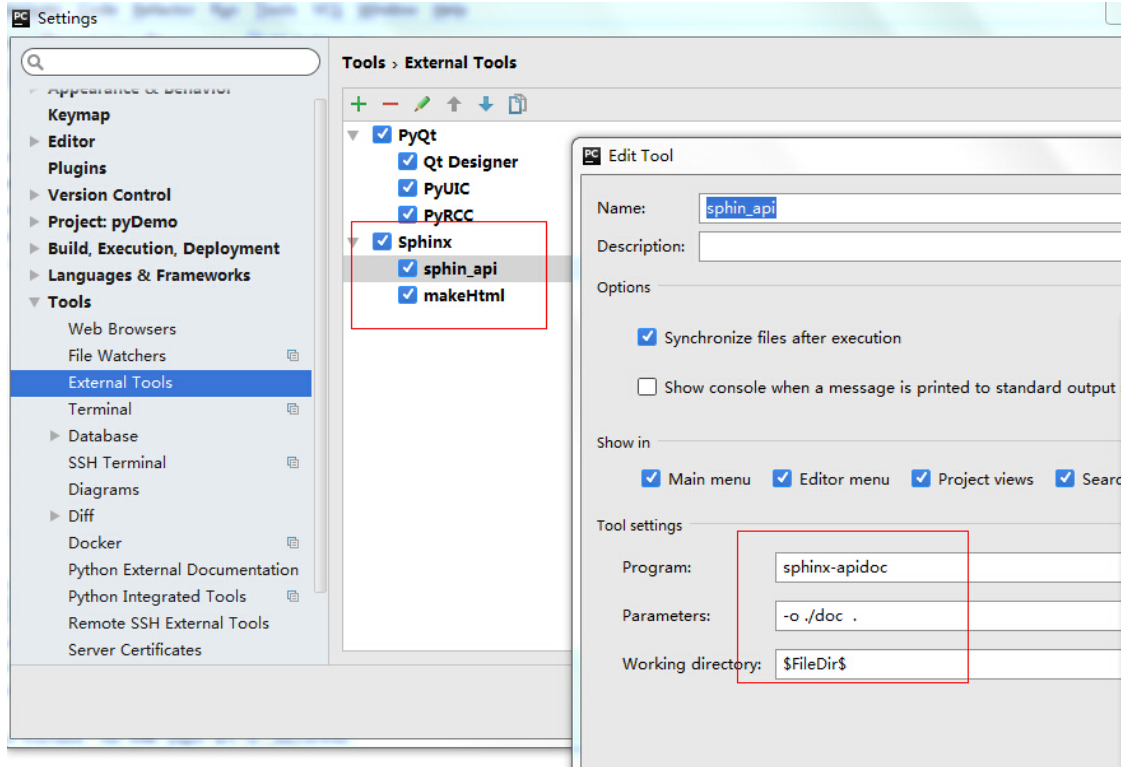
- 修改 doc/conf.py `sys.path.insert(0, os.path.abspath('../'))`
- doc目录下的index.rst作为主页文件，人工按需要修改，修改成自己需要的样子，把各个*.rst串起来，形成一本笔记
- 在项目路径下执行（此处是 `d:/project/pyDemo/`）**sphinx-apidoc -o ./doc .** 生成api文档（rst格式），不要遗忘最后的点，表示当前项目目录下寻找代码模块并生成API文档 如果不需要生成API文档，此步可以跳过
- 将rst文件转换成html网页文件，在项目路的文档径下执行如下命令（此处是 `d:/project/pyDemo/doc`）**make html**

1.5.2 Sphinx语法参考

- <http://pm.readthedocs.org/doc/sphinx.html#id11>
- <http://sphinx-doc-zh.readthedocs.org/en/latest/contents.html>
- <http://jwch.sdut.edu.cn/book/tool/UseSphinx.html>
- <http://avnpc.com/pages/writing-best-documentation-by-sphinx-github-readthedocs>
- <http://ju.outofmemory.cn/entry/64265>

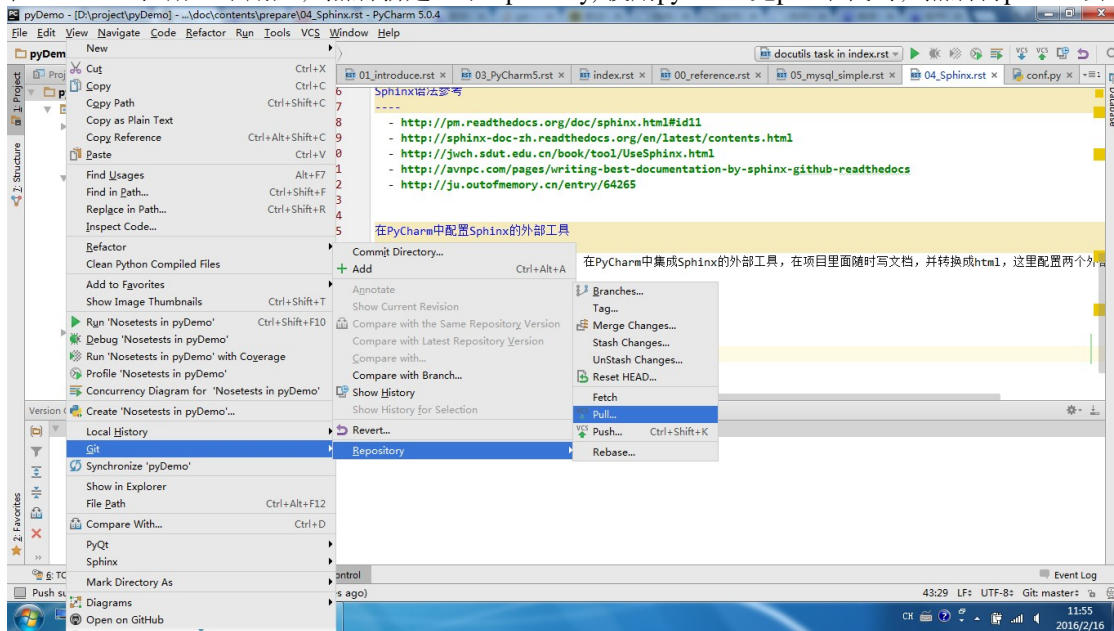
1.5.3 在PyCharm中配置Sphinx的外部工具

为了将在编码过程中将文档记录下来，在PyCharm中集成Sphinx的外部工具，在项目里面随时写文档，并转换成html，这里配置两个外部工具，见如下截图



1.5.4 上传项目到Github

在GitHub上注册一个用户，然后新建一个repository，使用pycharm先pull下代码，然后再push上去



首先在github->settings->Webhooks services->add services ->ReadTheDocs,激活这个选项。

The screenshot shows the GitHub repository settings for 'winsomexiao / PyDemo'. The 'Settings' tab is active. In the left sidebar, 'Webhooks & services' is selected. The main content area shows the 'Services' section, which includes a 'ReadTheDocs' integration. The 'Add service' button is highlighted. Below the screenshot is a 'Read the Docs' banner and an 'Import a Repository' section with buttons for 'Connect to GitHub' and 'Connect to Bitbucket'.

1.5.5 修改ReadTheDocs中的设置

需要修改ReadTheDocs中的`conf.py`的路径配置，设置为实际项目中的路径，本处就是 `doc/conf.py` 现在就可以在ReadTheDocs网页上看生成的笔记了。

2.1 Python 基础知识

2.1.1 Python模块

模块是一个包含了所有你定义的函数和变量以.py结尾的文件

- `__name__` 属性:
 - 如果模块是被导入, `__name__`的值为模块名字(文件名)
 - 如果模块是被直接执行, `__name__`的值为'`__main__`'

当我们编写Python库模块的时候, 我们往往运行一些测试语句。当这个程序作为库被import的时候, 我们并不需要运行这些测试语句。一种解决方法是在import之前, 将模块中的测试语句注释掉。而更优美的解决方法, 就是使用`__name__`。

- `__doc__` 属性:
 - 介绍模块的功能

aModule.py

```
class aClass:
    def whoCall():
        if __name__ == '__main__':
            print ("aClass selfcall: __name__: "+__name__)
        else:
            print( "extern call: __name__: "+__name__)
aClass.whoCall()
```

bModule.py

```
from aModule import aClass
aClass.whoCall()
```

直接运行aModule.py,输出如下

```
aClass selfcall: __name__: __main__
```

运行bModule.py,输出如下:

```
extern call: __name__: aModule
```

模块包



- 该文件夹中必须包含一个__init__.py的文件，提醒Python，该文件夹为一个模块包。__init__.py可以是一个空文件。
- 可以将功能相似的模块放在同一个文件夹中，构成一个模块包。通过import ModulePackage.aModule引入ModulePackage模块包中的module模块。

3.1 PyQt初阶

本文将自己学习PyQt的阶段记录下来

3.1.1 一个简单的连接mysql的例子

这个例子是在网上找的,改写成PyQt5和Python3.4下运行成功

```
1 from PyQt5 import QtGui
2 from PyQt5.QtSql import QSqlQuery, QSqlDatabase, QSqlTableModel
3 from PyQt5.QtWidgets import QWidget, QVBoxLayout, QTableView, QApplication
4 import sys
5
6 class DBManager():
7     def __init__(self):
8         self.db = QSqlDatabase.addDatabase("QMYSQL")
9         self.db.setHostName("localhost")
10        self.db.setUserName("winuser")
11        self.db.setPassword("winuser")
12        self.db.setDatabaseName("test")
13
14        def __del__(self):
15            self.db.close()
16
17        def execSQL(self, sql):
18            self.db.open()
19            self.query = QSqlQuery()
20            self.query.exec_(sql)
21            self.query.exec_("commit")
22
23        def close(self):
24            self.db.close()
25
26 class Model(QSqlTableModel):
27     def __init__(self, parent):
28         QSqlTableModel.__init__(self, parent)
29         self.setTable("t1")
30         # 这一步应该是执行查询的操作,不太理解
31         self.select()
32         # 数据更新的策略,详细可以查看Qt文档
```

```
33         self.setEditStrategy(QSqlTableModel.OnManualSubmit)
34
35
36 class TestWidget(QWidget):
37     def __init__(self):
38         QWidget.__init__(self)
39         vbox = QVBoxLayout(self)
40         self.view = QTableView()
41         self.model = Model(self.view)
42         self.view.setModel(self.model)
43         vbox.addWidget(self.view)
44
45
46 if __name__ == "__main__":
47     a = QApplication(sys.argv)
48     db=DBManager()
49     db.execSQL("delete from t1")
50     db.execSQL("create table if not exists t1 (f1 integer primary key,f2 varchar(20))")
51     db.execSQL(u"insert into t1 values(1,'我1')")
52     db.execSQL(u"insert into t1 values(2,'我test4')")
53     db.close()
54     w = TestWidget()
55     w.show()
56     sys.exit(a.exec_())
```

相关代码Demo

4.1 src package

4.1.1 Subpackages

src.dbdemo package

Subpackages

src.dbdemo.mysqlSimpleDemo package

Submodules

src.dbdemo.mysqlSimpleDemo.DBManager module

Module contents

Module contents

4.1.2 Module contents

索引和目录

- genindex
- modindex
- search

S

`src`, 19

`src.dbdemo`, 19

`src.dbdemo.mysqlSimpleDemo`, 19

S

src (模块), 19

src.dbdemo (模块), 19

src.dbdemo.mysqlSimpleDemo (模块), 19