

一、软件下载与安装

VScode下载地址: <https://code.visualstudio.com/>

<https://github.com/microsoft/vscode>

Python下载地址: <https://www.python.org/downloads/>

win版的VScode1.0和32位Python3.7, 安装Python时注意将Python添加到系统环境变量

二、VScode项目结构简介

VScode使用的是文件夹命名的项目, 也就是说你想写程序的话, 需要新建一个文件夹作为你的项目, 这个文件夹下放你的源文件, 如果需要运行, 还需要在这个文件夹下新建.vscode文件夹, 在.vscode文件夹下配置这个项目如何运行。

下面是一个典型的项目结构

```
1 | 项目名
2 | | 源文件1
3 | | 源文件2
4 | | .....
5 | | 源文件n
6 | |
7 | | .vscode
8 | |     tasks.json
9 | |     settings.json
```

这次配置坑比较多, VScode建议将地区改为en-US, 不然的话, 有些命令你必须打中文, 不能打英文, 打中文显示英文结果, 打英文没有结果。

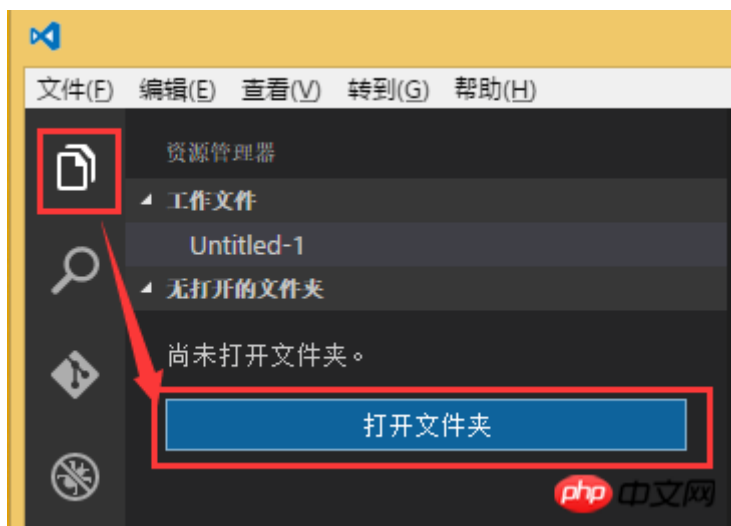
三、安装Python插件

安装Python插件能实现语法提示的一些功能, 建议还是安装一下。

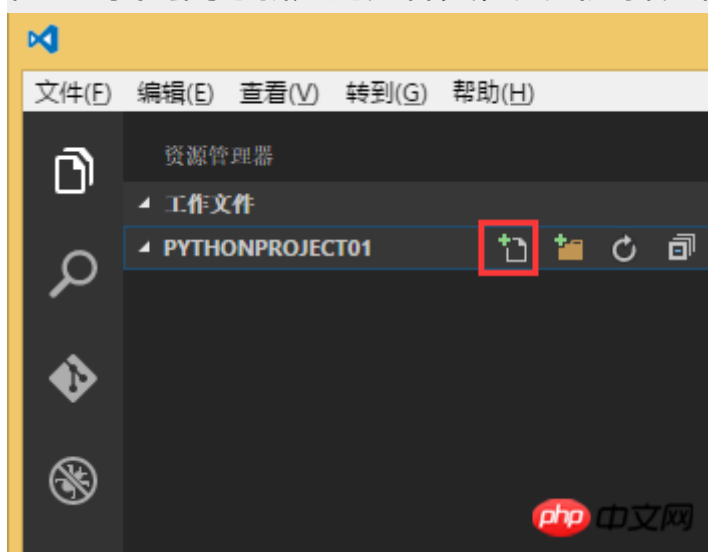
打开VScode, 查看-->命令面板(Ctrl+Shift+P), 输入ext install (中文输入: 扩展, 然后选择扩展: 安装扩展), 在出现的搜索结果中选择找到Python, 点右边的那一朵小云就可以安装了。

四、新建项目和编辑源代码

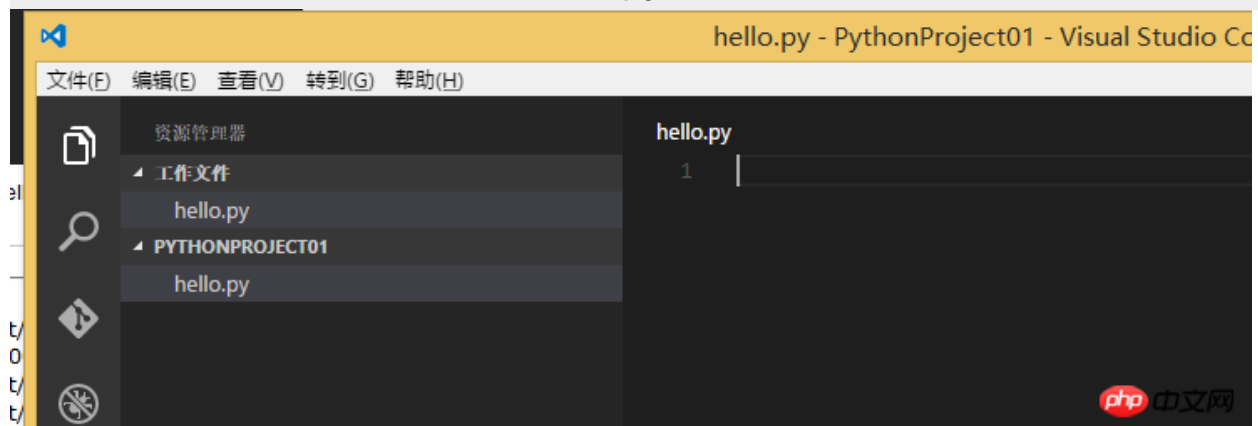
新建项目就是新建一个文件夹, 笔者先在D盘新建一个PythonProject01的文件夹(这一步在系统里面建, 不是VScode里), 点击VScode里的资源管理器按钮, 点击蓝色的打开文件夹按钮



在D盘找到刚才新建的文件夹，点击选择文件夹

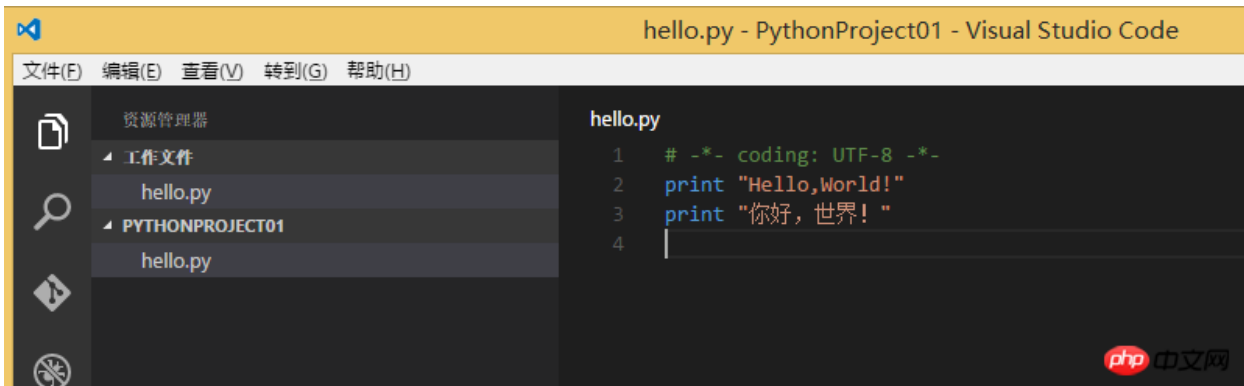


点击新建文件的按钮，文件名填hello.py



在右侧的编辑窗口输入以下代码，保存

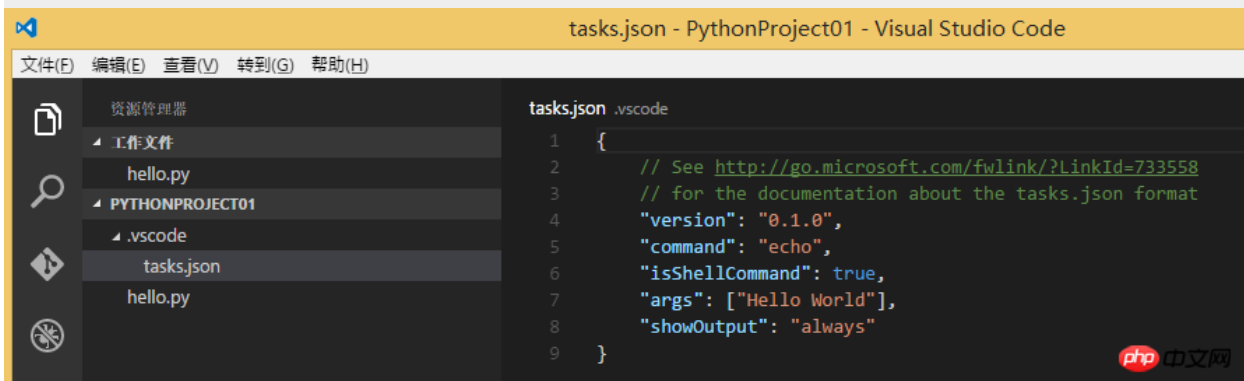
```
# -*- coding: UTF-8 -*-  
print "Hello,World!"  
print "你好，世界！"
```



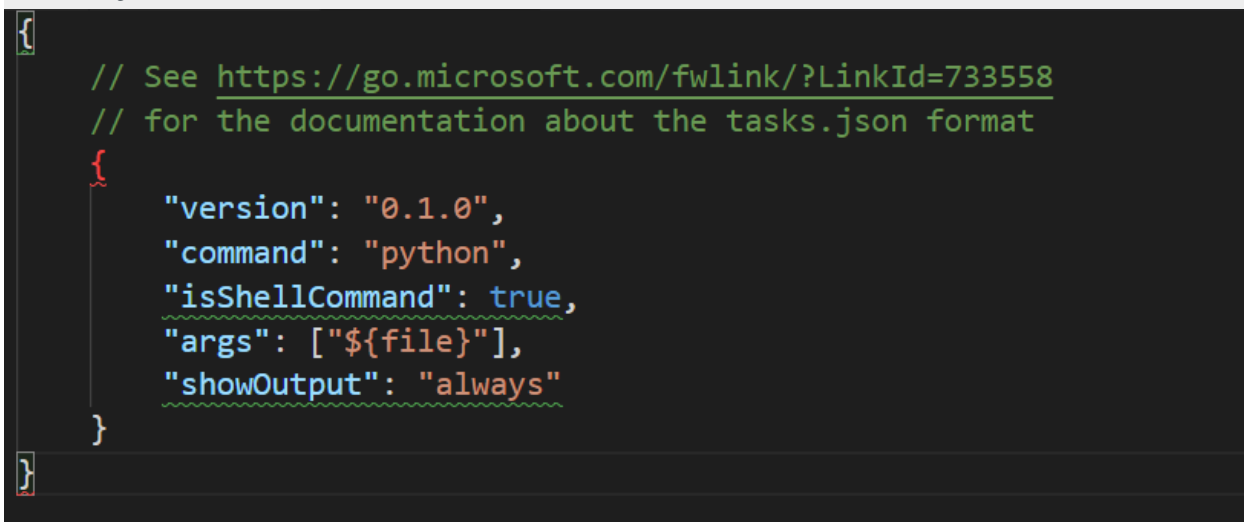
五、编辑task.json任务文件并运行该程序

查看--> 命令面板(Ctrl+Shift+P), 输入Tasks: Configure Task Runner (中文输入: 任务, 然后选择任务: 配置任务运行程序), 选择Other

此时VScode会自动生成.vscode文件夹并生成一个默认的task.json



将task.json内容改为如下内容并保存



或



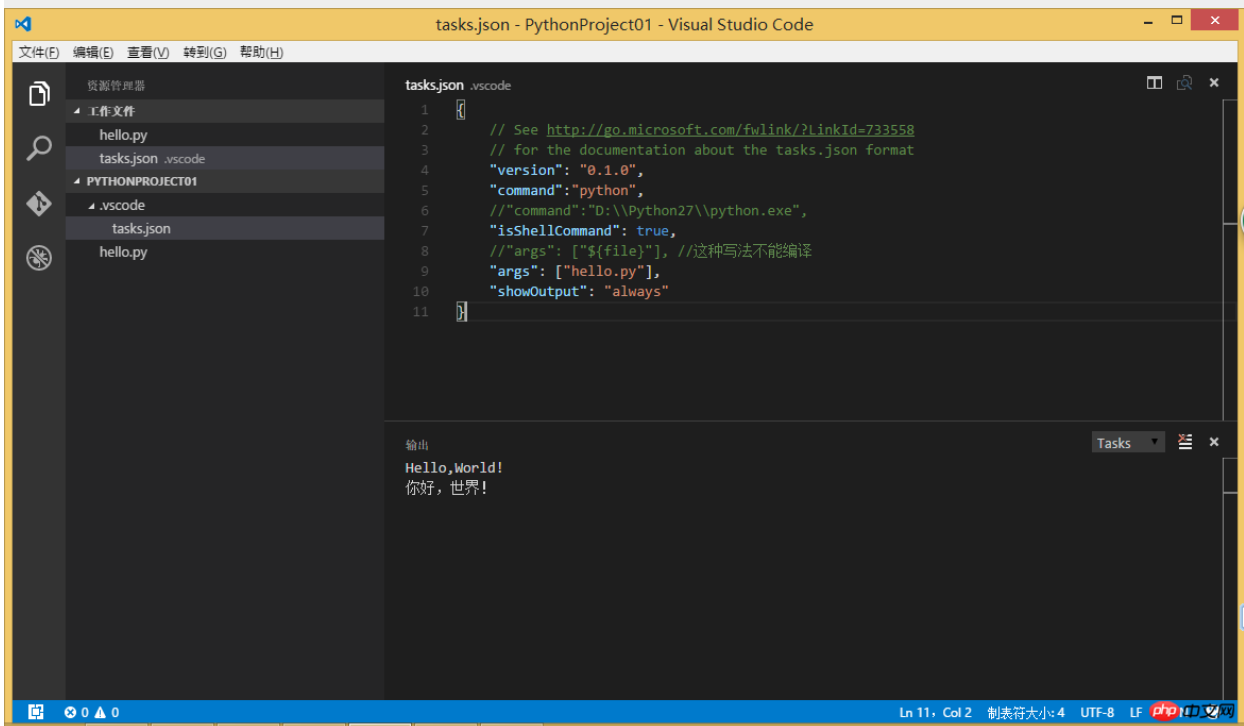
```
9  
10  
11  
    "args": ["hello.py"],  
    "showOutput": "always"  
}
```

运行方法如下：

查看--> 命令面板(**Ctrl+Shift+P**)，输入Tasks: Run Build Task（中文输入：任务，然后选择 任务：运行生成任务(Ctrl+Shift+B)）

或者按**F1**键直接输入

结果如下：



详细内容如下：

```
{  
    // See https://go.microsoft.com/fwlink/?LinkId=733558  
    // for the documentation about the tasks.json format  
    {  
        "version": "0.1.0",  
        "command": "python",  
        "isShellCommand": true,  
        "args": ["${file}"],  
        "showOutput": "always"  
    }  
}
```

}附：将语言更改为en-US

Ctrl+Shift+P, 输入语言(Language), 选择 配置语言(Configure Language), 会自动出现location.json文件

添加"locale":"en-US", 如下所示, 保存

```
1 {
2   // 定义 VSCode 的显示语言。
3   // 请参阅 http://go.microsoft.com/fwlink/?LinkId=761051, 了解支持的语言列
4   // 要更改值需要重启 VSCode。
5   "locale":"en-US"
6 }
```

重启VScode即可。

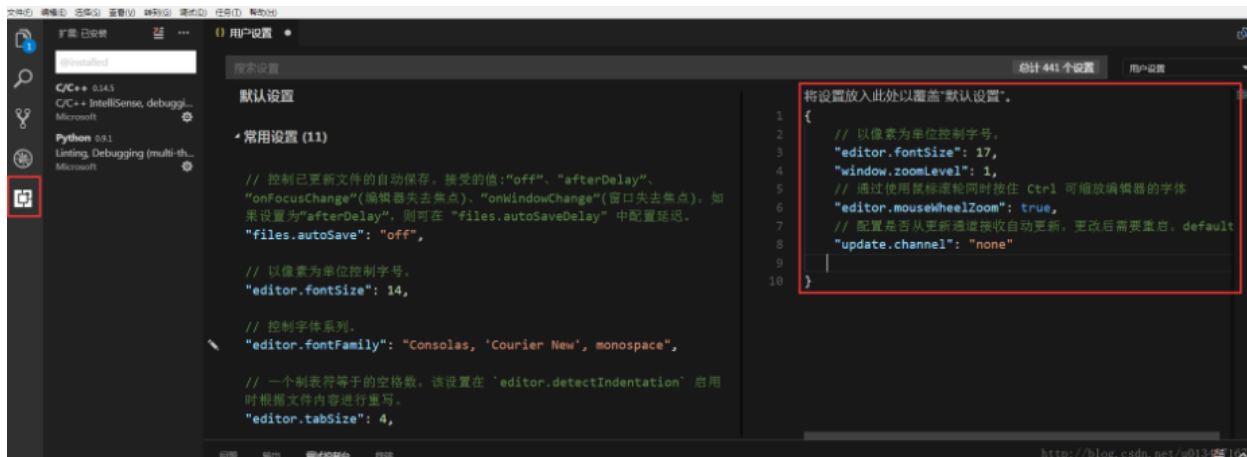
如果想改回中文, 就改为"locale":"zh-CN"或者删掉这个location.json文件。

六、Vscode下配置Python开发环境

基本配置vscode环境

Python扩展插件安装

如下图, 点击左侧边栏红色选中框, 在输入框输入Python扩展插件的安装, 安装完成后重新加载即可在已安装侧边栏中看到Python



vscode基本配置

vscode的基本配置是通过json文件管理的, 通过自定义配置覆盖默认配置。你可以在文件->首选项->设置, 打开一个setting.json

上图是我的基本配置, 右边红色选中框是我自己的用户自定义配置, 包括字体设定、是否滚动鼠标, 是否启动默认更新等。

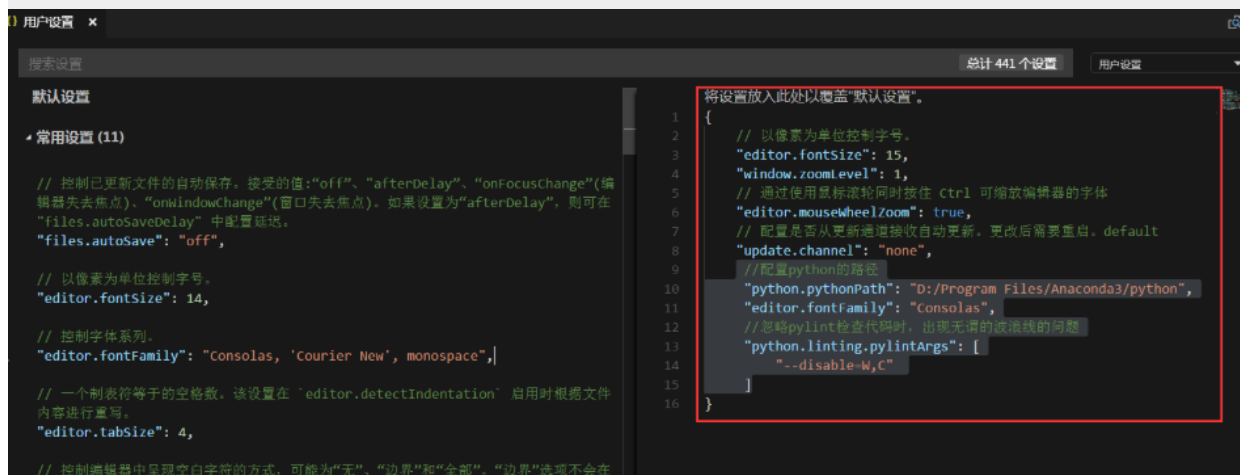
```
// 以像素为单位控制字号。
"editor.fontSize": 17,
"window.zoomLevel": 1,
// 通过使用鼠标滚轮同时按住 Ctrl 可缩放编辑器的字体
"editor.mouseWheelZoom": true,
// 配置是否从更新通道接收自动更新。更改后需要重启。default
"update.channel": "none"
```

添加Python编译配置

我们在上文右边添加关于Python编译器的配置命令（注意配置中最后一行不需要逗号），读者需要修改自己Python的安装路径（在上文中Anaconda的安装路径下）

```
//配置python的路径
"python.pythonPath": "D:/Program Files/Anaconda3/python",
"editor.fontFamily": "Consolas",
//忽略pylint检查代码时，出现无谓的波浪线的问题
"python.linting.pylintArgs": [
    "--disable=W,C"
]
```

最终效果为：



user setting内容如下：

```
{
  "workbench.colorTheme": "Visual Studio Dark",
  "sublimeTextKeymap.promptV3Features": true,
  "editor.multiCursorModifier": "ctrlCmd",
  "editor.snippetSuggestions": "top",
  "editor.formatOnPaste": true,
```

```

    "python.pythonPath":
"D:\\ProgramCJ\\Anaconda\\Anaconda3\\python",
    "window.zoomLevel": 1,
    "editor.fontSize": 18,
    "terminal.integrated.shell.windows":
"C:\\WINDOWS\\System32\\cmd.exe",
    "[python]": {

    },
    "python.jediEnabled": false
}

```

workspace setting内容如下:

```

{
    "python.autoComplete.addBrackets" :true
    "python.formatting.provider": "yapf"
    // 通过使用鼠标滚轮同时按住 Ctrl 可缩放编辑器的字体
    "editor.fontFamily": "monaco",
    "editor.mouseWheelZoom": true,
    "editor.fontFamily": "Consolas",
    // 忽略pylint检查代码时, 出现无谓的波浪线的问题
    "python.linting.pylintArgs": [
        "--disable=W,C"
    ]
    "python.terminal.activateEnvironment": false
    "python.autoComplete.extraPaths": [
        "D:\\ProgramCJ\\Anaconda\\Anaconda3\\python",
        "D:\\ProgramCJ\\Anaconda\\Anaconda3\\Lib",
        "D:\\ProgramCJ\\Anaconda\\Anaconda3\\Lib\\site-
packages",
        "D:\\ProgramCJ\\Anaconda\\Anaconda3\\DLLs"
    ]
}

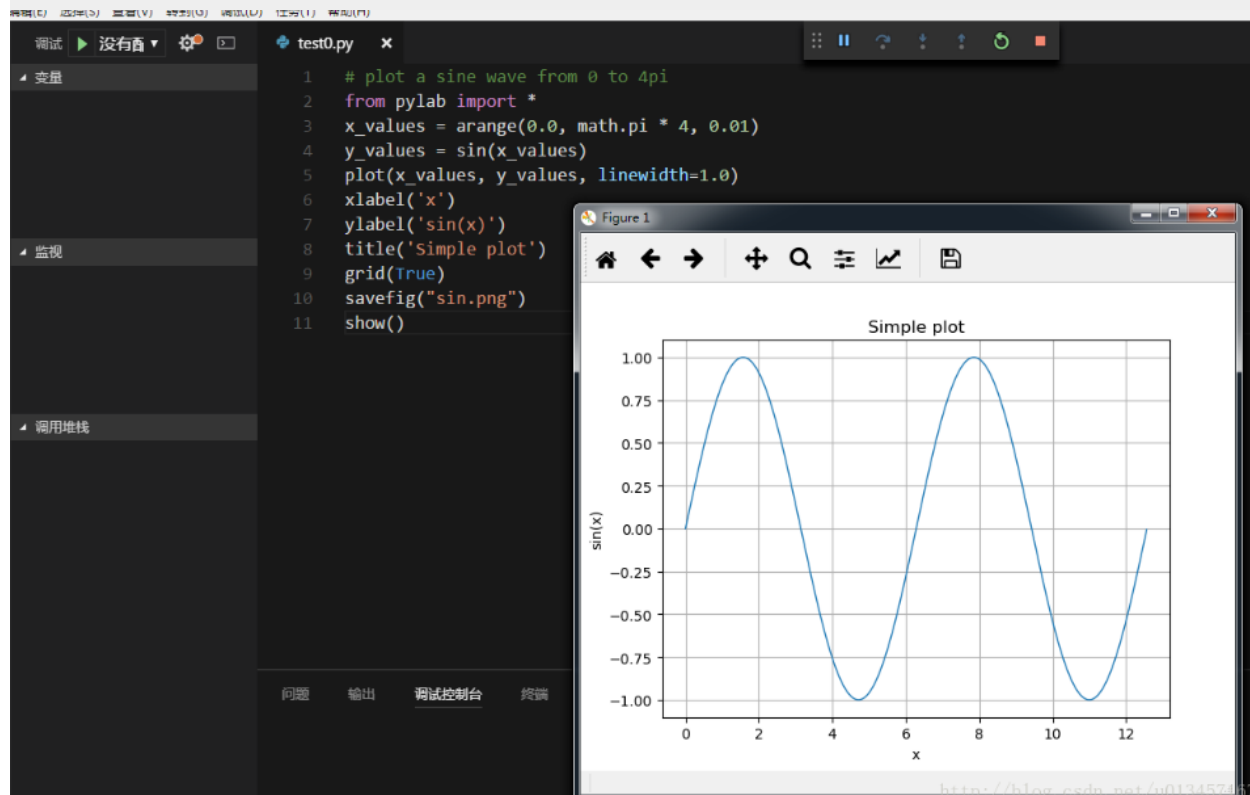
```

Python脚本编译测试

新建test.py, 代码如下:

```
# plot a sine wave from 0 to 4pi
from pylab import *
x_values = arange(0.0, math.pi * 4, 0.01)
y_values = sin(x_values)
plot(x_values, y_values, linewidth=1.0)
xlabel('x')
ylabel('sin(x)')
title('Simple plot')
grid(True)
savefig("sin.png")
show()
```

点击调试，或者快捷键ctrl+F5，编译：



我们可以看到，代码中调用了matplotlib数据包，并且代码编译通过，显示出所打印的正弦波图像。

这样安装的好处是我们不必在安装了Python后，一个个安装Python所需的各种工具包了。

完成