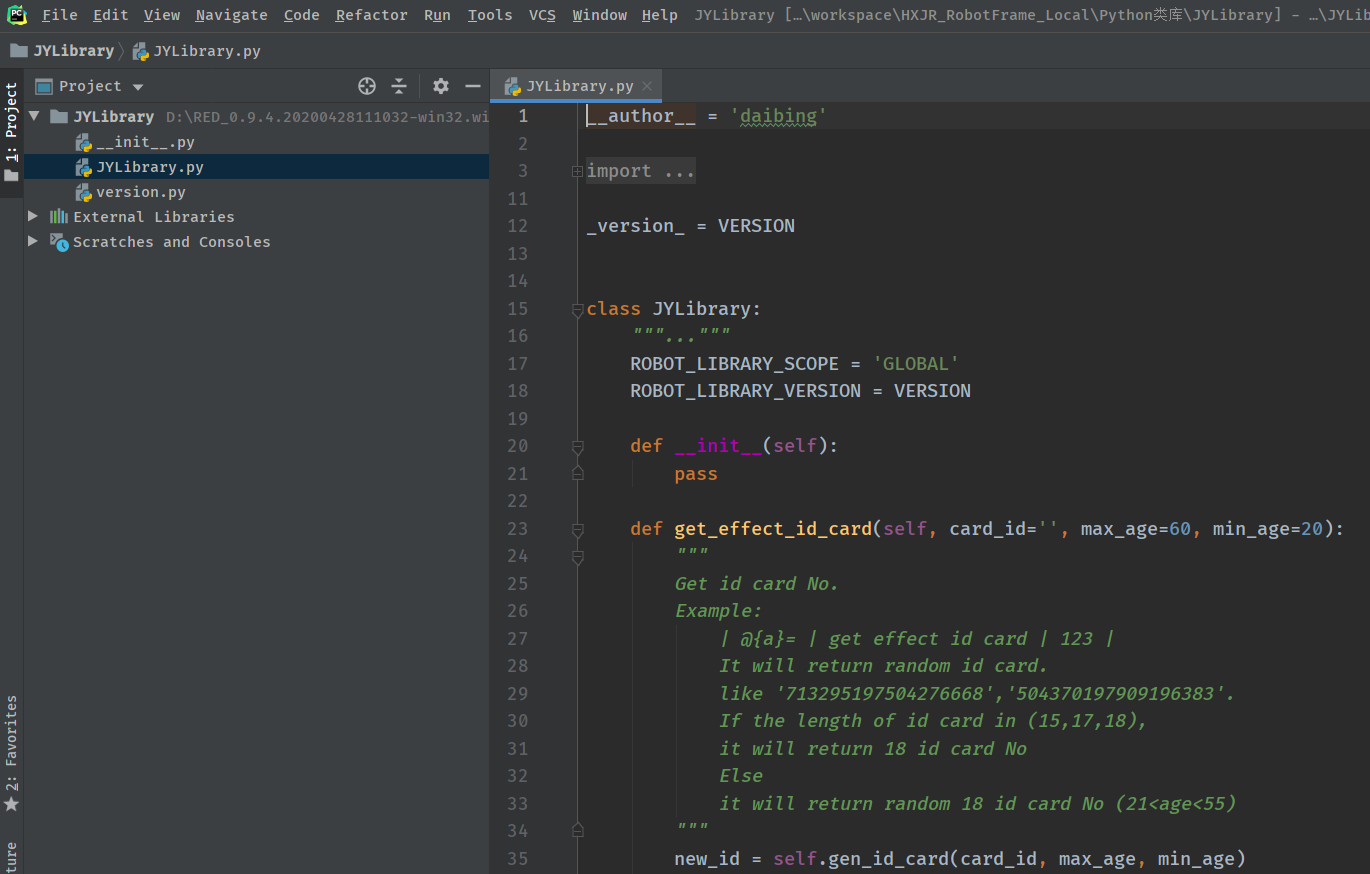
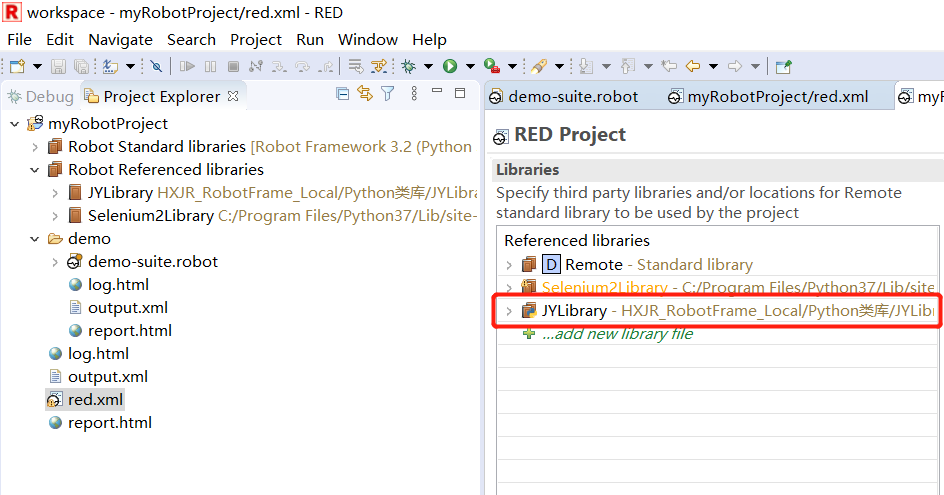
1. RED的扩展类引用

在本周的学习过程中，特意研究了一下RED框架对于第三方库的导入和工具支持。RED中新建项目后会出现一个文件名为red.xml的文件，总结加工了一下github文档说明，其实该文件类似于Maven项目中的pom.xml文件，这么做的好处是降低耦合和依赖。

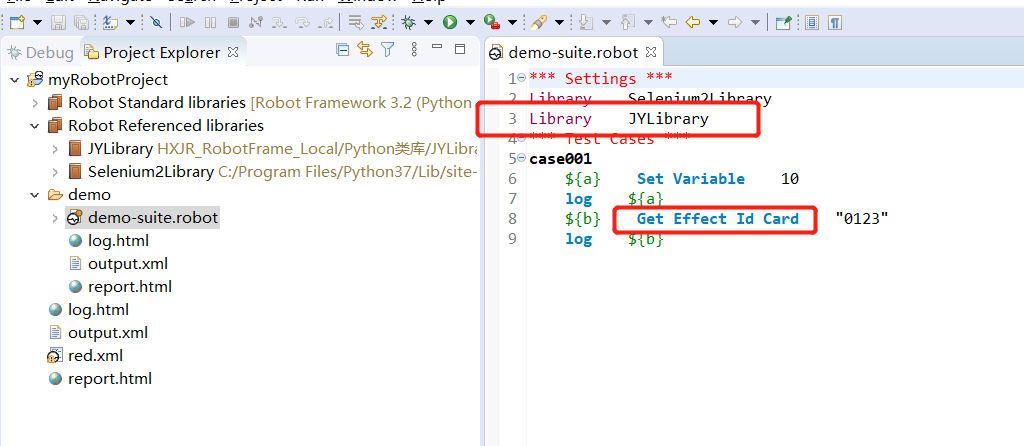
先看一下本地学习的例子，下图是本地编写的扩展类JYLibrary（其中包含方法get\_effect\_id\_card）：



接下来这里就和RIDE工具不同了，如果我们需要导入该扩展类，我们可以将其放置到任意文件夹下（文件夹路径尽量不要包含中文），之后在red.xml文件中将此py文件添加到library目录中，如图：

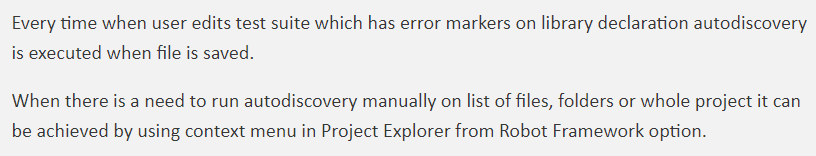


之后才可以在对应的工程目录下引入并使用：

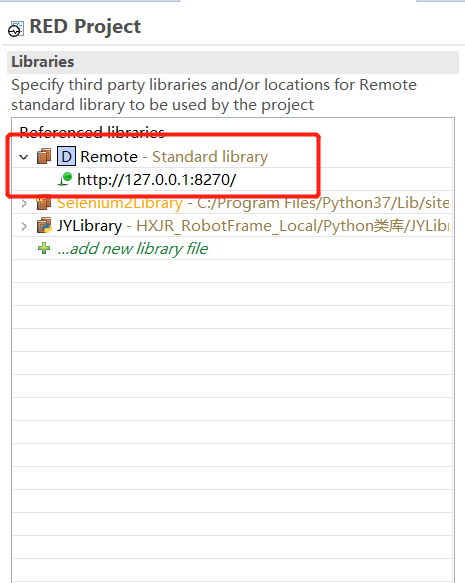


以上是第三方库在RED中的基本方法，通过学习我发现，除了人工将文件选择到red.xml中让工程识别扩展库之外，RED本身还支持其他几种加载方式：

1. RED有自我修复机制（CTRL+1），但是也可以通过人工方式识别，我们可以在对应工程目录上右键，之后选择 “Automatically discover and add libraries to red.xml”。这里若还有疑问，请参阅git官方文档：



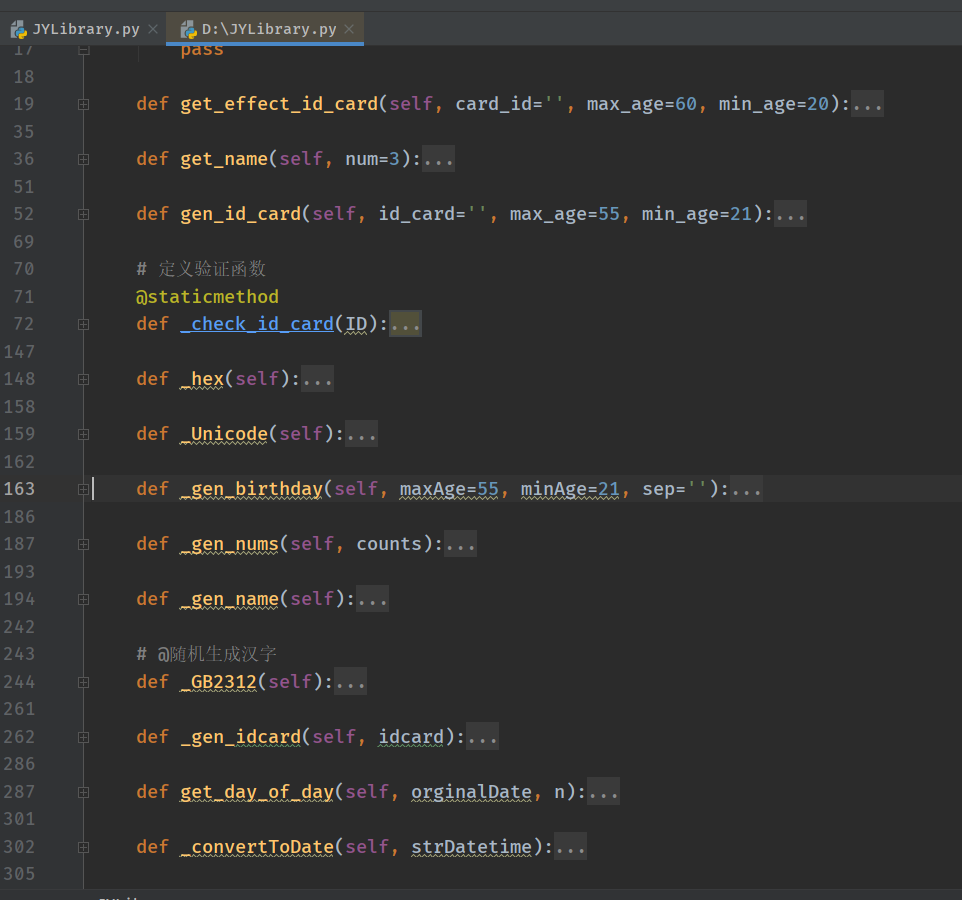
1. 第二种方式就是上面提到的自我修复机制，直接使用快捷键组合CTRL+1 进行；
2. 通过项目浏览器添加，我们可以在新建的project目录文件上右键选择 ”Robot Framework -> Add Library to red.xml ”；
3. 在实际工作中我们经常会遇到一个问题，远程执行节点如何导入？很幸运，RED已经提供给我们Remote标准动态库的导入功能了。



1. 简单说一下rf-builtin-libdoc.

上面提到了第三方扩展库，现在有一个假设，当执行机和开发机分离时，在执行机端可以通过RED的标准远程库使用扩展库，但是用户可能根本不清楚你这个扩展库是干嘛的，都包含什么对外API供使用，这个时候怎么办？每个用的人都讲一遍吗？显然不现实。

好在robotframework为我们考虑到这个问题了，我们完全可以使用rf内嵌libdoc库，进行api格式化文档输出，这里我依旧使用自己编写的JYLibraray进行演示：



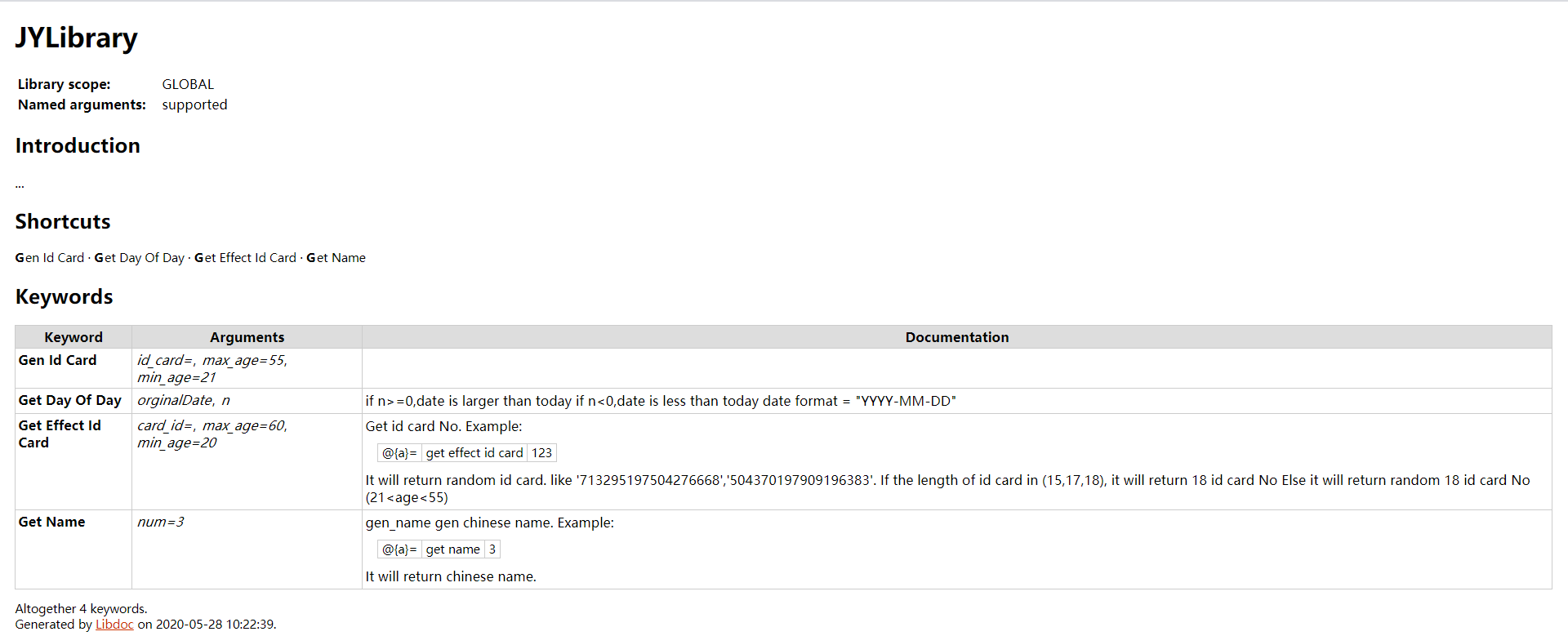
如上图，该扩展类包含如下成员方法，我们可以使用libdoc命令生成api文档，并格式化输出给使用者，执行命令：

python -m robot.libdoc d:/JYLibrary.py jy.html

之后到输出地址查看：



双击打开看一下效果：



当然这个只是该内置API的基本用法，它还支持其他argment的传入，请自行学习：

<https://robot-framework.readthedocs.io/en/latest/autodoc/robot.html#module-robot.libdoc>

1. python2和3的区别（从基础的说起）

很多同事朋友都问过我一个问题：出了python3是不是没必要再去学和使用python2了？我这里先引入一份jetbraind的调查问卷，该问卷统计截至到2019年底，python2的使用者保有量依旧高于python3。个人认为，这些都不是你在python2和3之间踌躇不决的客观条件，python3确实从堆栈、存储、效率等方面进行优化和提升。微软也曾说不在支持win7系统更新，可是依然有大批忠粉使用。我的意见是，两手都要抓都要硬，可以慢慢过渡。

今天先讲一下future模块，从名字就能看出来，未来，是的。Python的每个新版本都会增加一些新的功能，或者对原来的功能作一些改动。有些改动是不兼容旧版本的，也就是在当前版本运行正常的代码，到下一个版本运行就可能不正常了。要直接把代码升级到3.x是比较冒进的，因为有大量的改动需要测试。相反，可以在2.7版本中先在一部分代码中测试一些3.x的特性，如果没有问题，再移植到3.x不迟。

=================================================================

Python提供了\_\_future\_\_模块，把下一个新版本的特性导入到当前版本，于是我们就可以在当前版本中测试一些新版本的特性。说的通俗一点，就是你不用更新python的版本，直接加这个模块，就可以使用python新版本的功能。



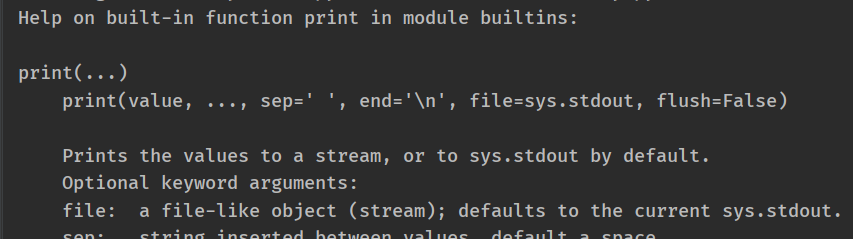
File "future\_demo.py", line 2

  help(print)

       ^

SyntaxError: invalid syntax

报错原因就是python2.x不支持print()语法，现在放开第一行的注释再运行一次：



我整理的一些常用\_\_future\_\_特性：

1. [除法特性-division](https://blog.csdn.net/jingbo18/article/details/79161501#6418-1516852451597)：地板除的差别，p2中整数相除==整数，余数会被扔掉，如果你要用精确除法，那请变成浮点吧；而p3中所有除法都是精确除；
2. [显示特性-print](https://blog.csdn.net/jingbo18/article/details/79161501#1517-1516852780828)：括号加不加的分别；
3. [字符编码-unicode](https://blog.csdn.net/jingbo18/article/details/79161501#3127-1517196390616)：p2中 ’study’ 表示字符串str，而u’study’则表示Unicode类型；而p3中字符串统一被视为Unicode，所以你如果还在单引号前加上u，其实是多此一举的，两者一致。

python2和python3的学习后面还会陆续更新加入。