

Python快速入门

嵩天

Python开发工具及环境配置

高天

Python开发工具及环 境配置

Python快速入门
单元开篇

单元开篇



Python开发工具及环境配置

单元开篇

目的：了解Python语言开发工具

掌握1-2种Python语言开发工具

- 1 知道一批Python语言开发工具
- 2 掌握IDLE和PyCharm集成开发环境

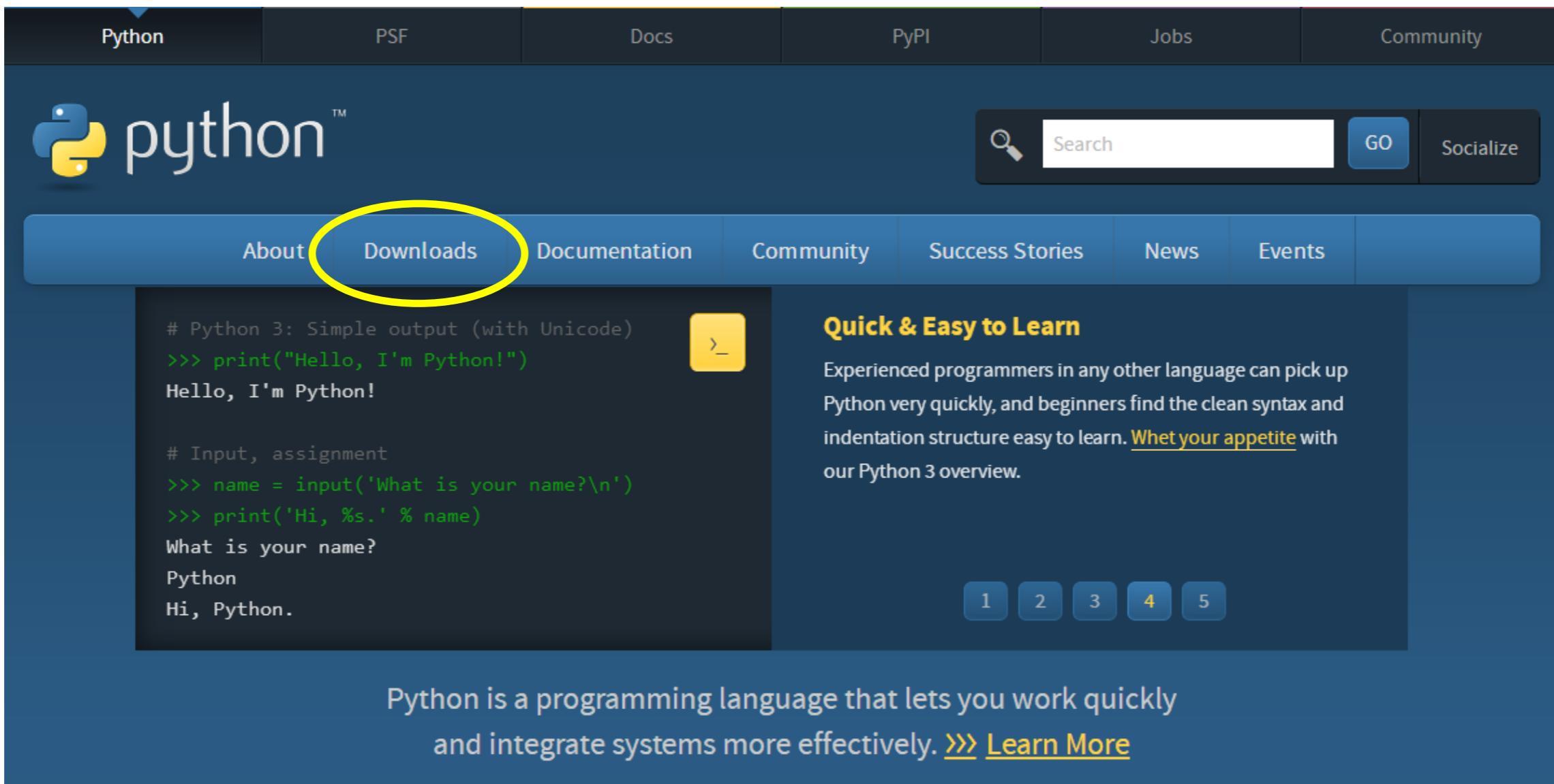
Python开发工具及环境配置

Python快速入门

IDLE开发工具

Python解释器安装

<http://www.python.org/>



The screenshot shows the Python.org homepage. The top navigation bar includes links for Python, PSF, Docs, PyPI, Jobs, and Community. The Python logo is on the left. A search bar with a magnifying glass icon and a 'GO' button is in the top right. Below the navigation is a blue header bar with links for About, Downloads (which is circled in yellow), Documentation, Community, Success Stories, News, and Events. The main content area features a dark box with Python code examples and a light blue box with text about Python's learnability. A navigation bar at the bottom of the content area shows pages 1 through 5.

Python 3: Simple output (with Unicode)
>>> print("Hello, I'm Python!")
Hello, I'm Python!

Input, assignment
>>> name = input('What is your name?\n')
>>> print('Hi, %s.' % name)
What is your name?
Python
Hi, Python.

Quick & Easy to Learn
Experienced programmers in any other language can pick up Python very quickly, and beginners find the clean syntax and indentation structure easy to learn. [Whet your appetite](#) with our Python 3 overview.

1 2 3 4 5

Python is a programming language that lets you work quickly and integrate systems more effectively. [» Learn More](#)

Python解释器安装

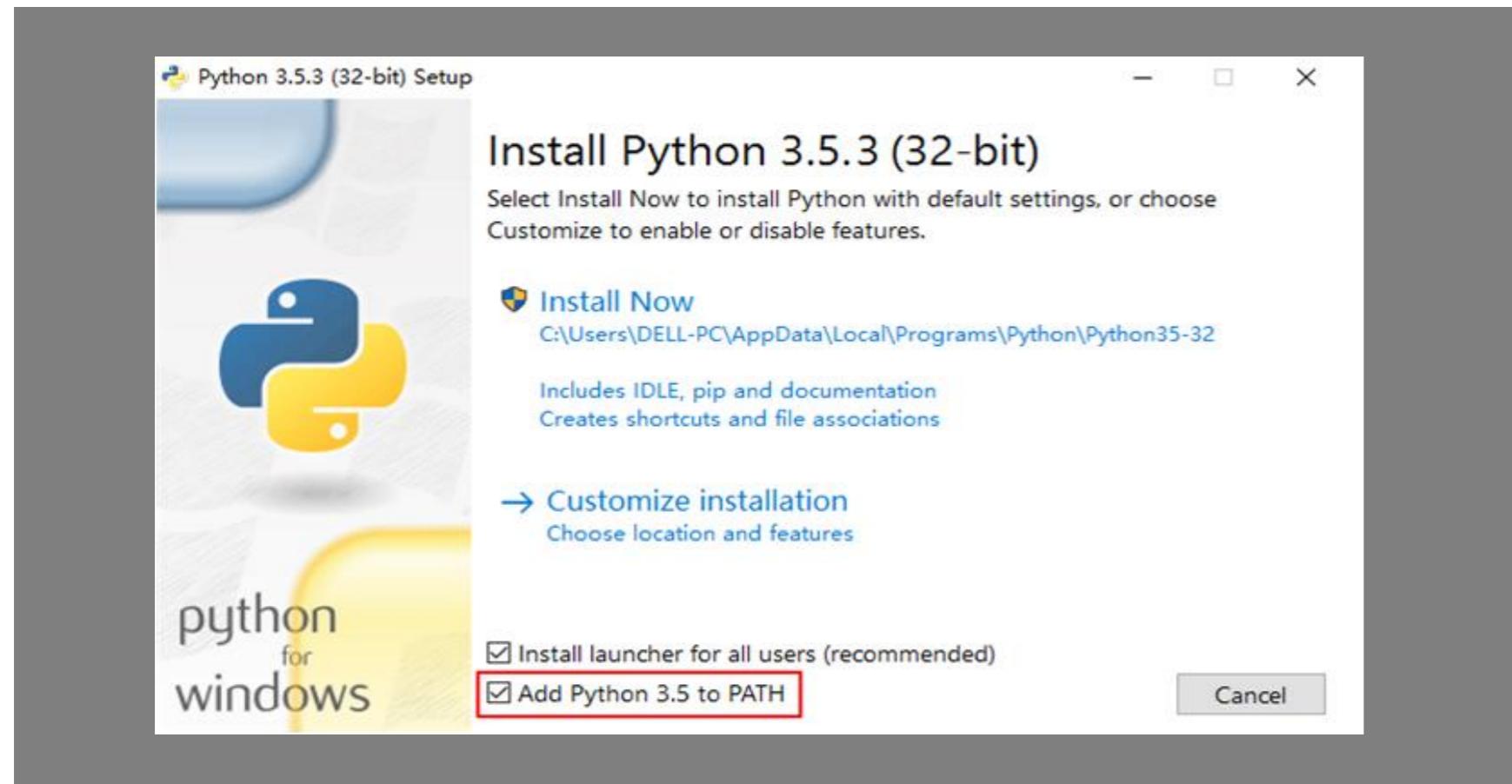
http://www.python.org/

- [Python 3.6.5 - 2018-03-28](#)
 - [Download Windows x86 web-based installer](#)
 - **[Download Windows x86 executable installer](#)**
 - [Download Windows x86 embeddable zip file](#)
 - [Download Windows x86-64 web-based installer](#)
 - **[Download Windows x86-64 executable installer](#)**
 - [Download Windows x86-64 embeddable zip file](#)
 - [Download Windows help file](#)
- x86 32位Python解释器
- X86-64 64位Python解释器

务必安装3.5.3以上版本，不要安装2.x系列

Python解释器安装

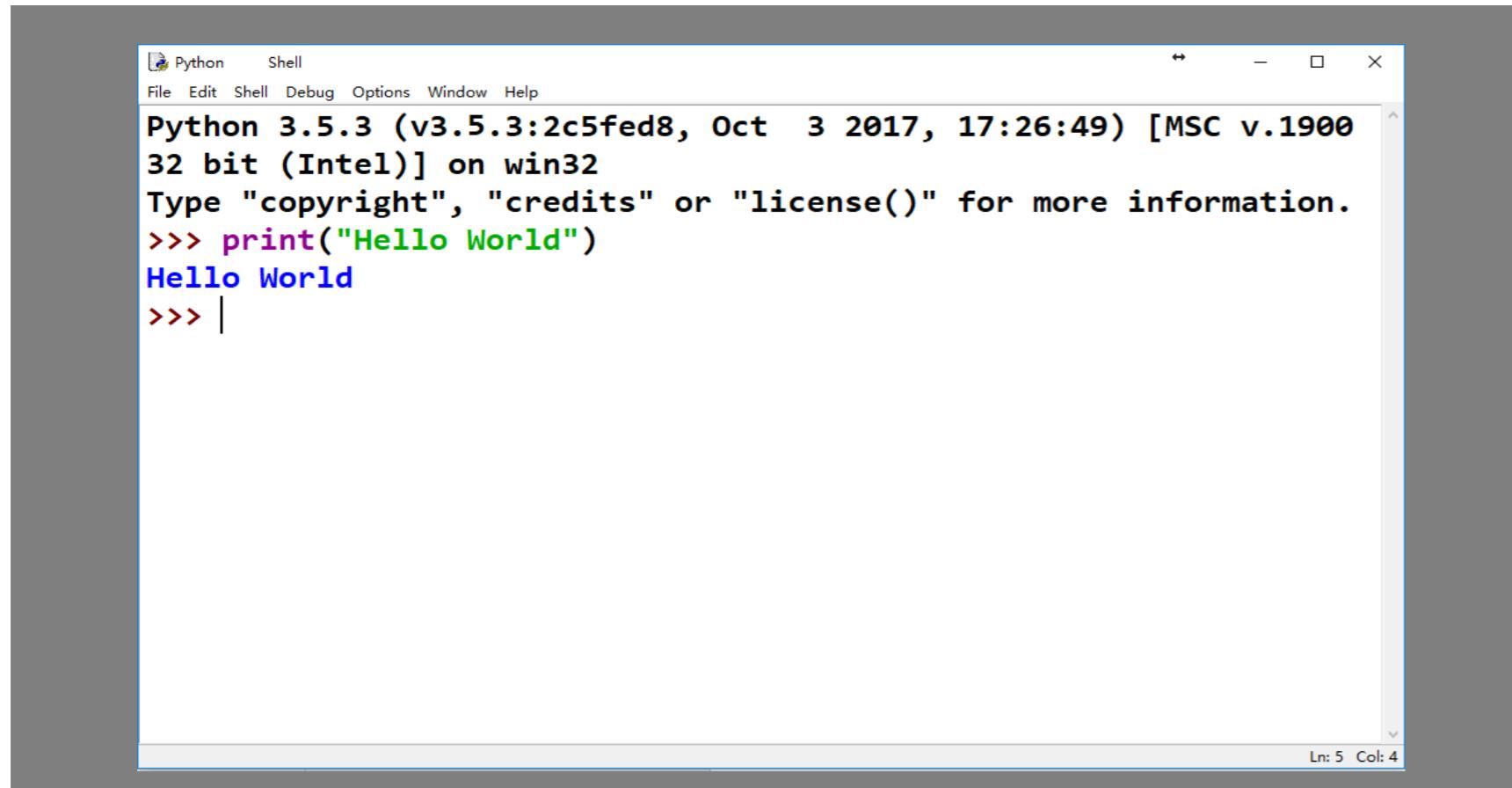
安装过程



- 选中添加路径选项

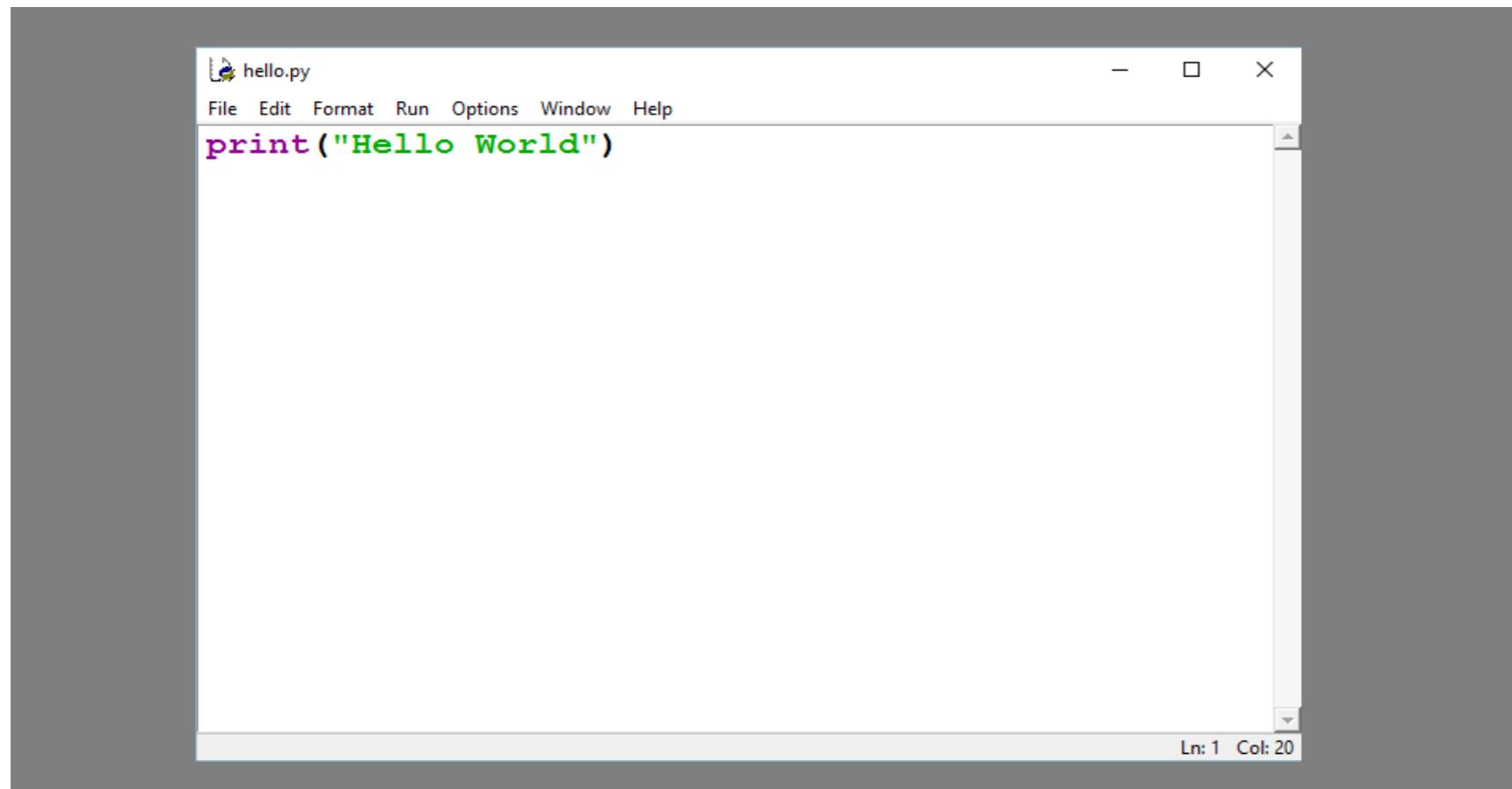
务必安装3.5.3以上版本，不要安装2.x系列

交互式环境



- >>> 是交互式的提示符
- 每输入一行代码，运行相应结果

文件式环境



- **CTRL+N启动一个文本编辑器**
- **IDLE提供的Python文件式环境**

快捷键

- **CTRL + N**: 在IDLE交互界面下，用来启动IDLE编辑器
- **CTRL + Q**: 退出IDLE或IDLE编辑器
- **ALT + 3**: 在IDLE编辑器内，注释选定区域文本
- **ALT + 4**: 在IDLE编辑器内，解除注释选定区域文本
- **ALT + Q**: 在IDLE编辑器内，将Python代码进行格式化布局
- **F5**: 在IDLE编辑器内，执行Python程序

代码尝试

写段代码试一试

```
#SimpleTempConvert.py  
TempStr = input("请输入摄氏温度值: ")  
F = 1.8*eval(TempStr) + 32  
print("对应的华氏温度是{:.2f}F".format(F))
```

Python快速入门

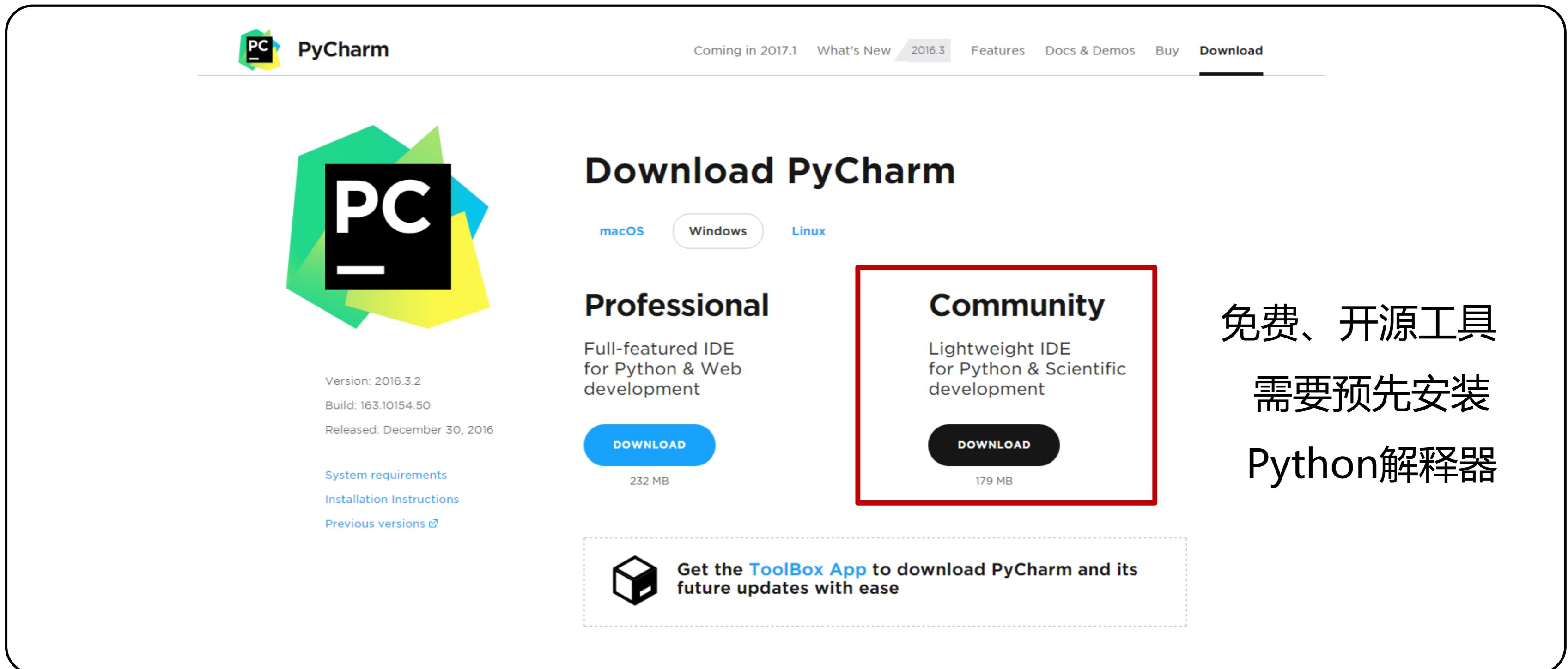
PyCharm开 发工具

Pycharm工具安装

<https://www.jetbrains.com/pycharm>



Pycharm工具安装



The screenshot shows the PyCharm download page. At the top, there is a navigation bar with links: Coming in 2017.1, What's New (highlighted in grey), 2016.3, Features, Docs & Demos, Buy, and Download (underlined).

On the left, there is a large PyCharm logo with the letters 'PC' inside a hexagonal shape, and text indicating the version: Version: 2016.3.2, Build: 163.10154.50, and Released: December 30, 2016. Below this are links for System requirements, Installation Instructions, and Previous versions.

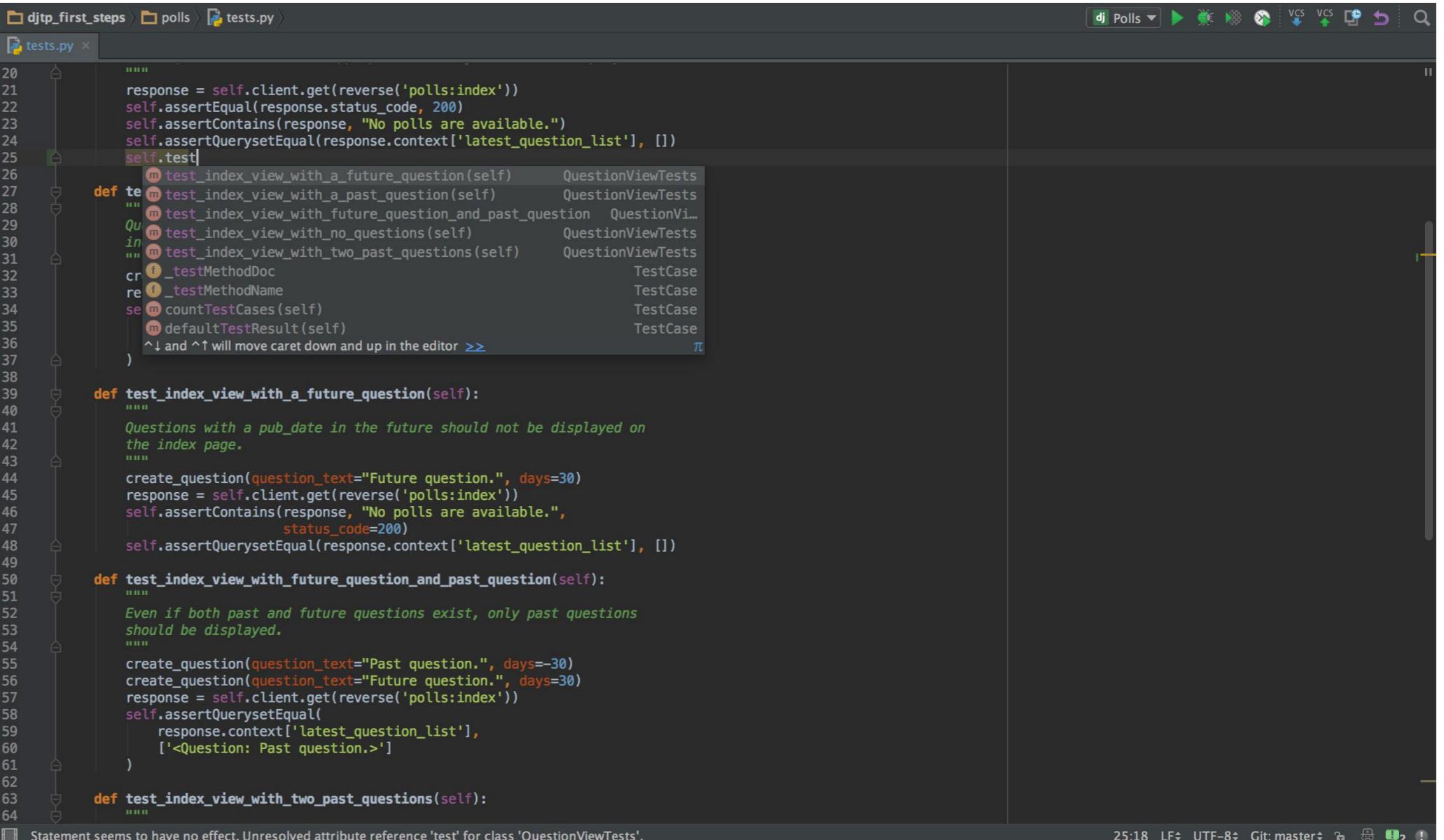
The main content area is titled 'Download PyCharm' and shows two download options: 'macOS', 'Windows' (selected), and 'Linux'. Each option has a 'DOWNLOAD' button and a file size: 232 MB for macOS and 179 MB for Windows.

The 'Community' edition is highlighted with a red border. Below the download buttons, there is a note: 'Get the [ToolBox App](#) to download PyCharm and its future updates with ease'.

On the right side of the page, there is a vertical column of text: '免费、开源工具', '需要预先安装', and 'Python解释器'.

Pycharm开发工具

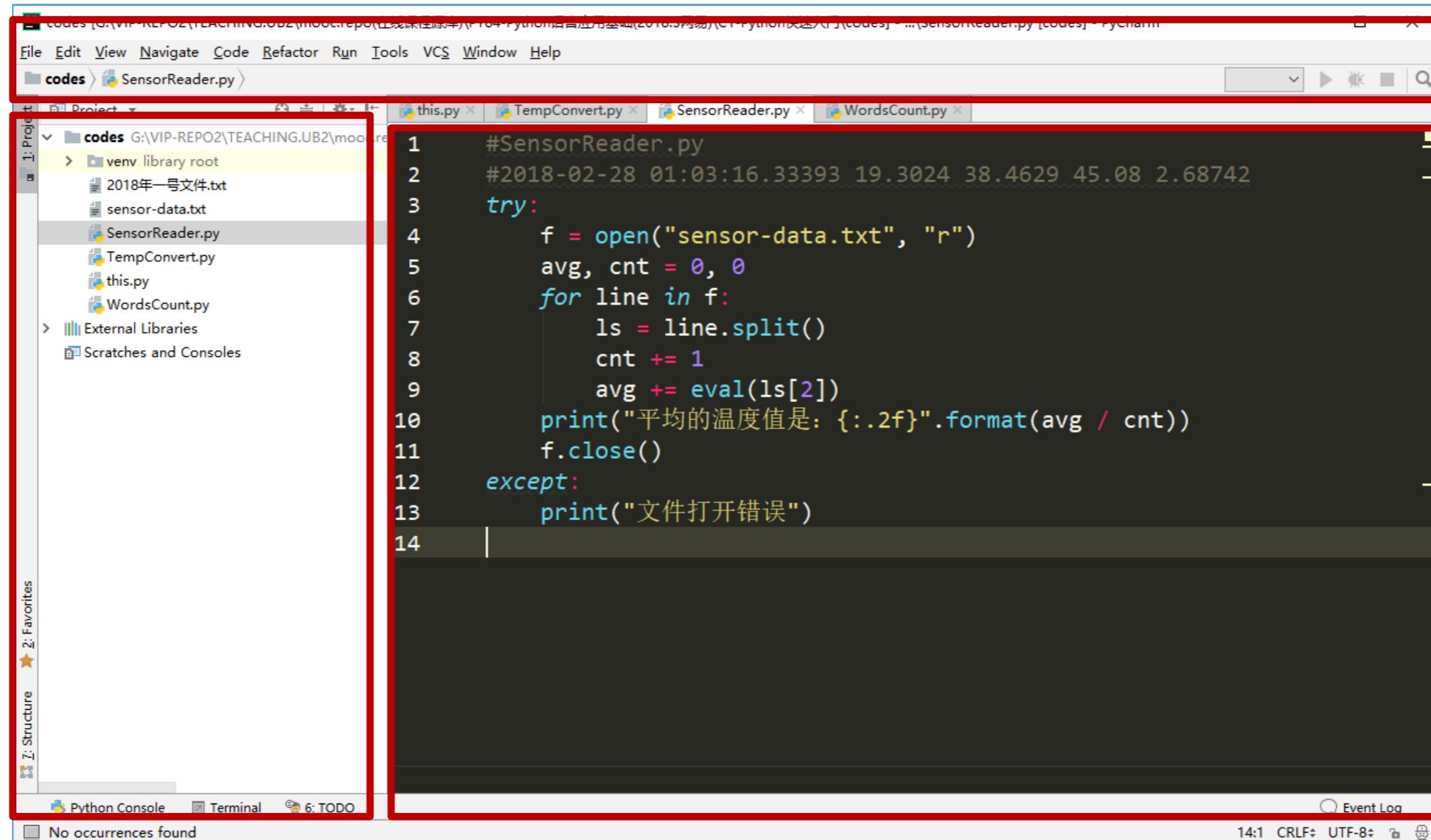
- 社区版免费
- 调试功能丰富
- 通用类Python开发
- 适合较复杂工程
- Win/Linux/Mac OS



```
20     response = self.client.get(reverse('polls:index'))
21     self.assertEqual(response.status_code, 200)
22     self.assertContains(response, "No polls are available.")
23     self.assertQuerysetEqual(response.context['latest_question_list'], [])
24     self.test
25
26     def test_index_view_with_a_future_question(self):
27         """Questions with a pub_date in the future should not be displayed on
28         the index page.
29         """
30         question = create_question(
31             question_text="Future question.", days=30)
32         response = self.client.get(reverse('polls:index'))
33         self.assertContains(response, "No polls are available.",
34                             status_code=200)
35         self.assertQuerysetEqual(response.context['latest_question_list'], [])
36
37     def test_index_view_with_a_past_question(self):
38         """
39         Even if both past and future questions exist, only past questions
40         should be displayed.
41         """
42         question = create_question(
43             question_text="Past question.", days=-30)
44         question = create_question(
45             question_text="Future question.", days=30)
46         response = self.client.get(reverse('polls:index'))
47         self.assertQuerysetEqual(
48             response.context['latest_question_list'],
49             ['<Question: Past question.>'])
50
51     def test_index_view_with_future_question_and_past_question(self):
52         """
53         Even if both past and future questions exist, only past questions
54         should be displayed.
55         """
56         question = create_question(
57             question_text="Past question.", days=-30)
58         question = create_question(
59             question_text="Future question.", days=30)
60         response = self.client.get(reverse('polls:index'))
61         self.assertQuerysetEqual(
62             response.context['latest_question_list'],
63             ['<Question: Past question.>'])
64
65     def test_index_view_with_no_questions(self):
66         """
67         An index page with no polls should display a message saying "No polls
68         are available."
69         """
70         response = self.client.get(reverse('polls:index'))
71         self.assertContains(response, "No polls are available.")
72
73     def test_index_view_with_two_past_questions(self):
74         """
75         An index page with two past polls should display both.
76         """
77         question1 = create_question(
78             question_text="Past question 1.", days=-30)
79         question2 = create_question(
80             question_text="Past question 2.", days=-30)
81         response = self.client.get(reverse('polls:index'))
82         self.assertQuerysetEqual(
83             response.context['latest_question_list'],
84             ['<Question: Past question 1.>', '<Question: Past question 2.>'])
```

Pycharm开发工具

工程导航



工具菜单

代码编写

Pycharm开发工具

哪些人适合用IDE（集成开发环境）？

- 专业程序员
- 致力于成为专业程序员的学习者
- 编程代码量一般超过100行
- 编程初学者、入门学习者，请用IDLE

Python快速入门

其他开发工具 概述

其他开发工具概述

文本工具类

- ❑ Notepad++
- ❑ Vim & Emacs
- ❑ Sublime Text
- ❑ Atom
- ❑ Komodo Edit

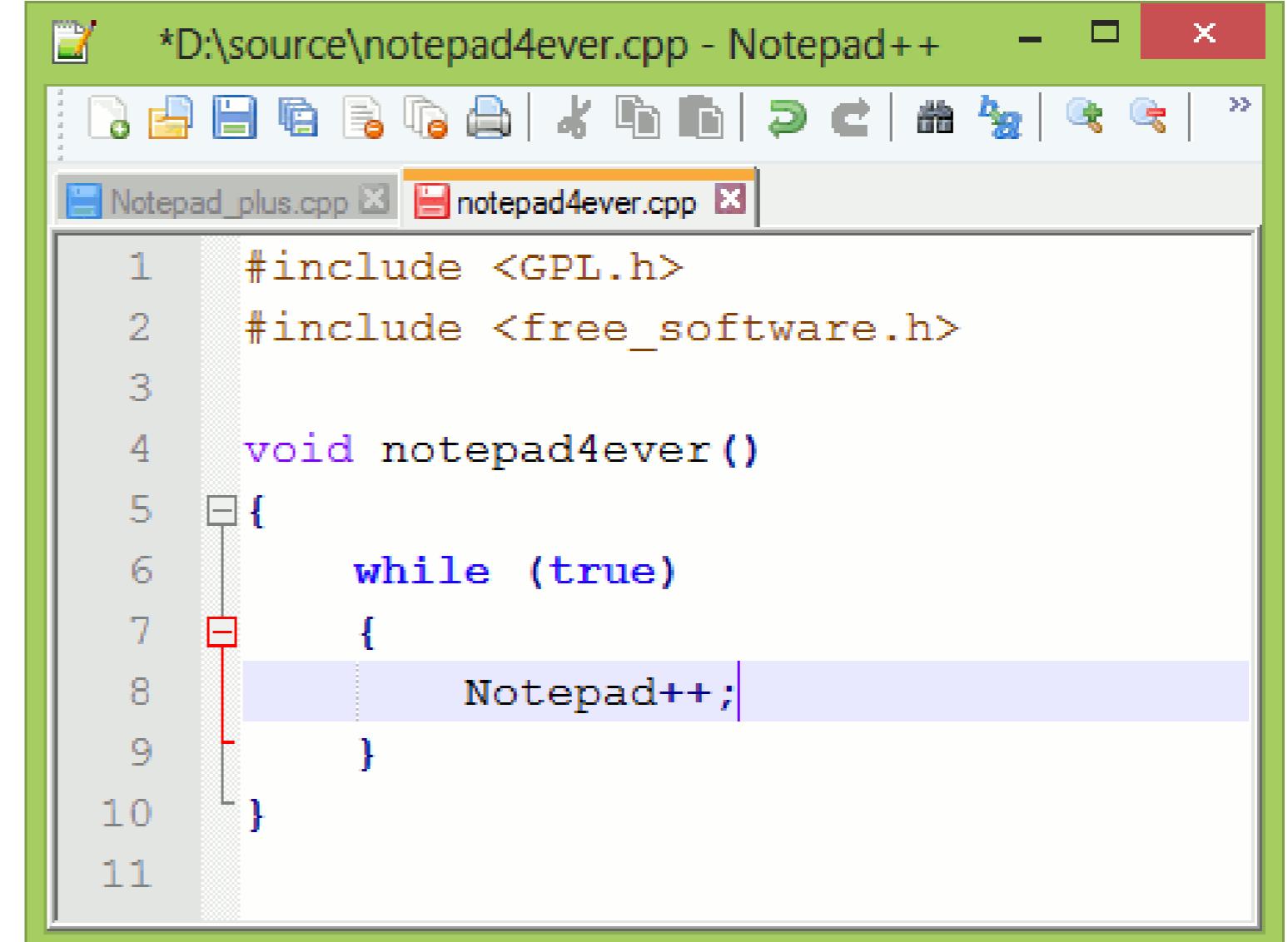
集成工具类

- ❑ Wing
- ❑ PyDev & Eclipse
- ❑ Visual Studio
- ❑ Visual Studio Code
- ❑ Anaconda & Spyder

Notepad++

<https://notepad-plus-plus.org>

- 轻巧的文本编辑器
- 免费使用
- 适合各类编程语言
- 适合专业程序员
- Win为主



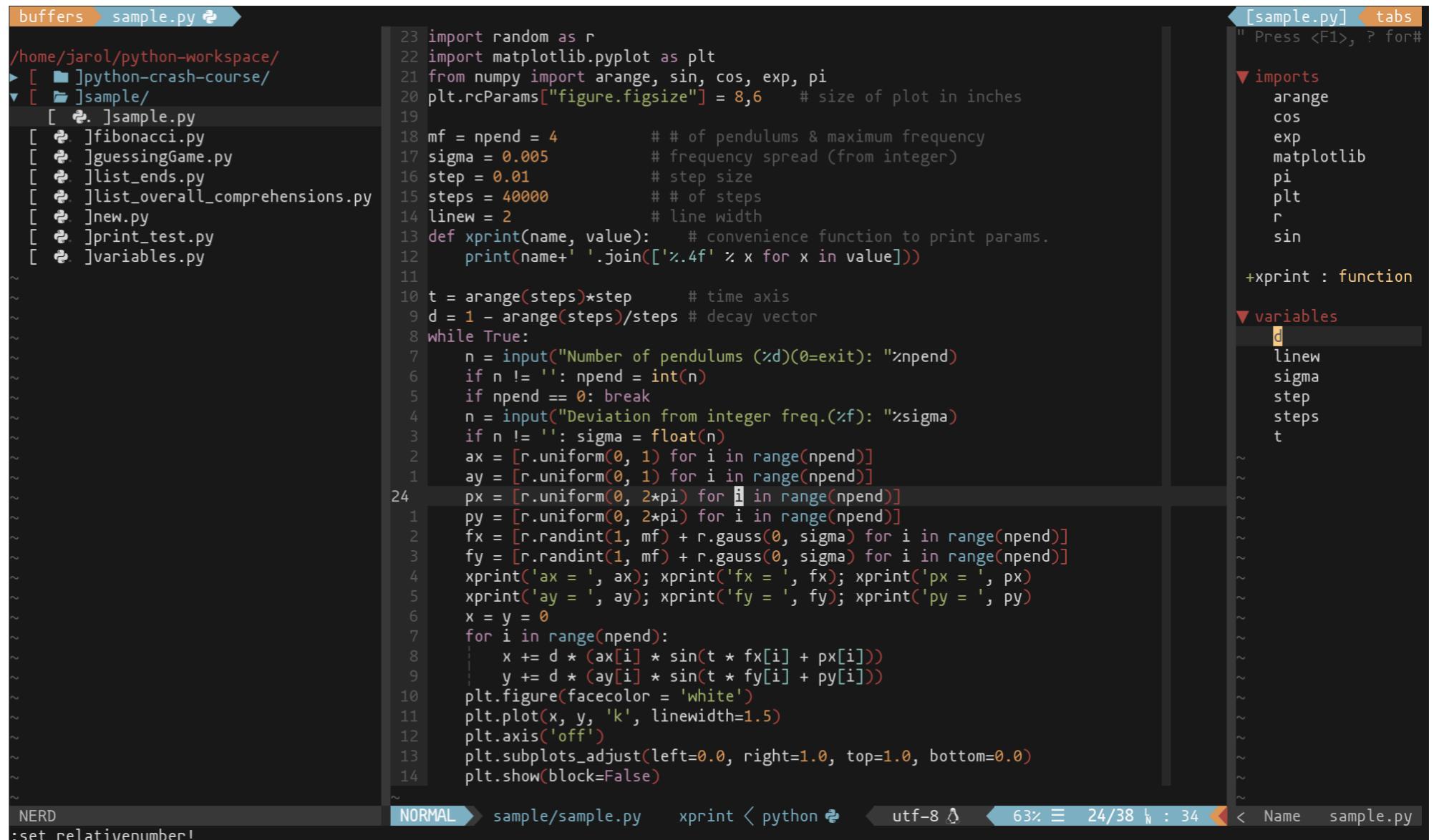
The screenshot shows the Notepad++ interface with a C++ code editor. The title bar reads "*D:\source\notepad4ever.cpp - Notepad++". The toolbar has various icons for file operations. The status bar shows "Notepad_plus.cpp" and "notepad4ever.cpp". The code editor displays the following C++ code:

```
1 #include <GPL.h>
2 #include <free_software.h>
3
4 void notepad4ever()
5 {
6     while (true)
7     {
8         Notepad++;
9     }
10 }
```

The code is syntax-highlighted, with keywords in blue and identifiers in purple. The line "Notepad++;" is highlighted in light blue, indicating an auto-completed suggestion. The Notepad++ logo is visible in the status bar.

Vim & Emac

- 老牌经典文本编辑器
- 免费使用
- 专业编程体验
- 适合专业程序员
- Linux为主



```
buffers > sample.py <▶
/home/jarol/python-workspace/
  ▶ [ ]python-crash-course/
  ▶ [ ]sample/
    [ ]sample.py
      [ ]fibonacci.py
      [ ]guessingGame.py
      [ ]list_ends.py
      [ ]list_overall_comprehensions.py
      [ ]new.py
      [ ]print_test.py
      [ ]variables.py

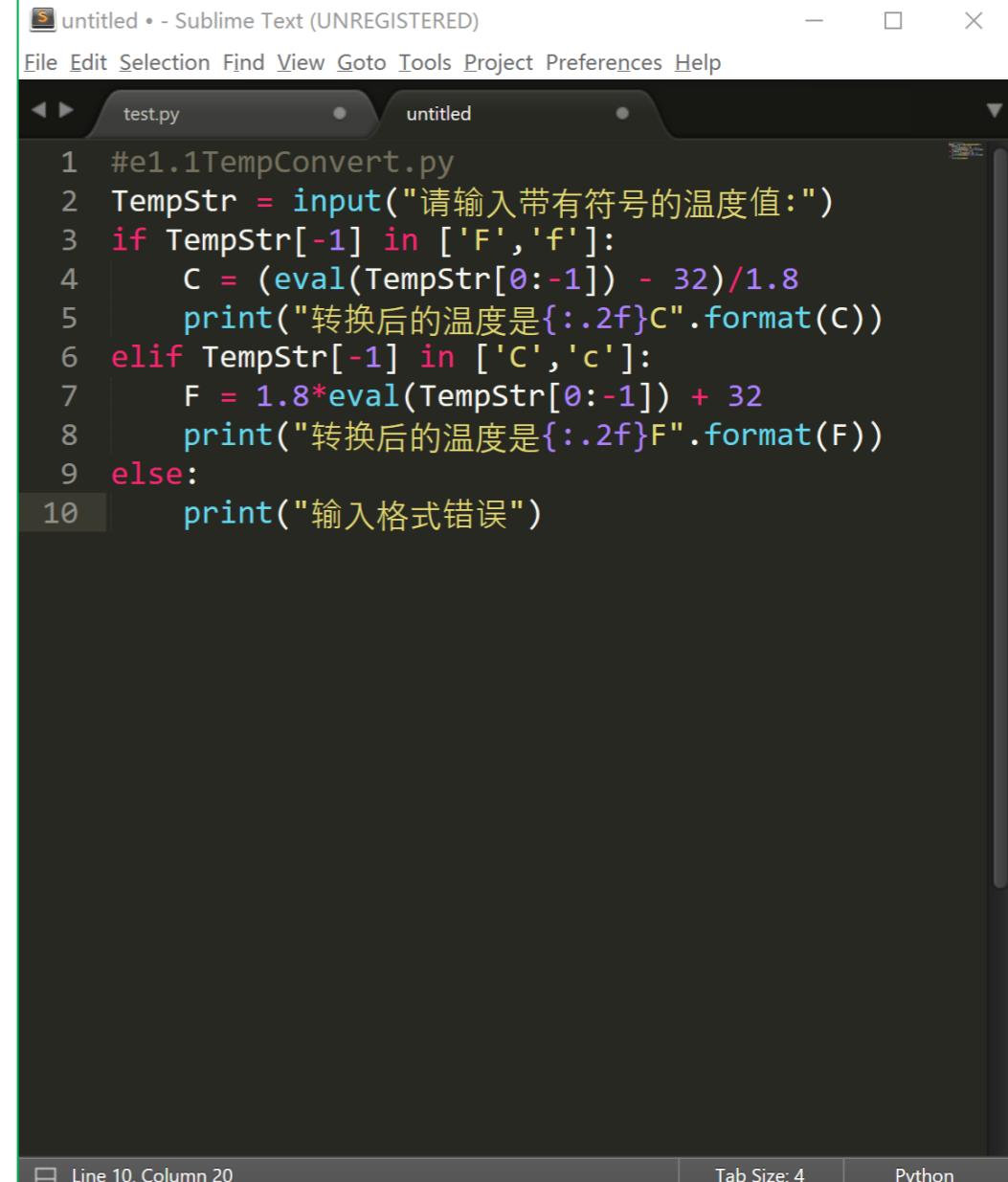
23 import random as r
22 import matplotlib.pyplot as plt
21 from numpy import arange, sin, cos, exp, pi
20 plt.rcParams["figure.figsize"] = 8,6    # size of plot in inches
19
18 mf = npend = 4          # # of pendulums & maximum frequency
17 sigma = 0.005           # frequency spread (from integer)
16 step = 0.01             # step size
15 steps = 40000           # # of steps
14 linewidth = 2           # line width
13 def xprint(name, value): # convenience function to print params.
12     print(name+' '.join(['%.4f' % x for x in value]))
11
10 t = arange(steps)*step    # time axis
9 d = 1 - arange(steps)/steps # decay vector
8 while True:
7     n = input("Number of pendulums (2d)(0=exit): "%pPEND)
6     if n != '': npend = int(n)
5     if npend == 0: break
4     n = input("Deviation from integer freq.(%f): "%SIGMA)
3     if n != '': sigma = float(n)
2     ax = [r.uniform(0, 1) for i in range(npend)]
1     ay = [r.uniform(0, 1) for i in range(npend)]
24     px = [r.uniform(0, 2*pi) for i in range(npend)]
1     py = [r.uniform(0, 2*pi) for i in range(npend)]
2     fx = [r.randint(1, mf) + r.gauss(0, sigma) for i in range(npend)]
3     fy = [r.randint(1, mf) + r.gauss(0, sigma) for i in range(npend)]
4     xprint('ax = ', ax); xprint('fx = ', fx); xprint('px = ', px)
5     xprint('ay = ', ay); xprint('fy = ', fy); xprint('py = ', py)
6     x = y = 0
7     for i in range(npend):
8         x += d * (ax[i] * sin(t * fx[i] + px[i]))
9         y += d * (ay[i] * sin(t * fy[i] + py[i]))
10    plt.figure(facecolor = 'white')
11    plt.plot(x, y, 'k', linewidth=1.5)
12    plt.axis('off')
13    plt.subplots_adjust(left=0.0, right=1.0, top=1.0, bottom=0.0)
14    plt.show(block=False)

NERD NORMAL > sample/sample.py xprint < python < utf-8 < 63% 24/38 34 < Name sample.py
:set relativenumber!
```

Sublime Text

<https://www.sublimetext.com>

- 轻巧的文本编辑器
- 免费使用
- 专业编程体验
- 单人独立开发
- 适合专业程序员
- Win/Linux/Mac OS

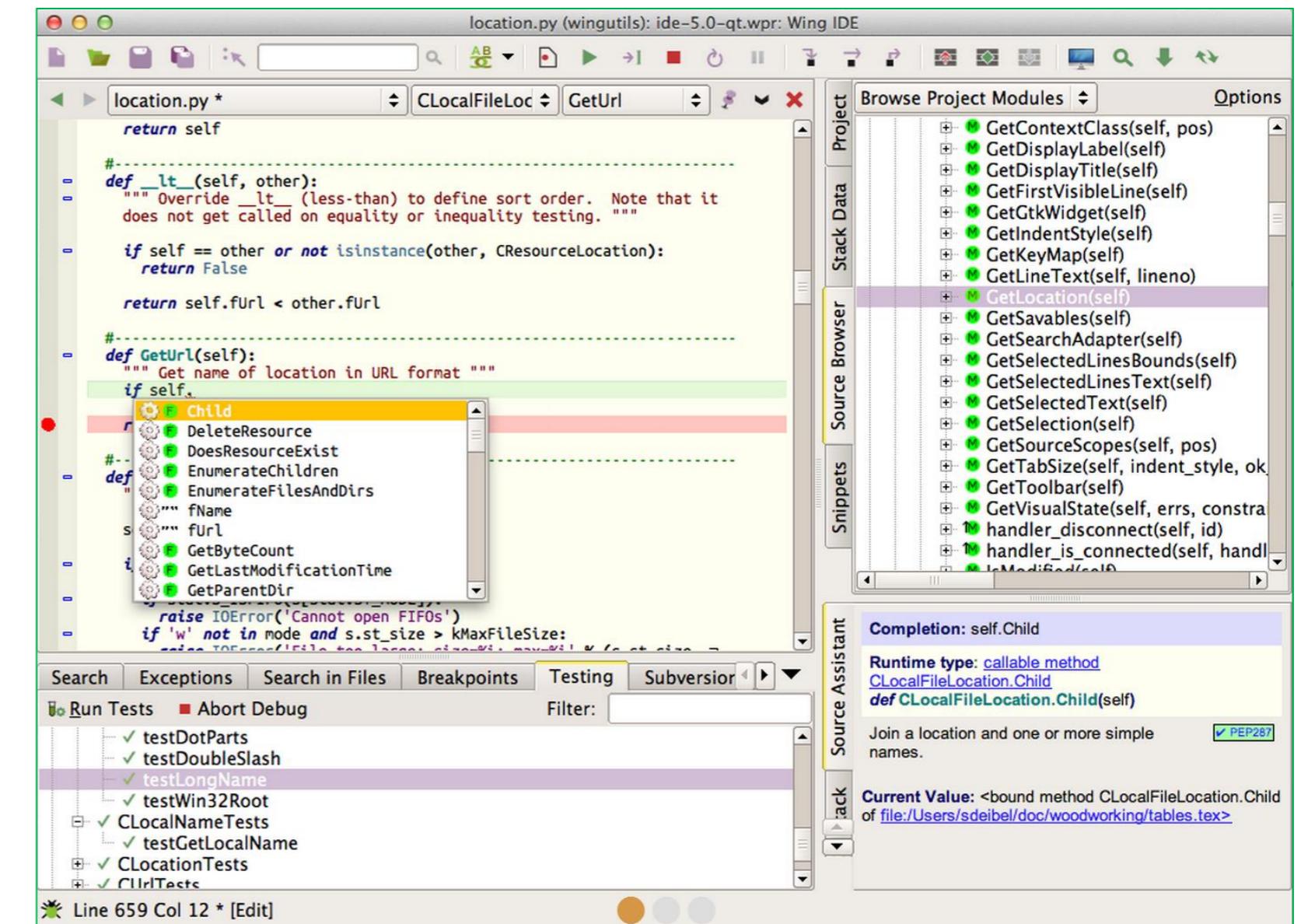


A screenshot of the Sublime Text editor window. The title bar says "untitled • - Sublime Text (UNREGISTERED)". The menu bar includes File, Edit, Selection, Find, View, Goto, Tools, Project, Preferences, and Help. The status bar at the bottom shows "Line 10, Column 20", "Tab Size: 4", and "Python". The main editor area contains the following Python code:

```
1 #e1.1TempConvert.py
2 TempStr = input("请输入带有符号的温度值:")
3 if TempStr[-1] in ['F', 'f']:
4     C = (eval(TempStr[0:-1]) - 32)/1.8
5     print("转换后的温度是{:.2f}C".format(C))
6 elif TempStr[-1] in ['C', 'c']:
7     F = 1.8*eval(TempStr[0:-1]) + 32
8     print("转换后的温度是{:.2f}F".format(F))
9 else:
10    print("输入格式错误")
```

<https://wingware.com>

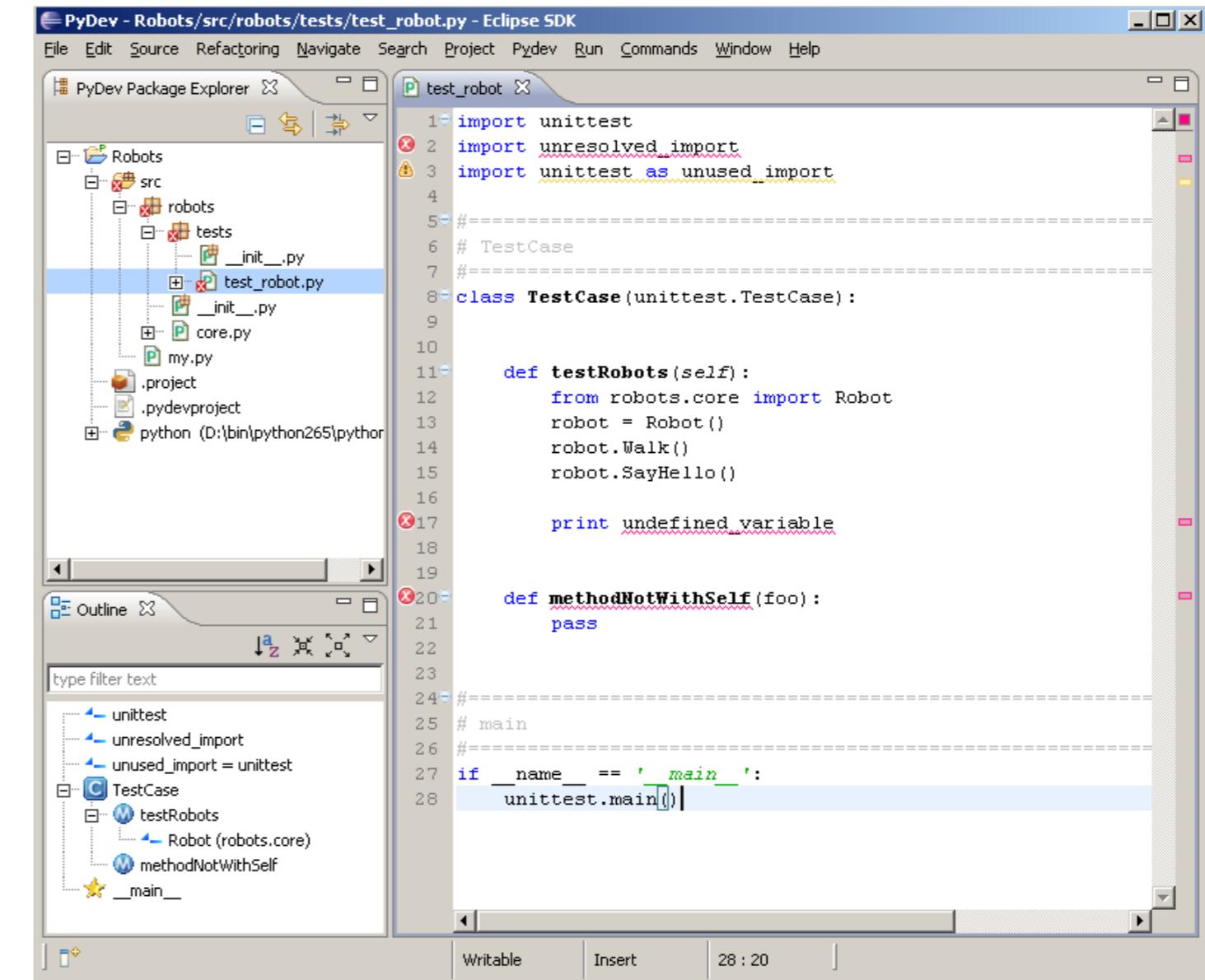
- 收费工具
- 调试功能丰富
- 具有版本控制功能
- 适合多人共同开发
- Win/Linux/Mac OS



Eclipse & PyDev

<https://www.pydev.org>

- Eclipse集成
- 开源IDE, 免费工具
- 需要有一定开发经验
- 适合Eclipse死粉
- Win/Linux/Mac OS



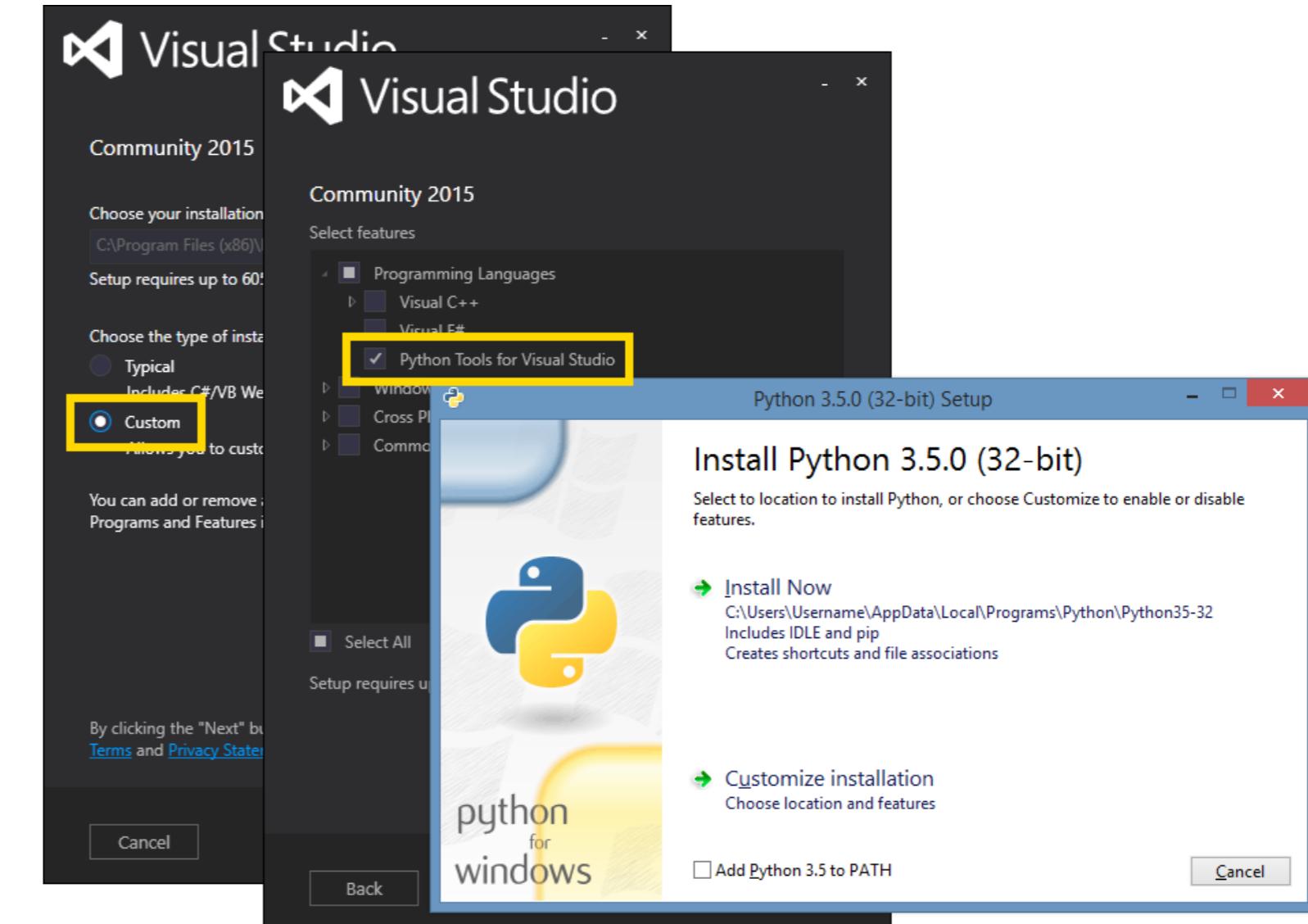
The screenshot shows the Eclipse PyDev interface with the following details:

- PyDev Package Explorer:** Shows a project structure for "Robots" with subfolders "src", "tests", and files like "test_robot.py", "core.py", and ".project".
- Outline View:** Shows a tree of symbols including "unittest", "unresolved_import", "unused_import = unittest", "TestCase", "testRobot", "Robot (robots.core)", "methodNotWithSelf", and "main".
- Editor View:** Displays the content of "test_robot.py". The code includes imports for "unittest" and "Robot", and defines a "TestCase" class with methods "testRobot" and "methodNotWithSelf". It also includes a main block with "unittest.main()". Several syntax errors are highlighted with red squiggly lines.

```
1 import unittest
2 import unresolved_import
3 import unittest as unused_import
4
5 =====
6 # TestCase
7 =====
8 class TestCase(unittest.TestCase):
9
10
11     def testRobot(self):
12         from robots.core import Robot
13         robot = Robot()
14         robot.Walk()
15         robot.SayHello()
16
17         print undefined_variable
18
19
20     def methodNotWithSelf(foo):
21         pass
22
23
24 =====
25 # main
26 =====
27 if __name__ == '__main__':
28     unittest.main()
```

PTVS: Python Tools for Visual Studio

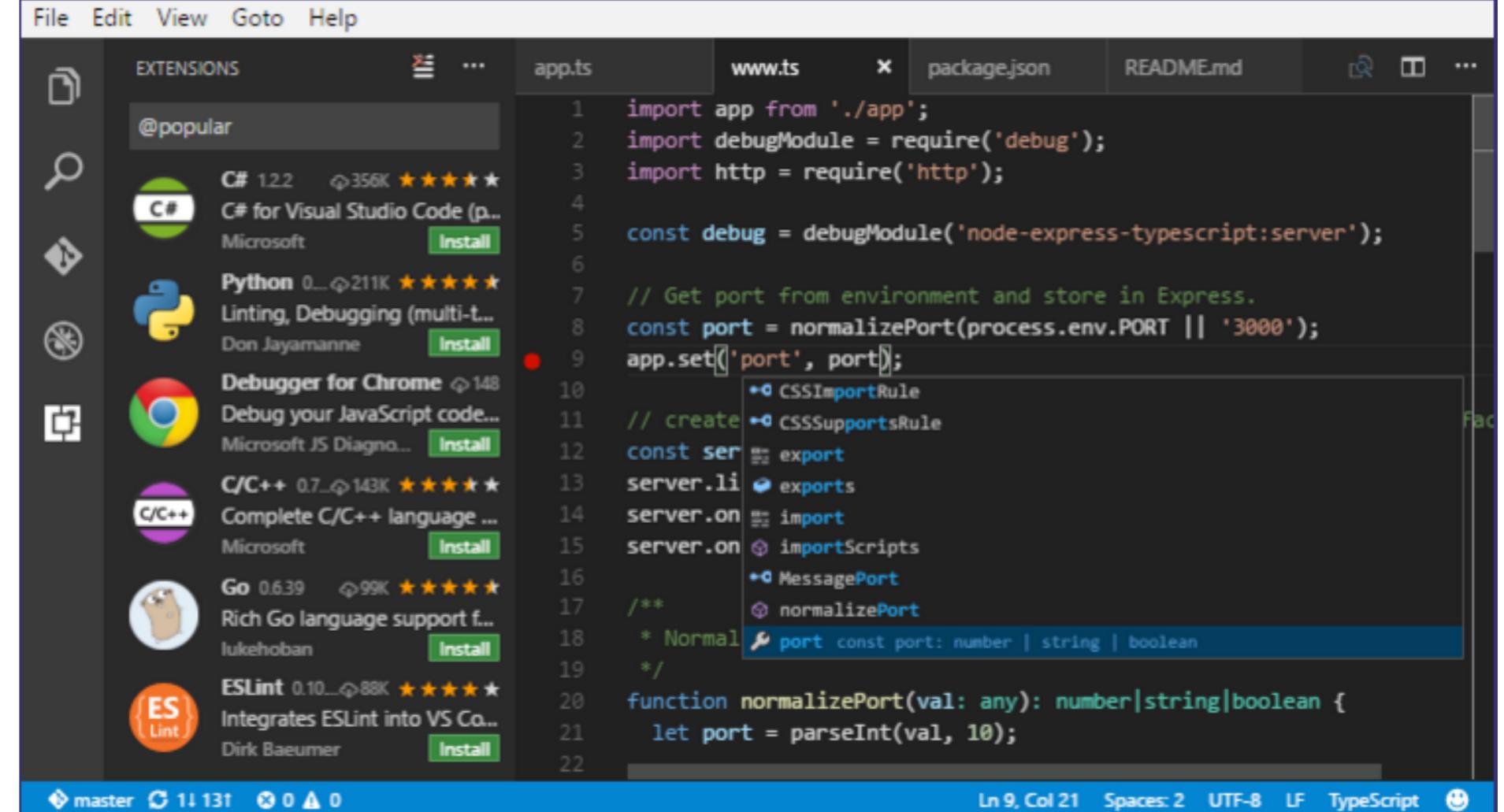
- Visual Studio集成
- 微软出品，收费工具
- 功能非常丰富
- 适合多人共同开发
- Win为主



Visual Studio Code

<https://code.visualstudio.com>

- 独立软件工具
- 微软出品，免费工具
- 功能非常丰富
- 可扩展性很强
- 有AI开发扩展模块
- Win为主



The screenshot shows the Visual Studio Code interface. The left sidebar displays the 'EXTENSIONS' view with several popular extensions listed: C# 1.2.2 by Microsoft, Python 0.1.0 by Don Jayamanne, Debugger for Chrome 0.7.0 by Microsoft, C/C++ 0.7.0 by Microsoft, Go 0.6.39 by lukehoban, and ESLint 0.10.0 by Dirk Baeumer. The main editor area shows a TypeScript file named 'app.ts' with the following code:

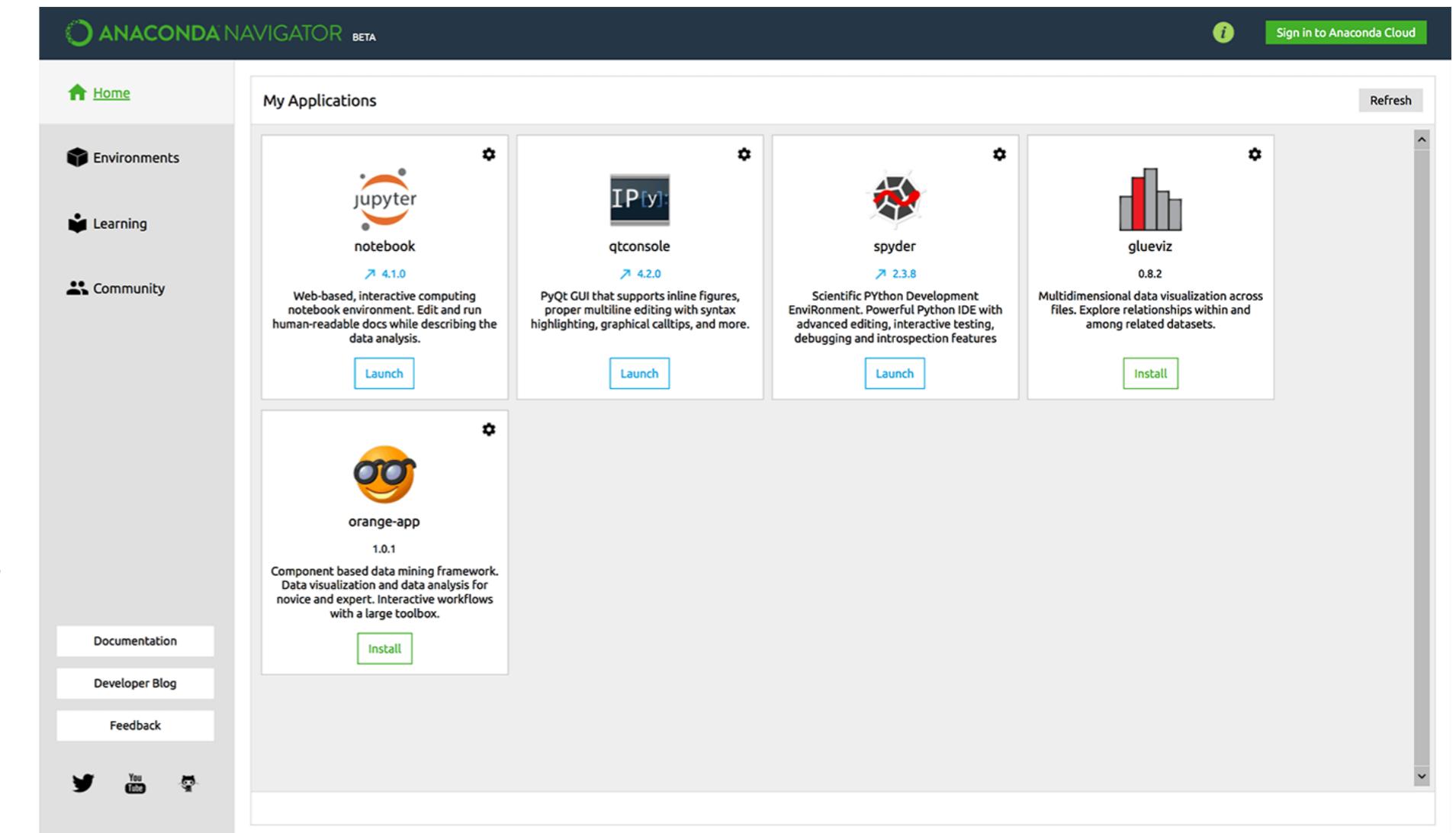
```
1 import app from './app';
2 import debugModule = require('debug');
3 import http = require('http');
4
5 const debug = debugModule('node-express-typescript:server');
6
7 // Get port from environment and store in Express.
8 const port = normalizePort(process.env.PORT || '3000');
9 app.set('port', port);
10
11 // create
12 const server = app.listen(port, () => {
13   console.log(`Server listening on port ${port}`);
14 });
15
16 // Normalize port
17 const normalizePort = (val: string): number => {
18   const port = parseInt(val, 10);
19   if (isNaN(port)) {
20     return undefined;
21   }
22   return port;
23 }
24
25 // Error handling
26 const handleListeningError = (err: Error) => {
27   if (err.code === 'EADDRINUSE') {
28     console.error(`Port ${port} is already in use. Please choose a different port.`);
29     process.exit(1);
30   }
31 }
```

The status bar at the bottom shows: master, 11131, 0, 0, 0, Ln 9, Col 21, Spaces: 2, UTF-8, LF, TypeScript, and a smiley face icon.

Anaconda

<https://www.continuum.io>

- 开源免费
- 支持超过800个第三方库
- 包括多种主流工具
- 适合数据分析及计算领域
- Win/Linux/Mac OS



其他开发工具概述

文本工具类

- ❑ Notepad++
- ❑ Vim & Emacs
- ❑ Sublime Text
- ❑ Atom
- ❑ Komodo Edit

集成工具类

- ❑ Wing
- ❑ PyDev & Eclipse
- ❑ Visual Studio
- ❑ Visual Studio Code
- ❑ Anaconda & Spyder

该选用哪个开发工具呢？

适合自己的IDE才是最好的IDE

从IDLE和Pycharm开始

Python开发工具及环 境配置

Python快速入门

单元小结

单元小结

(1) IDLE开发环境

安装Python基础开发环境、使用IDLE进行程序开发

(2) PyCharm开发环境

了解并初步使用PyCharm开发环境

(3) 其他开发环境概述

Nodepad++、Visual Studio Code、Anaconda等开发工具



Thank you