



清华大学
Tsinghua University

关心学习，更关心你

清华大学学生学习与发展指导中心

答疑坊

只为你答疑解惑

一对一咨询

一心为你

唐仲英计划

全面提升你的公共领导力

写作助理

最贴心，最有针对性的写作辅导



官方微信公众号：**乐学**

扫码关注，获取一手资讯和资源

关注乐学，参与学协活动！

也欢迎加入学协大家庭！

清华大学学生学业发展协会

清华大学学生学习与发展中心指导

清华大学十佳社团协会

一个关心学习，更关心你的社团

小伙伴计划

线上打卡、线下交流

工作坊

“私人订制”的小型讲座

全清华GPA最高的社团

全清华最关心学习的社团

小伙伴计划暑期学习营——零基础Python入门

第一讲：认识Python

张智帅 电子系

清华大学学生学业与发展指导中心
2019-2020学年夏季学期

第一讲：认识Python

- 宏观认识
- 直观认识
- 微观认识

引言



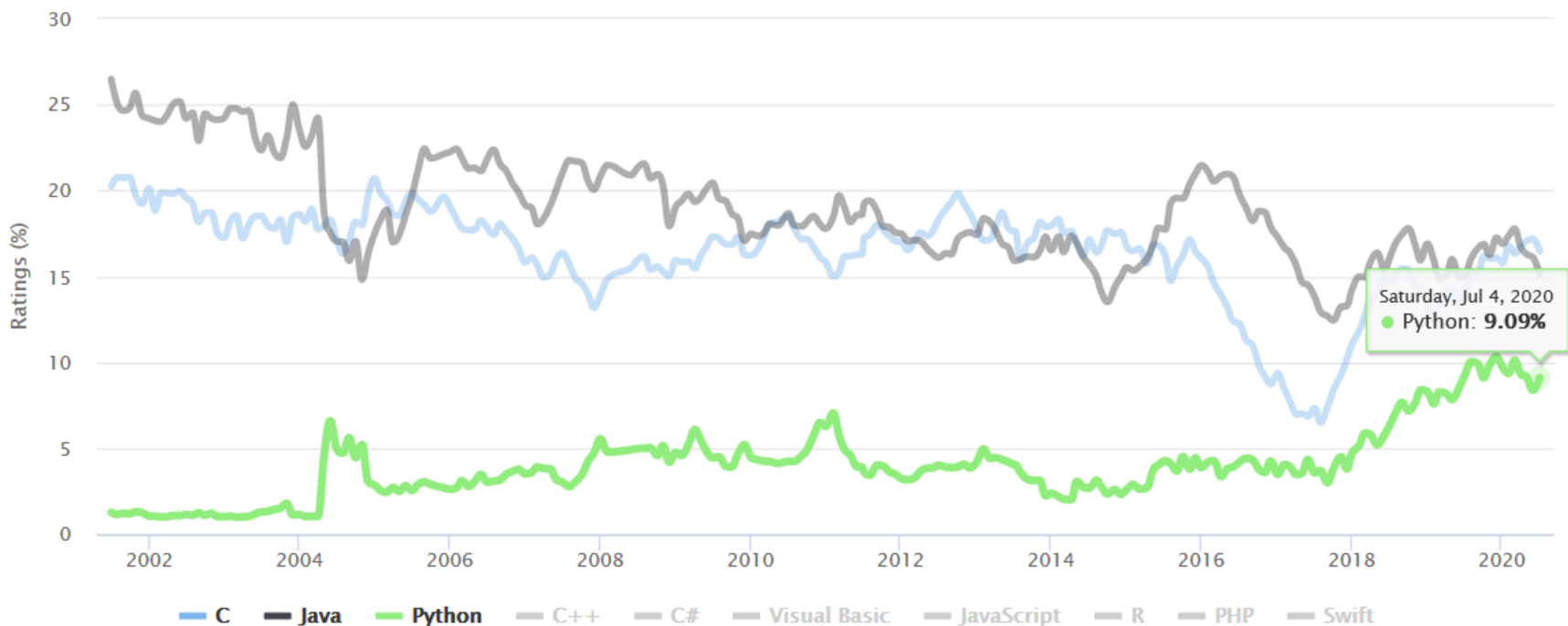
- ❑ 发明者：荷兰籍操作系统工程师Guido van Rossum (1956-)
- ❑ 发明动机：继承ABC语言，负责Amoeba操作系统的交互和异常处理
- ❑ 发明时间：1989圣诞节-1991

引言

- 源远流长，长盛不衰
- Language of the Year: 2007, 2010, 2018

TIOBE Programming Community Index

Source: www.tiobe.com



来源: [TIOBE开发语言排行榜](https://www.tiobe.com)

引言

□ 近年流行的广告

python工资收入

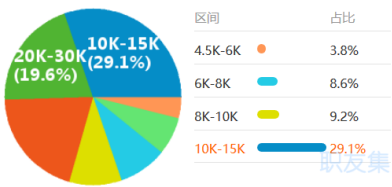
全国python平均工资: ¥13760/月, 取自 17375 份样本, 较 2019 年, 减少 15.3%。统计来自职友集 (jobui.com)

最新招聘

工资收入

就业前景

岗位职责



¥ 13760

近1年 17375 份样本 / 可信度: 高

你觉得该统计数据准确吗?

偏低

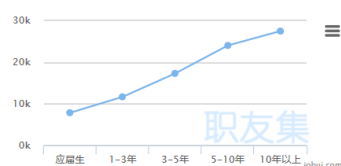
靠谱

偏高

103票 (47%) 45票 (21%) 69票 (32%)

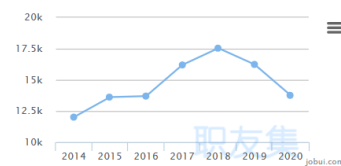
按工作经验统计

数据说明



历年工资变化趋势

数据说明



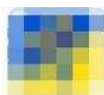
来源: 职友集



Python训练营

广告

会Python的人, 工作都不会太差。
追上同龄人, 就趁现在!



Python

广告

每天处理几十种数据, 效率低
9.9元学个Python, 掌握自动化办
公基本功, 高效工作

9.9元



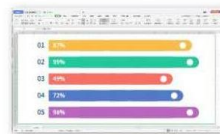
人工智能Python



自动化办公



大数据分析



学习够简单



实操够好玩



模式更贴心



Python天天学

数据分析 | 自动办公 | 人工...

关注公众号

来源: 朋友圈

Python到底可以用来干什么？

- 网络应用开发：架设个人网站、企业网站。
- **自动化脚本**：自动处理机械的、重复的操作。
 - “从每天工作8小时，进化成每天工作10分钟。”
 - 知乎：[用 Python 自动化办公能做到哪些有趣或有用的事情？](#)
- **数据科学**：科学计算、数据分析、可视化。
 - 可以部分替代MATLAB的功能。
 - 可以基本替代R语言、Fortran语言的功能。
- **人工智能：交叉学科，科研热点。**
 - [Nature封面：每天工作21.5小时的AI化学家，8天内完成688个实验，已自主发现一种全新催化剂](#)



人生苦短，我用Python
Life is short, you need Python



Python到底可以用来干什么？

□ 身边的同学用Python做过什么？

- ✓ 写网站、ftp.....
- ✓ 人工智能、科学计算、数据分析.....
- ✓ 发邮件、文本处理、按键控制、视频播放器、自动回复微信、自动下载课件.....

采访你一个问题

你用 Python 做过哪些事情？

科学计算 文本处理 机器学习 深度学习 可视化绘图这些？

zjx@电子

数据清洗，机器学习，网络爬虫，绘制图像，符号求导，文件管理，按键控制

lt@人大金融

人工智能！

自动下载课件

处理 excel 表格

对 PDF 进行文字处理

自动回复微信消息

yyf@电子

发邮件

写网站

ftp

视频播放器

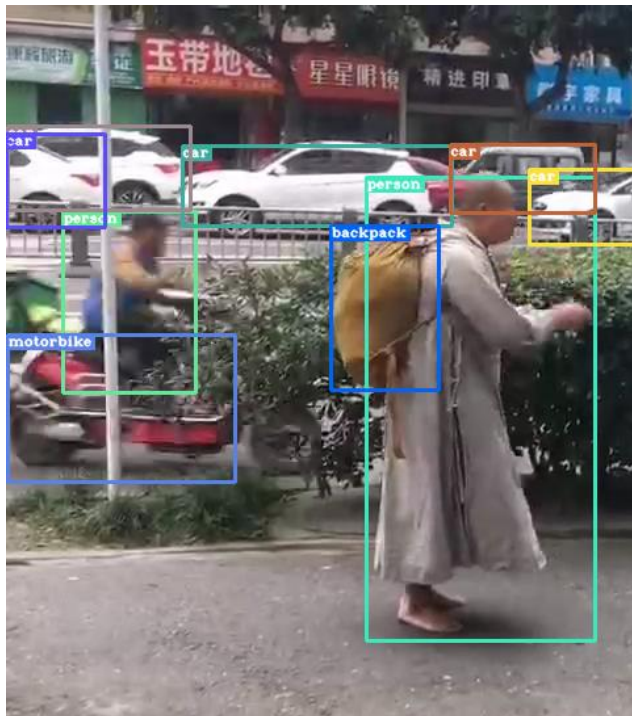
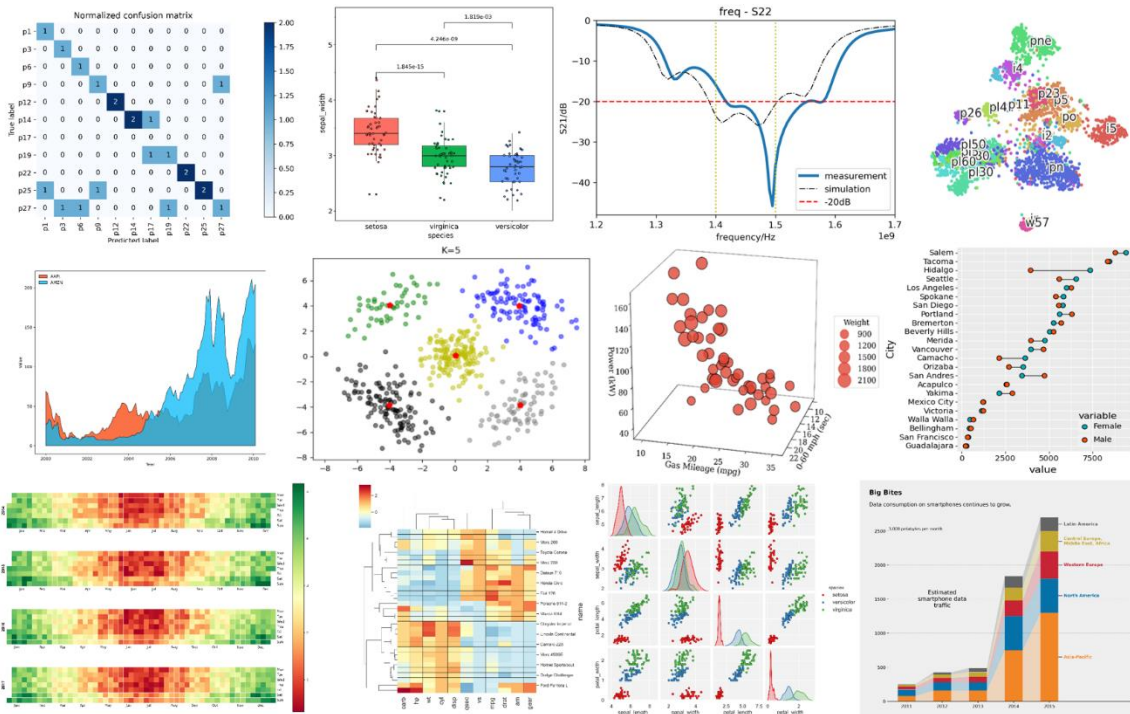
炼丹

lsl@工业工程

Python到底可以用来干什么？

□ 我用Python做过什么？

- ✓ 自动化脚本。处理Word、Excel、PDF，文本处理，微信机器人，爬虫.....
- ✓ 数据处理。科学计算，处理实验数据，绘制图表.....
- ✓ 人工智能。传统机器学习、深度学习.....



Python到底是什么？

python

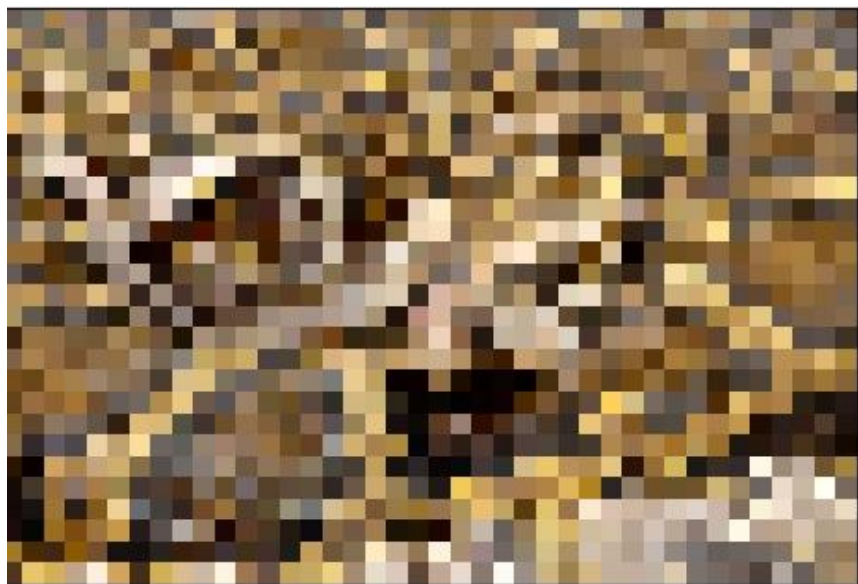
英 ['paɪθən]

美 ['paɪθɑːn]

n. 蟒, 蚺蛇

□ 这个编程语言与蟒蛇有什么关系？

- BBC喜剧《巨蟒的飞行马戏团》（Monty **Python**' s Flying Circus）
- 起名标准：简短、独特、略带神秘感。
- 语言特点：**优雅、明确、简单。**



真正的蟒蛇（已打码）



虚假的蟒蛇

Python到底是什么？

□ Python是一种**解释性的**、**交互性的**、**面向对象的高级**编程语言。

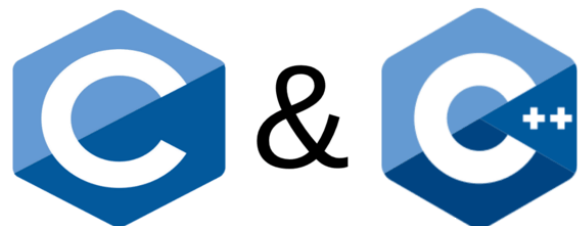
- 高级：抽象层次高，更接近人类语言。
- 优点：**优雅、明确、简单**。❤❤❤
- 缺点：慢。

□ 对比

- 运行速度：C/C++ >> Python
- **开发效率：Python >> C/C++**
 - (1) 语法简单❤
 - (2) 第三方库❤



VS



语法简单

```
1 def reverseLeftWords(self, s: str, n: int) -> str:  
2     return s[n:] + s[:n]
```



```
1 char *reverseLeftWords(char *s, int n)  
2 {  
3     int len = strlen(s), k = 0;  
4     char *res = malloc((len + 1) * sizeof(char));  
5     char *p = s + n;  
6     for (int i = 0; i < len - n; i++)  
7     {  
8         res[k++] = p[i];  
9     }  
10    for (int i = 0; i < n; i++)  
11    {  
12        res[k++] = s[i];  
13    }  
14    res[k] = '\0';  
15    return res;  
16 }
```

VS



课程安排

- 本培训的定位为**初级**，面向零基础。



```
>>> print("Hello, world!")
Hello, world!
>>> a=[1,2,3]
>>> a
[1, 2, 3]
>>>
```



- 我的学习经历：讲座、看书、**getting my hands dirty**
- 教训：

什么需要讲？

- 在哪写、在哪运行、在哪看结果？
- 有什么最主要的语法特征？
- 具体细节去哪查、怎么查？
- 可以怎么与实用场景联系起来？

什么不需要讲？

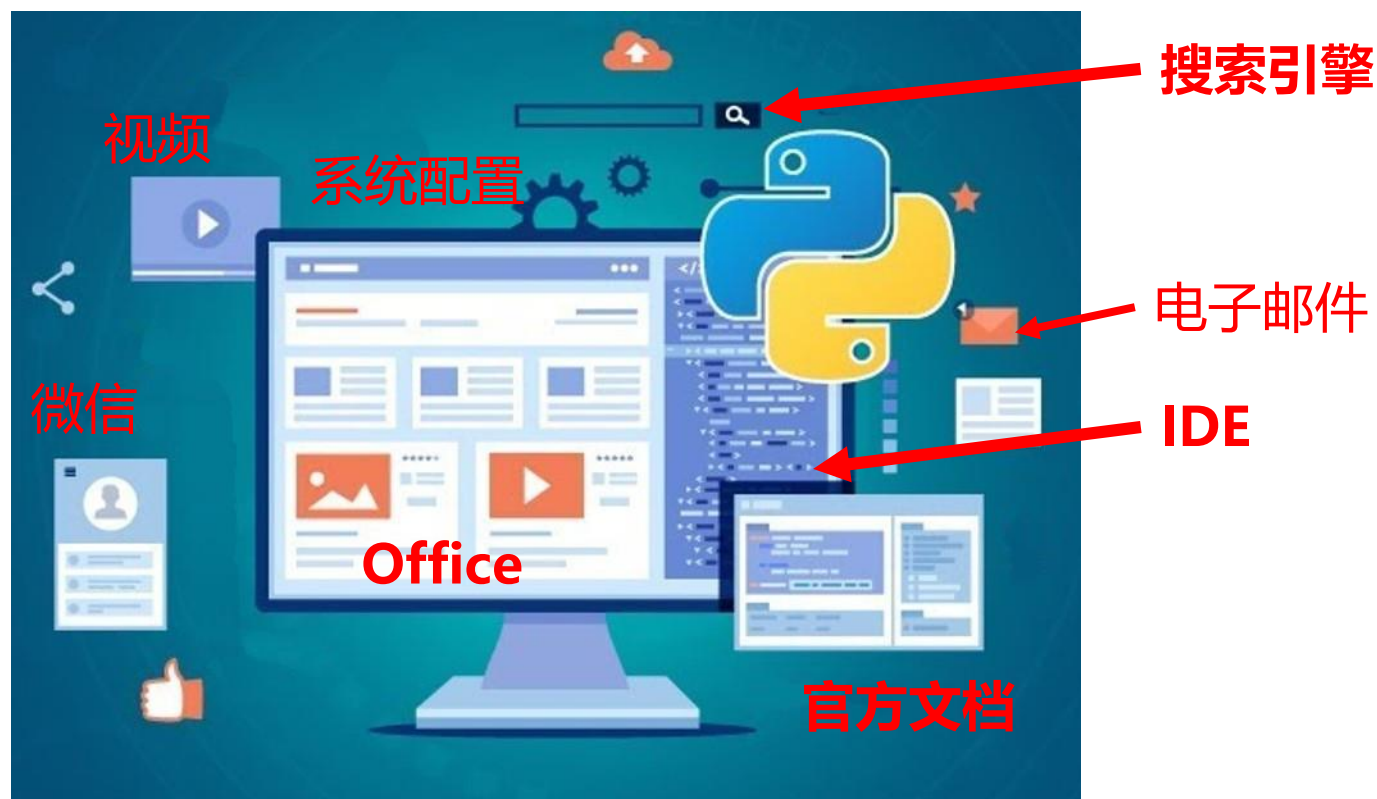
- API有四样写法；
- 语法糖。



课程安排

□ 培训目标:

- 上手一个好用、主流的工具 ⇒ 在哪写、在哪运行、在哪看结果
- 了解最主要的语法特征 ⇒ 对于细节问题, 知道去哪查、怎么查
- 了解python与外界的交互 ⇒ 可以怎么与实用场景联系起来



课程安排

□ 培训目标:

- 上手一个好用、主流的工具 \Rightarrow 在哪写、在哪运行、在哪看结果
- 了解最主要的语法特征 \Rightarrow 对于细节问题, 知道去哪查、怎么查
- 了解python与外界的交互 \Rightarrow 可以怎么与实用场景联系起来

| 周次 | 课程大纲 | 课程内容 |
|-----|-------------|-------------|
| 第一周 | 认识Python | 课程介绍; 安装与使用 |
| 第二周 | Python的基本语法 | 基础语法元素与结构 |
| | | 类与对象、模块 |
| 第三周 | Python的简单应用 | 文档处理入门 |
| | | 数据处理入门 |
| | | 网络爬虫入门 |

配置环境-安装python解释器

- 安装Python 3.8.3, 官网<https://www.python.org/downloads/>



- 参考教程: [可能是全网最详细的 Python 安装教程 \(windows\)](#)

配置环境-Vscode

- ❑ 集成开发环境 (IDE, Integrated Development Environment)
 - =代码编辑器+编译器+调试器.....
- ❑ 经历Vim、Sublime、Spyder、Pycharm、Atom、记事本、Jupyter.....
- ❑ 推荐Vscode:



IDLE
Python Soft...



PyCharm
Apache Lic...



Eclipse
Common P...



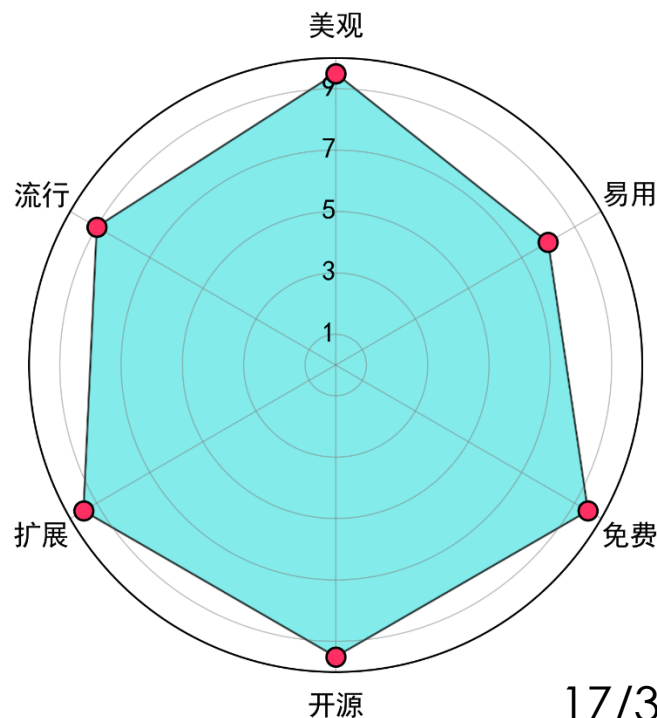
Spyder
MIT License



Microsoft
Visual Studio
Proprietary ...

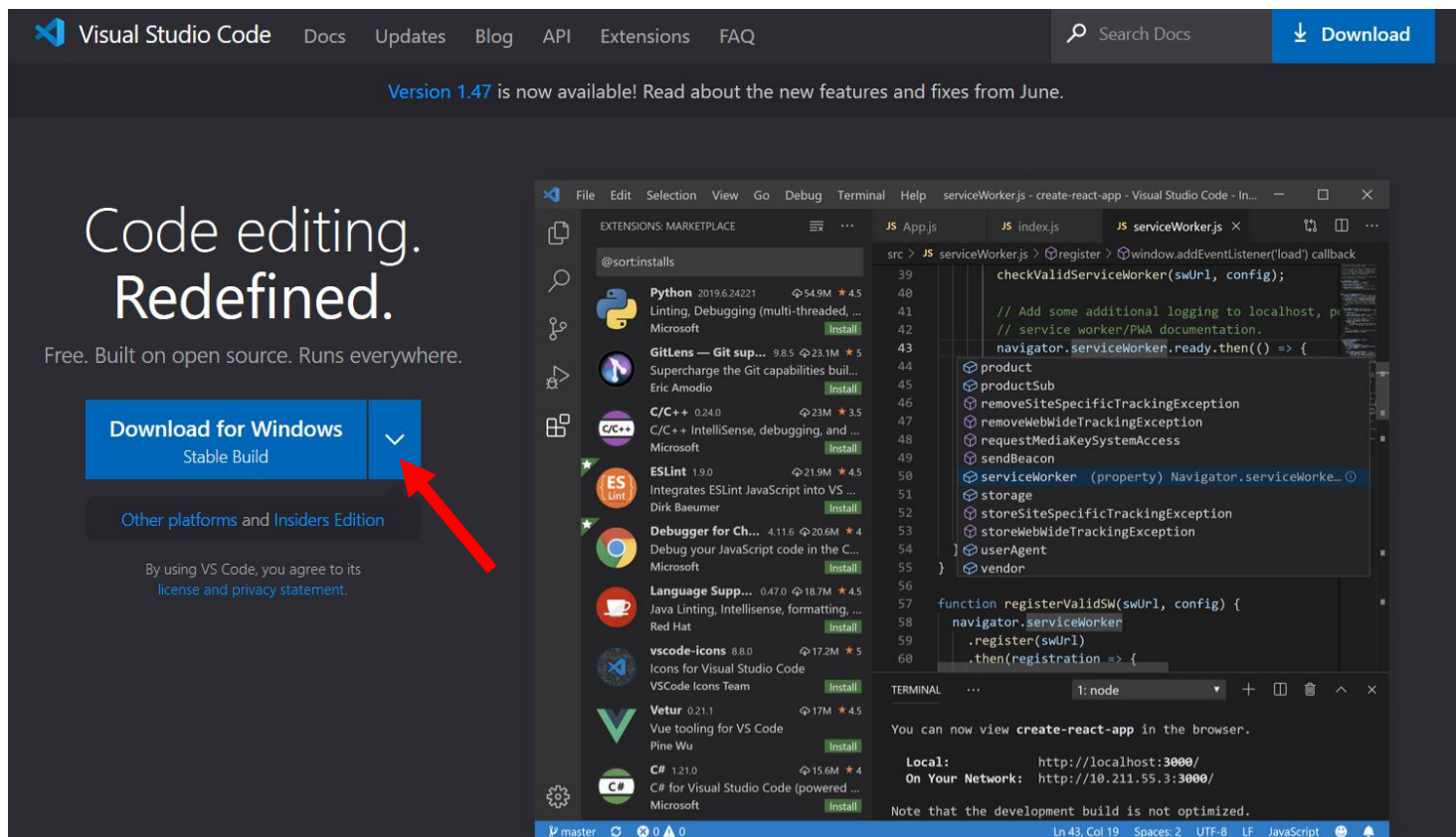


PyDev
Eclipse Pub...



配置环境-Vscode

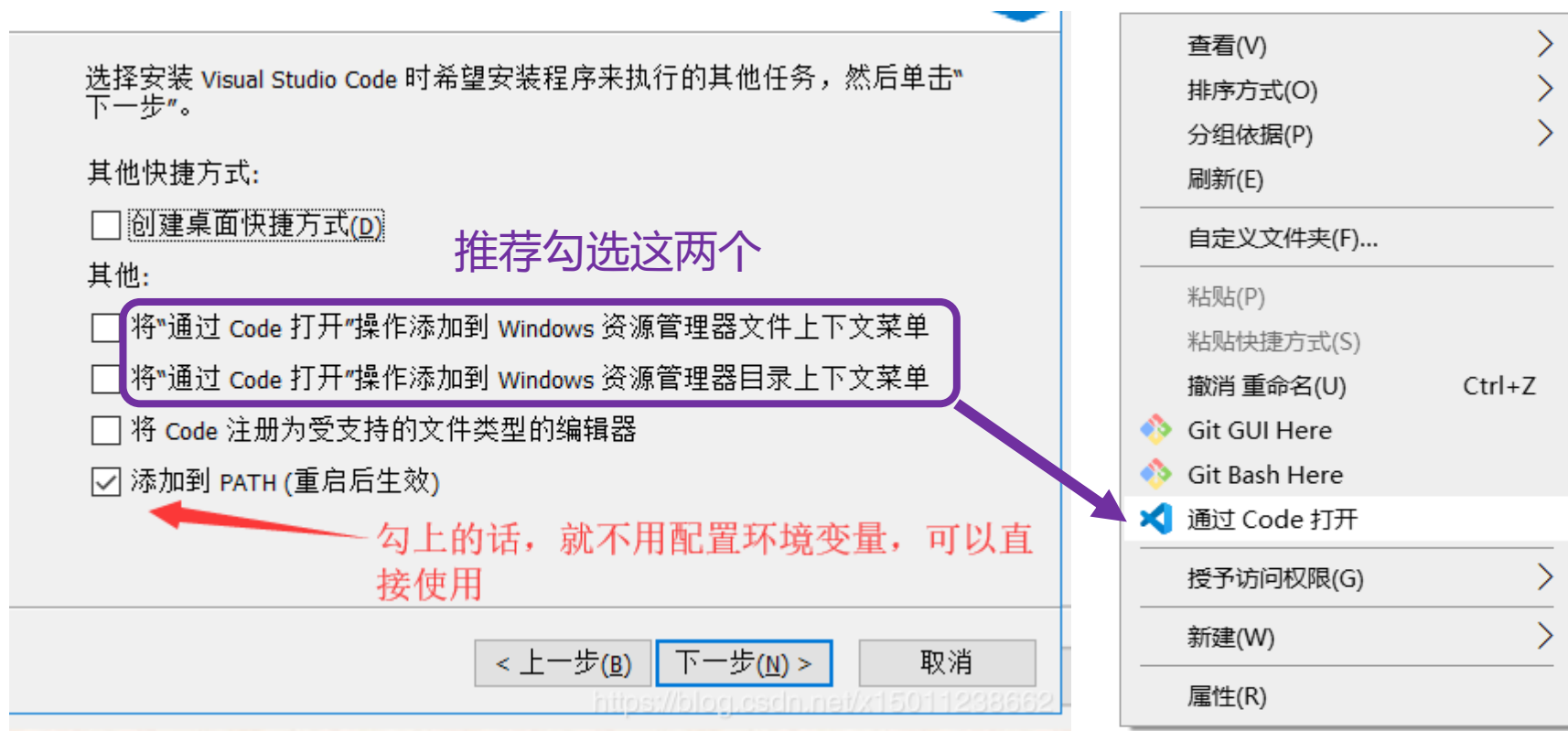
- 安装vscode, 官网<https://code.visualstudio.com/>



- 参考教程: [Windows下安装VScode, 并使用, 以及中文配置](#)

配置环境

- 安装vscode, 官网<https://code.visualstudio.com/>

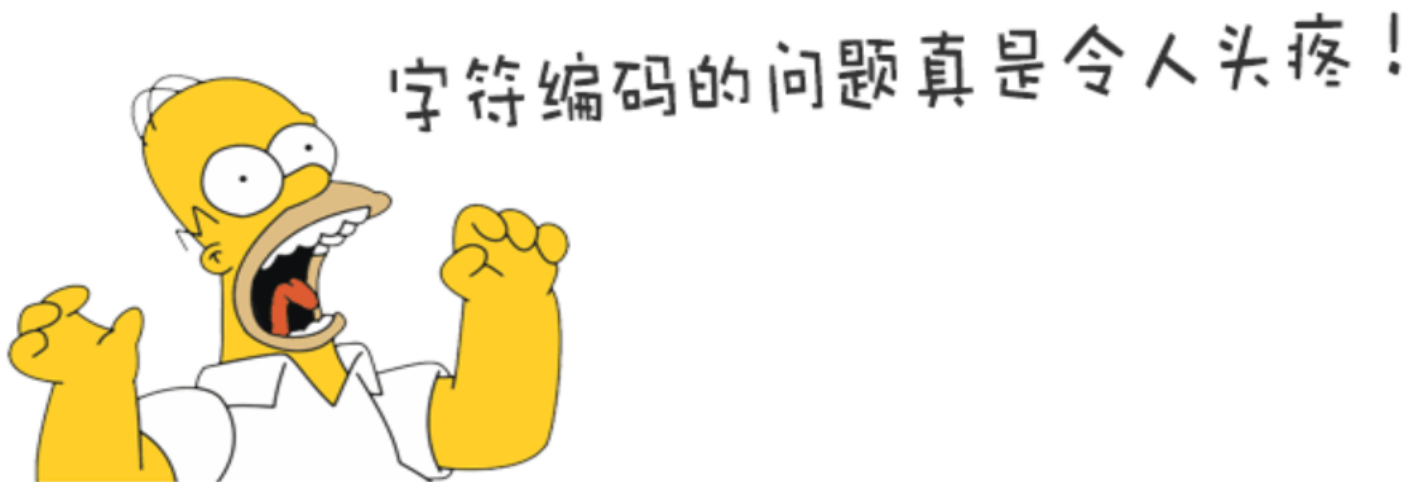


- 参考教程: [Windows下安装VScode, 并使用, 以及中文配置](#)

配置环境

□ Tips:

- 有bug是正常的，先查后问
- 安装路径不建议含有中文
- 安装后最好重启



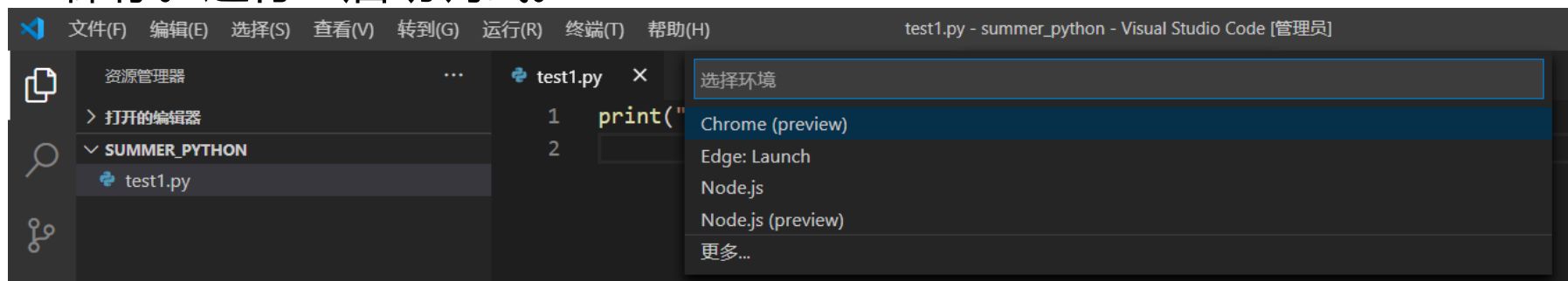
第一句代码

□ 第一句代码

- 打开vscode→新建文件→保存为 “xxxxx.py”
- 键入： `print("Hello world!")`



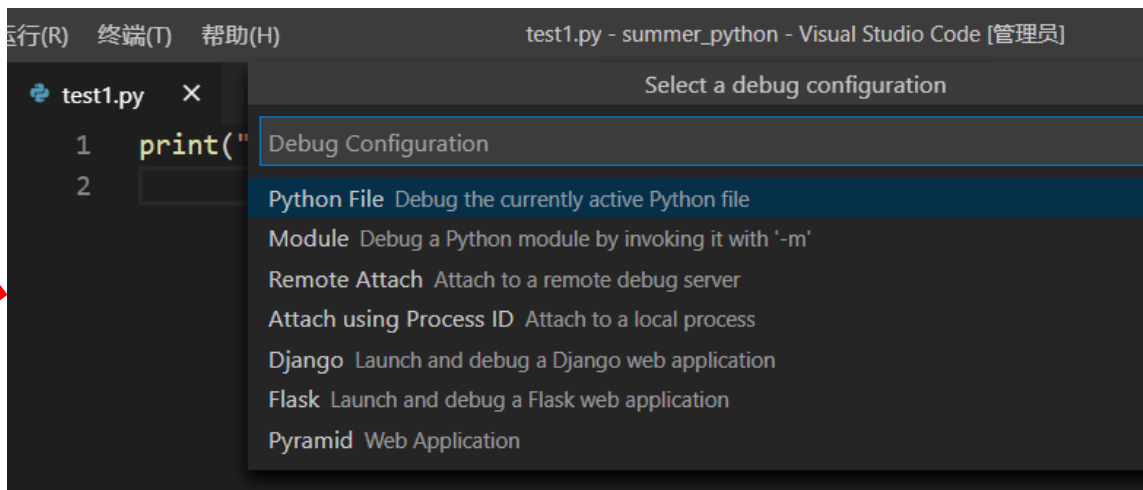
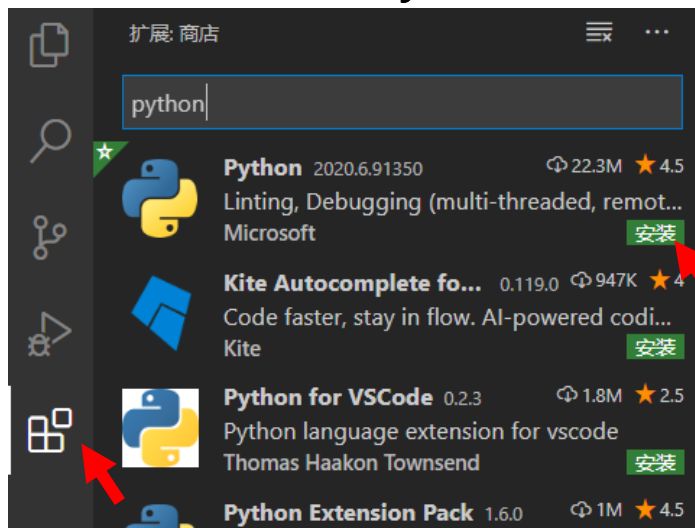
- 保存。运行→启动调试。



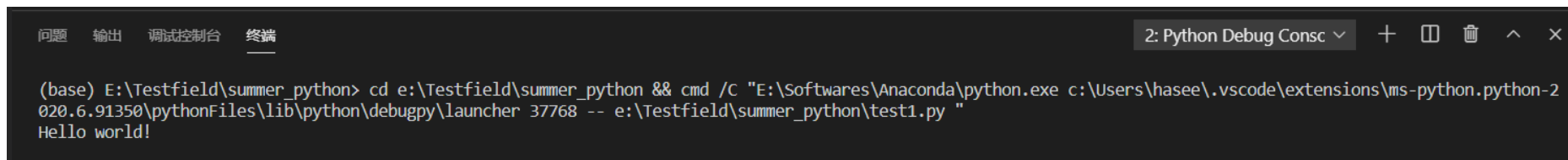
- 没有Python，无法运行。

扩展

- ❑ 安装Python扩展。
- 扩展→搜索Python→安装



- 运行→启动调试→选择Python File
- 输出窗口应有输出:



运行机制



扩展

□ 推荐扩展：

| 插件名 | 主要功能 |
|------------------------------|------------------------|
| python | 支持python |
| code-runner | 右键即可运行代码 |
| vscode-language-pack-zh-hans | 支持中文 |
| bracket-pair-colorizer-2 | 美化括号 |
| indent-rainbow | 高亮缩进 |
| Kite Autocomplete | 智能代码补全（需下载Kite engine） |
| gitlens | 有助于使用git进行团队合作 |
| WakaTime | 记录写代码时间 |

- 更多关于插件的推荐：[Visual Studio Code \(VS code\) 你们都在用吗？或许你们需要看一下这篇博文](#)

第一讲：认识Python

- 宏观认识
- 直观认识
- 微观认识

直观认识Python

- 注释
- 缩进
- 分号
- 代码风格?

```
114     def loss(self, X, y=None):
115         """
116         Compute loss and gradient for the fully-connected net.
117         """
118         X = X.astype(self.dtype)
119         mode = "test" if y is None else "train"
120
121         # Set train/test mode for batchnorm params and dropout param since they
122         # behave differently during training and testing.
123         if self.use_dropout:
124             self.dropout_param["mode"] = mode
125             if self.normalization == "batchnorm":
126                 for bn_param in self.bn_params:
127                     bn_param["mode"] = mode
128         scores = None
129
130         XX = X.reshape(X.shape[0], -1).copy()
131         N, D = XX.shape
132
133         # the params between input layer and first hidden layer
134         hidden_mid_value = []
135         hidden_out_value = []
```

首尾各三个引号的注释块

**以 # 开头的注释
快捷键: ctrl+/**

缩进

行末没有分号

代码风格

- 输入import this直接运行，输出《Python之禅》：

优美胜于丑陋 (Python 以编写优美的代码为目标)

明了胜于晦涩 (优美的代码应当是明了的，命名规范，风格相似)

简洁胜于复杂 (优美的代码应当是简洁的，不要有复杂的内部实现)

.....

间隔胜于紧凑 (优美的代码有适当的间隔，不要奢望一行代码解决问题)

可读性很重要 (优美的代码是可读的)

.....

做也许好过不做，但不假思索就动手还不如不做 (动手之前要细思量)

.....

- PEP 8 编码风格指南：[PEP 8 -- Style Guide for Python Code](#)
 - 译文：[Python PEP8 编码规范中文版](#)

直观认识Python

```
86         for i in range(self.num_layers):
87             self.params["W%d"%(i+1)]=weight_scale*np.random.randn(dims[i],dims[i+1])
88             self.params["b%d"%(i+1)]=np.zeros(dims[i+1])
89
90         if self.normalization is not None:
91             for i in range(len(hidden_dims)):
92                 self.params["gamma"+str("%d"%(i+1))]=np.ones(dims[i+1])
93                 self.params["beta"+str("%d"%(i+1))]=np.zeros(dims[i+1])
```

“格式化文档” 前

```
84         dims = np.hstack((input_dim, hidden_dims, num_classes))
85         for i in range(self.num_layers):
86             self.params["W%d" % (i + 1)] = weight_scale * np.random.randn(
87                 dims[i], dims[i + 1]
88             )
89             self.params["b%d" % (i + 1)] = np.zeros(dims[i + 1])
90
91         if self.normalization is not None:
92             for i in range(len(hidden_dims)):
93                 self.params["gamma" + str("%d" % (i + 1))] = np.ones(dims[i + 1])
94                 self.params["beta" + str("%d" % (i + 1))] = np.zeros(dims[i + 1])
```

“格式化文档” 后

第一讲：认识Python

- 宏观认识
- 直观认识
- 微观认识

数学表达式，变量与赋值

□ **变量** (variable) : 表示特定值的名称。

- 只能由字母（大小写敏感）、数字和下划线组成，且不能以数字开头
- 变量没有默认值，在使用前都必须赋值 (assignment)
- **赋值**：把值绑定或重新绑定到变量上

□ **基本数学运算**

- 加减乘除、整除、取模、指数、绝对值、进制转换.....

```
1  a = 2
2  b = a + 3
3  b = 2 * (3 + 5) - 9 / 4
4  c1 = 2 ** 5 + 9 // 4
5  _f2 = 16 % 5
```

□ **动态类型**，不需要声明变量的类型：int a = 2

输入与输出

□ 输出（打印到终端）

```
7  print(a)
8  print("a=", a)
9  print(a, b, c1)
10 print(a + b + c1)
```

print()

↓ ↓
动作（动作执行的内容）

```
2
a= 2
2 13.75 34
49.75
```

□ 输入（在终端输入）

```
2  x = input()
3  print(x, x, x)
```

```
2020
2020 2020 2020
```

□ 输入2020，能运行吗？

```
2  x = input()
3  print(x + 1)
```

数据类型

□ 类型错误

```
2020
Traceback (most recent call last):
  File "e:\Testfield\summer_python\test5_input.py", line 3, in <module>
    print(x+1)
TypeError: must be str, not int
```

➤ 动态类型也是有类型的。

□ 内置类型：数字、序列、映射、类、实例和异常。

- 数字类型：int, float, complex
- 序列类型：list, str, range, tuple
- 映射类型：dict
- 集合类型：set
- 二进制序列类型：bytes.....

详见第二、第三讲

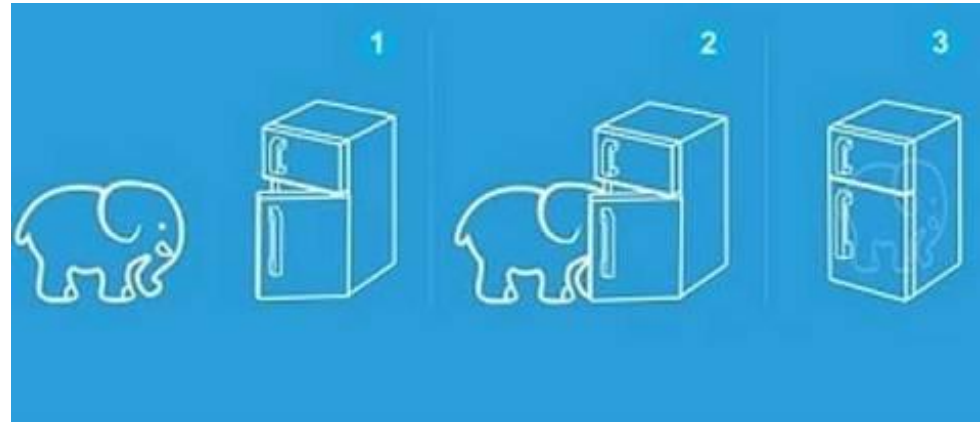
读写文件

□ 写入txt

```
# file IO
filename = "summer.txt"
f = open(filename, "w")
f.write("I miss Tsinghua.")
f.close()
```

□ 写入Word?

```
# Word?
filename = "summer.docx"
f = open(filename, "w")
f.write("I miss Tsinghua.")
f.close()
```



把大象装进冰箱需要几步？

读写更多类型的文件，详见第四讲

怎么解决问题

□ 不要害怕**报错**。

1. **尝试上网搜索以找到答案**（百度、Google、StackOverflow.....）
2. 尝试阅读文档以找到答案
3. 尝试自己检查或试验以找到答案
4. **向你身边的强者朋友打听以找到答案、到群里问**
 - ① 描述你的问题
 - ② **贴上你的代码**
 - ③ **贴上报错信息**
 - ④ 贴上相关截图
 - ⑤ 已经尝试过哪些方法仍然没有解决？（附上相关链接）

□ 强烈建议阅读：How To Ask Questions The Smart Way, **提问的智慧**

课后练习

- 按照前述教程安装Python和Vscode，能够运行以下代码即可：

```
1  for i in range(3):  
2      print("Hello world!")
```

- 加小伙伴小助手微信，从而进入学习营群聊：



小伙伴小助手

中国大陆

- 填写满意度调查问卷：

