

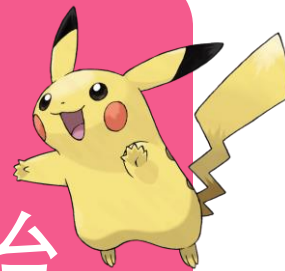
开场白

为什么做这个教程

- 十年编程感受
 - 入门太难了
- 尝试一种入门方式
 - 可操作
 - 由简入繁
 - 有点意思
- 未来?
 - 技术上给点方向
 - 技术外?
 - 逻辑
 - 简历、面试
 - 等



编程入门：从寻找宝可梦开始



编程目标：宝可梦搜索引擎



人

输入：皮卡丘

程序

1. 接收输入

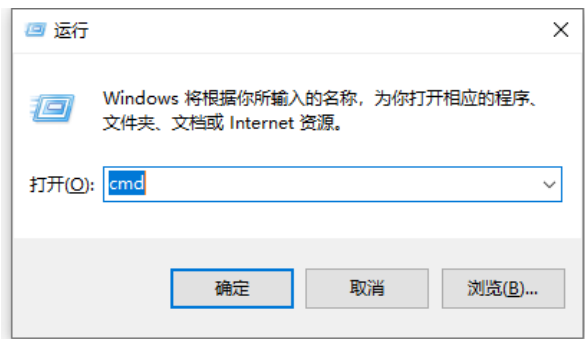
2. 寻找输入的宝可梦

3. 输出寻找结果

输出：

- 皮卡丘
- Pikachu
- <https://wiki.52poke.com/wiki/皮卡丘>

什么是输入、输出



```
C:\Windows\system32\cmd.exe
Microsoft Windows [版本 10.0.18362.2391]
(c) 2019 Microsoft Corporation。保留所有权利。

C:\Users\family>dir C:\Users
驱动器 C 中的卷没有标签。
卷的序列号是 0EA1-2641

C:\Users 的目录

2019/08/23  23:11    <DIR>        .
2019/08/23  23:11    <DIR>        ..
2019/08/28  22:21    <DIR>        family
2019/08/28  22:21    <DIR>        Public
                0 个文件                0 字节
                4 个目录 194,152,841,216 可用字节

C:\Users\family>
```

dir C:\Users

程序名

输入

空格键



回车键

输出

我们的输入、输出是什么



```
C:\Windows\system32\cmd.exe

C:\Users\family>FindPoke 皮卡丘

皮卡丘
Pikachu
https://wiki.52poke.com/wiki/皮卡丘
```

The image shows a Windows command prompt window with a black background and yellow text. The title bar at the top reads 'C:\Windows\system32\cmd.exe'. The command prompt shows the user 'family' at the 'C:\Users\family' directory. The command 'FindPoke 皮卡丘' has been entered and executed. The output consists of three lines: '皮卡丘', 'Pikachu', and a URL 'https://wiki.52poke.com/wiki/皮卡丘'.

如何编写 FindPoke



阶段一

命令行内实现

- Python
- 编码规范: 代码结构、可读性



阶段二

Web页面内实现

- HTML/CSS
- HTTP
- Python



阶段三

系统优化

- 编码规范: 代码结构、可读性、单元测试
- 数据采集: 爬虫
- 数据查找: 索引



阶段四

功能优化

- 异步加载: JavaScript
- 按图查找: 机器学习 (人工智能)



阶段零： 开发环境



在哪儿写代码

文稿



表格



幻灯片



编程？



<https://code.visualstudio.com/Download>

Visual Studio Code, 又叫VSCode, 是一种IDE
IDE (Integrated Development Environment),
即集成开发环境

用哪种语言写代码

5. C++

```
#include  
  
int main()  
{  
    std::cout << "Hello, world!"  
};  
    return 0;  
}
```

12. JavaScript

```
document.write('Hello, world!');
```

22. Python

```
print "Hello, world!"
```

<https://learn.excelwithbusiness.com/blog/post/web-design/say-hello-world-in-28-different-programming-languages>



<https://www.python.org/downloads/release/python-374/>

如果你特别着急，想尝试一下

<https://www.python.org/shell/>



阶段一： 命令行内实现 FindPoke



回顾： 我们的输入、输出是什么



```
C:\Windows\system32\cmd.exe

C:\Users\family>FindPoke 皮卡丘

皮卡丘
Pikachu
https://wiki.52poke.com/wiki/皮卡丘
```

A screenshot of a Windows command prompt window. The title bar shows 'C:\Windows\system32\cmd.exe'. The command prompt shows the command 'FindPoke 皮卡丘' being executed. The output consists of three lines: '皮卡丘', 'Pikachu', and 'https://wiki.52poke.com/wiki/皮卡丘'. The text is displayed in a yellow font on a black background.

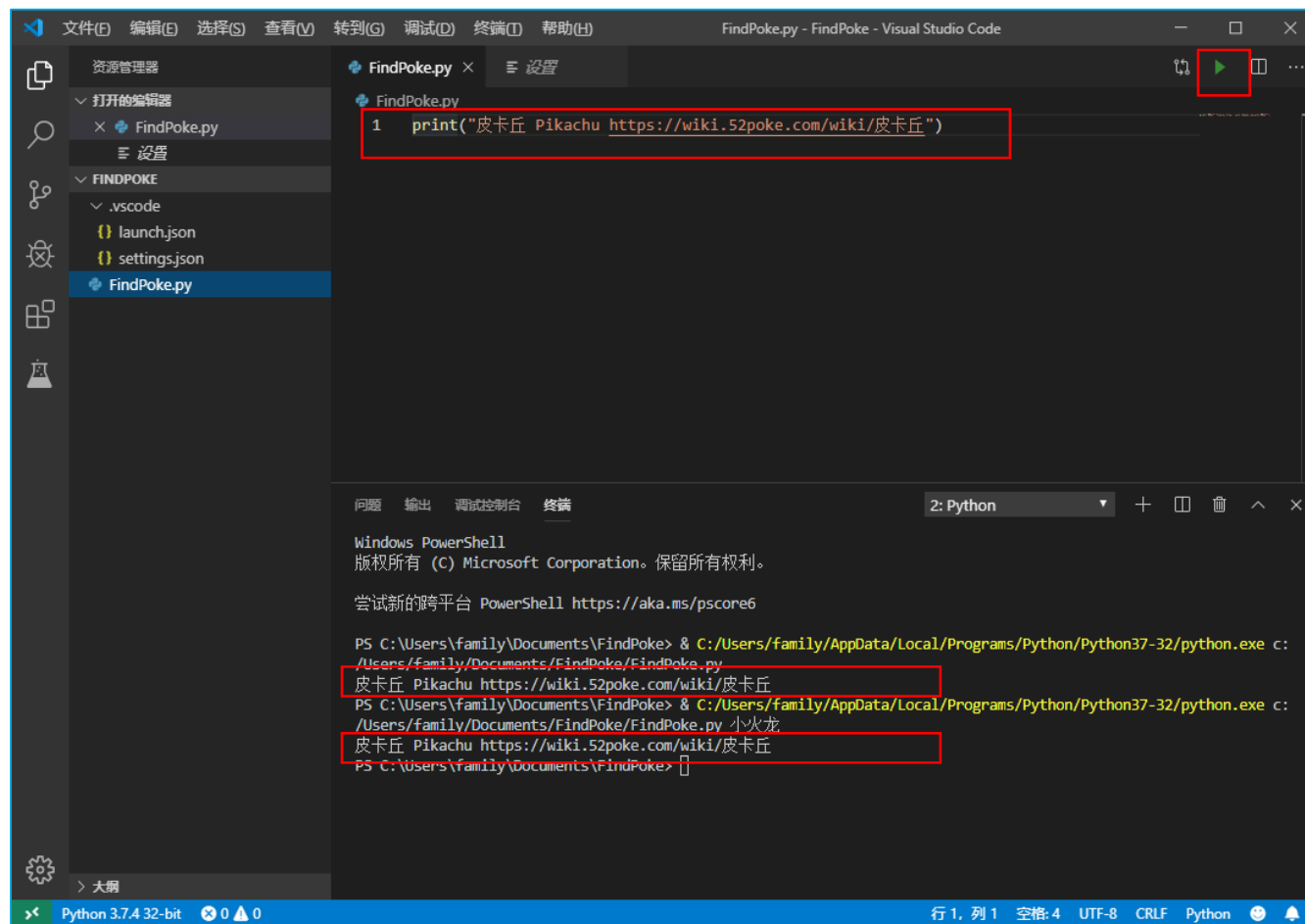
解决输出问题：考虑最简单的情况，永远输出皮卡丘



```
C:\Windows\system32\cmd.exe

C:\Users\family>FindPoke 小火龙

皮卡丘
Pikachu
https://wiki.52poke.com/wiki/皮卡丘
```



```
FindPoke.py
1 print("皮卡丘 Pikachu https://wiki.52poke.com/wiki/皮卡丘")

问题 输出 调试控制台 终端
2: Python

Windows PowerShell
版权所有 (C) Microsoft Corporation。保留所有权利。

尝试新的跨平台 PowerShell https://aka.ms/pscore6

PS C:\Users\family\Documents\FindPoke> & C:/Users/family/AppData/Local/Programs/Python/Python37-32/python.exe c:/Users/family/Documents/FindPoke/FindPoke.py
皮卡丘 Pikachu https://wiki.52poke.com/wiki/皮卡丘
PS C:\Users\family\Documents\FindPoke> & C:/Users/family/AppData/Local/Programs/Python/Python37-32/python.exe c:/Users/family/Documents/FindPoke/FindPoke.py 小火龙
皮卡丘 Pikachu https://wiki.52poke.com/wiki/皮卡丘
PS C:\Users\family\Documents\FindPoke>
```


解决输入问题：忽略查找，直接输出

```
import sys
poke_name = sys.argv[1]
print(poke_name + " Pikachu https://wiki.52poke.com/wiki/皮卡丘")
```

```
PS C:\Users\family\Documents\FindPoke> &
C:/Users/family/AppData/Local/Programs/Python/Python37-32/python.exe
c:/Users/family/Documents/FindPoke/FindPoke.py 小火龙
小火龙 Pikachu https://wiki.52poke.com/wiki/皮卡丘
```

import sys

- 导入已经开发好的功能（模块、包），这里是sys这个模块

sys.argv

- 存放N个输入，sys.argv[0]存放FindPoke.py
- sys.argv[1]存放第一个传入的参数，即小火龙
- sys.argv[2]存放第二个传入的参数，这里没传

解决查找问题： 基于字典实现查找

```
import sys

poke_dict = {
    "皮卡丘": "Pikachu https://wiki.52poke.com/wiki/皮卡丘",
    "小火龙": "Charmander https://wiki.52poke.com/wiki/小火龙"
}

poke_name = sys.argv[1]
print(poke_name + " " + poke_dict[poke_name])
```

```
PS C:\Users\family\Documents\FindPoke> &
C:/Users/family/AppData/Local/Programs/Python/Python37-32/python.exe
c:/Users/family/Documents/FindPoke/FindPoke.py 小火龙
小火龙 Charmander https://wiki.52poke.com/wiki/小火龙
```

```
PS C:\Users\family\Documents\FindPoke> &
C:/Users/family/AppData/Local/Programs/Python/Python37-32/python.exe
c:/Users/family/Documents/FindPoke/FindPoke.py 皮卡丘
皮卡丘 Pikachu https://wiki.52poke.com/wiki/皮卡丘
```



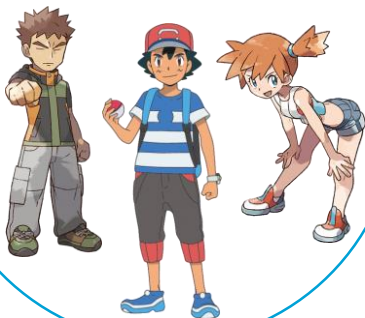
疑问：什么是字典？从类型说起

宝可梦的世界

宝可梦类型



训练师类型



编程语言的世界

数字类型

1 -1
1.2 888888
0

字符串类型

'1' 'Hello'
"皮卡丘" ""

■ ■ ■

列表类型

[] [1, 2]
["皮卡丘", "Hi"]
[1, ['Hi', 2]]

字典类型

{"name": "皮卡丘"}
{ 'Hi' : { 'Hello': 123 } }
{"皮卡丘": ['Hi'] }

自定义类型
class

疑问：字典和列表中有多个元素，如何获取？

```
dict_poke = { 'name': '皮卡丘' }  
# 添加一个字典元素（键、值对）  
dict_poke['english_name']: 'Pikachu'  
  
# 获取字典中存放的name  
print(dict_poke['name'])  
  
# 检查是否存在键： english_name  
key = 'english_name'  
key in dict_poke
```

```
list_poke = ['皮卡丘', '小火龙']  
  
# 获取列表中存放的第一个元素（从0编号），即皮卡丘  
print(list_poke[0])  
  
# 获取列表中存放的第二个元素，即小火龙  
print(list_poke[1])  
  
# 获取列表中存放的倒数第一个元素，即小火龙  
print(list_poke[-1])  
  
# 如果确定list_poke有两个元素，可以一次取两个元素  
pikaqu, xiaohuolong = list_poke  
print(pikaqu, xiaohuolong)
```

井号（#）开头的内容为“注释”，随便写什么都行，常用于对代码进行解释，方便理解



回顾： 获取输入， 构造字典， 输出匹配结果

```
import sys

poke_dict = {
    "皮卡丘": "Pikachu https://wiki.52poke.com/wiki/皮卡丘",
    "小火龙": "Charmander https://wiki.52poke.com/wiki/小火龙"
}

poke_name = sys.argv[1]
print(poke_name + " " + poke_dict[poke_name])
```

```
PS C:\Users\family\Documents\FindPoke> &
C:/Users/family/AppData/Local/Programs/Python/Python37-32/python.exe
c:/Users/family/Documents/FindPoke/FindPoke.py 小火龙
小火龙 Charmander https://wiki.52poke.com/wiki/小火龙
```

```
PS C:\Users\family\Documents\FindPoke> &
C:/Users/family/AppData/Local/Programs/Python/Python37-32/python.exe
c:/Users/family/Documents/FindPoke/FindPoke.py 皮卡丘
皮卡丘 Pikachu https://wiki.52poke.com/wiki/皮卡丘
```

问题： 我们有上百个宝可梦，如何准备数据？

全国编号	中文	宝可梦 日文	英文
第一世代			
#001	妙蛙种子	フシギダネ	Bulbasaur
#002	妙蛙草	フシギソウ	Ivysaur
#003	妙蛙花	フシギバナ	Venusaur
#004	小火龙	ヒトカゲ	Charmander
#005	火恐龙	リザード	Charmeleon
#006	喷火龙	リザードン	Charizard
#007	杰尼龟	ゼニガメ	Squirtle
#008	卡咪龟	カメール	Wartortle
#009	水箭龟	カメックス	Blastoise
#010	绿毛虫	キャタピー	Caterpie
#011	铁甲蛹	トランセル	Metapod
#012	巴大蝶	バタフリー	Butterfree
#013	独角虫	ビードル	Weedle
#014	铁壳蛹	コクーン	Kakuna
#015	大针蜂	スピアー	Beedrill
#016	波波	ポッポ	Pidgey
#017	比比鸟	ピジョン	Pidgeotto
#018	大比鸟	ピジョット	Pidgeot
#019	小拉达	コラッタ	Rattata
#020	拉达	ラッタ	Raticate
#021	烈雀	オニスズメ	Spearow
#022	大嘴雀	オニドリル	Fearow
#023	阿柏蛇	アーボ	Ekans
#024	阿柏怪	アーボック	Arbok
#025	皮卡丘	ピカチュウ	Pikachu

```
import sys
```

```
poke_dict = {  
    "皮卡丘": "Pikachu https://wiki.52poke.com/wiki/皮卡丘",  
    "小火龙": "Charmander https://wiki.52poke.com/wiki/小火龙"
```

难道要在这儿手动加上吗

```
}
```

```
poke_name = sys.argv[1]  
print(poke_name + " " + poke_dict[poke_name])
```


如何准备数据：一种准备数据的方法

CTRL+C复制

全国编号	宝可梦		
	中文	日文	英文
第一世代			
#001	妙蛙种子	フシギダネ	Bulbasaur
#002	妙蛙草	フシギソウ	Ivysaur
#003	妙蛙花	フシギバナ	Venusaur
#004	小火龙	ヒトカゲ	Charmander
#005	火恐龙	リザード	Charmeleon
#006	喷火龙	リザードン	Charizard
#007	杰尼龟	ゼニガメ	Squirtle
#008	卡咪龟	カメール	Wartortle
#009	水箭龟	カメックス	Blastoise
#010	绿毛虫	キャタピー	Caterpie
#011	铁甲蛹	トランセル	Metapod
#012	巴大蝶	バタフリー	Butterfree
#013	独角虫	ビードル	Weedle
#014	铁壳蛹	コクーン	Kakuna
#015	大针蜂	スピアー	Beedrill
#016	波波	ポッポ	Pidgey
#017	比比鸟	ビジョン	Pidgeotto
#018	大比鸟	ビジョット	Pidgeot
#019	小拉达	コラッタ	Rattata
#020	拉达	ラッタ	Raticate
#021	烈雀	オニスズメ	Spearow
#022	大嘴雀	オニドリル	Fearow

创建poke.txt文件，CTRL+V粘贴

```
文件(F) 编辑(E) 选择(S) 查看(V) 转到(G) 调试(D) 终端(T) 帮助(H)
资源管理器
  打开的编辑器 1个未保存
    poke.txt
  FINDPOKE
    .vscode
      launch.json
      settings.json
      FindPoke.py
    poke.txt
poke.txt
1 全国编号 宝可梦
2 中文 日文 英文
3 第一世代
4 #001 妙蛙种子 フシギダネ Bulbasaur
5 #002 妙蛙草 フシギソウ Ivysaur
6 #003 妙蛙花 フシギバナ Venusaur
7 #004 小火龙 ヒトカゲ Charmander
8 #005 火恐龙 リザード Charmeleon
9 #006 喷火龙 リザードン Charizard
10 #007 杰尼龟 ゼニガメ Squirtle
11 #008 卡咪龟 カメール Wartortle
12 #009 水箭龟 カメックス Blastoise
13 #010 绿毛虫 キャタピー Caterpie
14 #011 铁甲蛹 トランセル Metapod
15 #012 巴大蝶 バタフリー Butterfree
16 #013 独角虫 ビードル Weedle
17 #014 铁壳蛹 コクーン Kakuna
18 #015 大针蜂 スピアー Beedrill
19 #016 波波 ポッポ Pidgey
20 #017 比比鸟 ビジョン Pidgeotto
21 #018 大比鸟 ビジョット Pidgeot
22 #019 小拉达 コラッタ Rattata
23 #020 拉达 ラッタ Raticate
24 #021 烈雀 オニスズメ Spearow
25 #022 大嘴雀 オニドリル Fearow
26 #023 阿柏蛇 アーボ Ekans
27 #024 阿柏怪 アーボック Arbok
28 #025 皮卡丘 ピカチュウ Pikachu
29 #026 雷丘 ライチュウ Raichu
30 #027 穿山鼠 サンド Sandshrew
31 #028 穿山王 サンドパン Sandslash
32 #029 尼多兰 ニドラン♀ Nidoran♀
33 #030 尼多娜 ニドリーナ Nidorina
34 #031 尼多后 ニドクイン Nidoqueen
35 #032 尼多朗 ニドラン♂ Nidoran♂
36 #033 尼多力诺 ニドリーノ Nidorino
37 #034 尼多王 ニドキング Nidoking
38 #035 皮皮 ビッピ Clefairy
39 #036 皮可西 ビクシー Clefable
```

如何准备数据：将数据读入字典

```
import sys

# 定义空字典，存放宝可梦数据
poke_dict = {}

# 读取宝可梦数据
file = open('poke.txt', 'r', encoding='utf-8')
content = file.read()
file.close()

# 将宝可梦数据导入字典
line_list = content.split('\n')
for line in line_list:
    field_list = line.strip().split('\t')
    if len(field_list) == 4:
        id, name, japan_nam, english_name = field_list
        url = 'https://wiki.52poke.com/wiki/' + name
        poke_dict[name] = [id, japan_nam, english_name, url]

# 读取输入的参数，即待查找的宝可梦名称
poke_name = sys.argv[1]

# 查找宝可梦
print(poke_dict[poke_name])
```

open

- 接受3个参数，分别表示要操作的文件名、文件操作方式（r，即read，表示读取）、文件编码（用VSCode打开文件，查看右下角编码）



file.read

- 读取文件内容

file.close

- 关闭文件

如何准备数据： 将数据读入字典

```
import sys

# 定义空字典，存放宝可梦数据
poke_dict = {}

# 读取宝可梦数据
file = open('poke.txt', 'r', encoding='utf-8')
content = file.read()
file.close()

# 将宝可梦数据导入字典
line_list = content.split('\n')
for line in line_list:
    field_list = line.strip().split('\t')
    if len(field_list) == 4:
        id, name, japan_nam, english_name = field_list
        url = 'https://wiki.52poke.com/wiki/' + name
        poke_dict[name] = [id, japan_nam, english_name, url]

# 读取输入的参数，即待查找的宝可梦名称
poke_name = sys.argv[1]

# 查找宝可梦
print(poke_dict[poke_name])
```

content.split

- content是字符串，split可以按给定的字符串对content做切割，如
`'a|b|c'.split('|')`会返回`['a', 'b', 'c']`

`'\n'`

- 表示换行符，虽然肉眼看不见，但是行间都有`'\n'`分隔

for XXX in YYY:

- 依次读取YYY的每一个元素（又叫遍历），通常YYY是一个列表

line.strip

- line是字符串，strip表示将字符串前、后的空白字符删掉（空格、制表符、换行符都属于空白字符），制表符写做`'\t'`，换行符写做`'\n'`

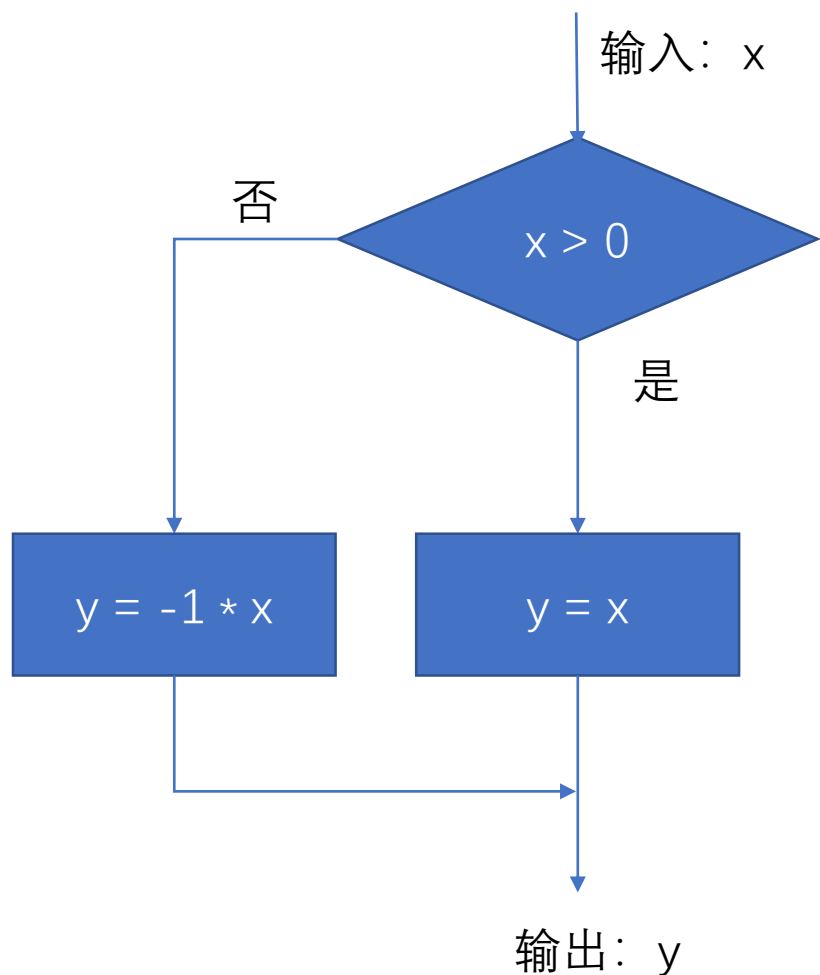
if XXX:

- 当XXX条件满足时执行，仔细查询复制的数据，会有一些非法数据，这里会检查下是一行中是否有4个字段

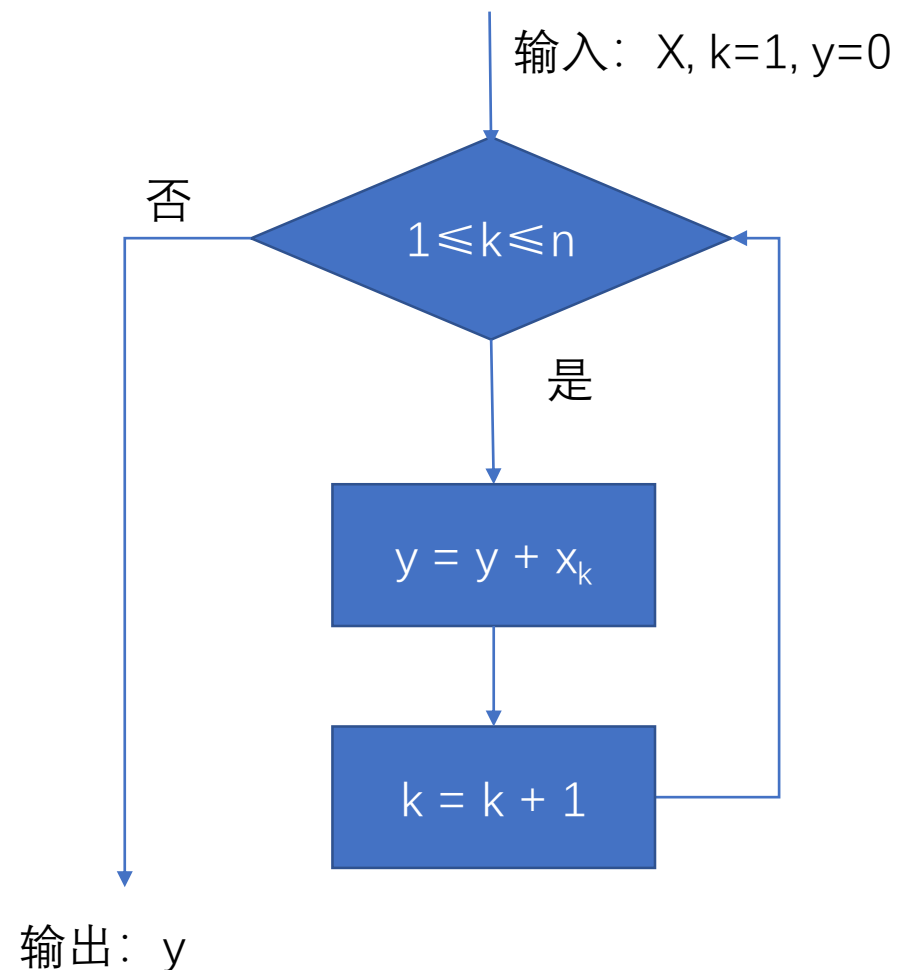


疑问：什么是for、if？从流程控制说起

使用if，求x的绝对值 $y = |x|$



使用for，求 $X = x_1, x_2, \dots, x_n$ 的和



思考：什么是编程？

疑问：什么是字典？从类型说起

宝可梦的世界



训练师类型 (Trainer Type) diagram showing Ash, Brock, Misty, and others.

编程语言的世界

数字类型 (Number Type)

1 -1
1.2 888888
0

字符串类型 (String Type)

'1' 'Hello'
"皮卡丘" ""

...

列表类型 (List Type)

[] [1, 2]
["皮卡丘", "Hi"]
[1, ["Hi", 2]]

字典类型 (Dictionary Type)

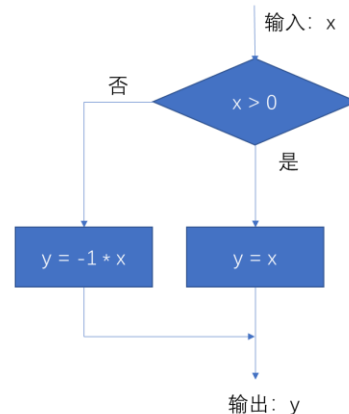
{"name": "皮卡丘"}
{"Hi": {"Hello": 123}}
{"皮卡丘": ["Hi"]}

自定义类型 (Custom Type)
class

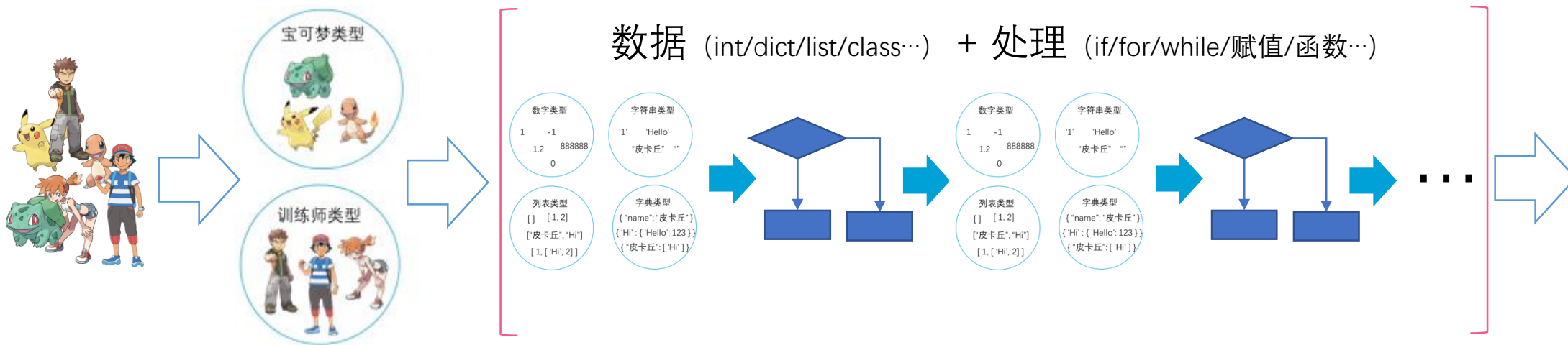
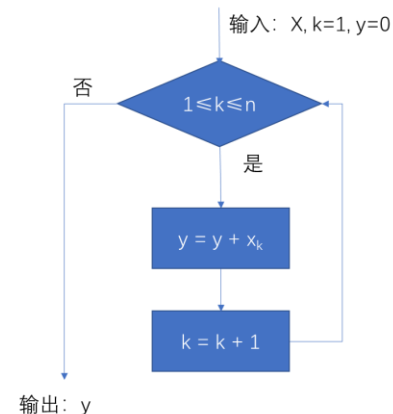
<https://www.runoob.com/python3/python3-data-type.html>

疑问：什么是for、if？从流程控制说起

使用if，求x的绝对值 $y = |x|$



使用for，求 $X = x_1, x_2, \dots, x_n$ 的和



思考：什么是编程？

做饭的教程



<https://www.xiachufang.com/recipe/104003927/>

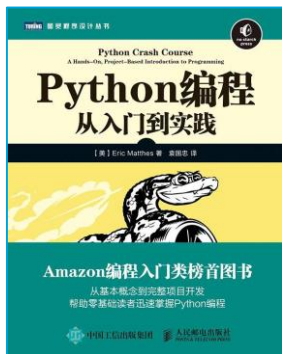
食材



加工（洗、切、炒、混合）



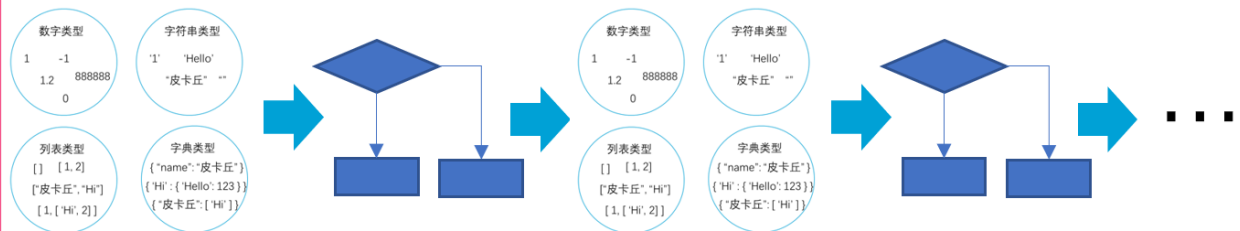
编程的教程



编程入门：从寻找宝可梦开始



数据 (int/dict/list/class...) + 处理 (if/for/while/赋值/函数...)





代码有漏洞： 初见Bug

```
import sys

# 定义空字典，存放宝可梦数据
poke_dict = {}

# 读取宝可梦数据
file = open('poke.txt', 'r', encoding='utf-8')
content = file.read()
file.close()

# 将宝可梦数据导入字典
line_list = content.split('\n')
for line in line_list:
    field_list = line.strip().split('\t')
    if len(field_list) == 4:
        id, name, japan_nam, english_name = field_list
        url = 'https://wiki.52poke.com/wiki/' + name
        poke_dict[name] = [id, japan_nam, english_name, url]

# 读取输入的参数，即待查找的宝可梦名称
poke_name = sys.argv[1]

# 查找宝可梦
print(poke_dict[poke_name])
```

万一没有输入宝可梦？

万一输入的宝可梦不在字典里？

提示1

- sys.argv是一个列表，可以通过len(sys.argv)获取元素个数

提示2

- 利用in来判断字典是否存在某一个键

代码有漏洞：修Bug

```
import sys

# 定义空字典，存放宝可梦数据
poke_dict = {}

# 读取宝可梦数据
file = open('poke.txt', 'r', encoding='utf-8')
content = file.read()
file.close()

# 将宝可梦数据导入字典
line_list = content.split('\n')
for line in line_list:
    field_list = line.strip().split('\t')
    if len(field_list) == 4:
        id, name, japan_name, english_name = field_list
        url = 'https://wiki.52poke.com/wiki/' + name
        poke_dict[name] = [id, japan_name, english_name, url]

# 读取输入的参数，即待查找的宝可梦名称
poke_name = sys.argv[1]

# 查找宝可梦
print(poke_dict[poke_name])
```

万一没有输入宝可梦？

万一输入的宝可梦不在字典里？

提示1

- `sys.argv`是一个列表，可以通过 `len(sys.argv)` 获取元素个数

提示2

- 利用 `in` 来判断字典是否存在某一个键



代码可读性： 我的代码我做主？

```
line_list = content.split('\n')
for line in line_list:
    field_list = line.strip().split('\t')
if len(field_list) == 4:
    id, name, japan_name, english_name = field_list
    url = 'https://wiki.52poke.com/wiki/' + name
    poke_dict[name] = [id, japan_name, english_name, url]
```

英文贼溜

```
l = c.split('\n')
for _l in l:
    f = _l.strip().split('\t')
if len(f) == 4:
    i, n, j, e = f
    u = 'https://wiki.52poke.com/wiki/' + n
    p[n] = [i, j, e, u]
```

惜字如金

```
liebiao = wenjianneirong.split('\n')
for yihang in liebiao:
    zidianliebiao = yihang.strip().split('\t')
if len(zidianliebiao) == 4:
    bianhao, mingcheng, riwen, yingwen = zidianliebiao
    wangzhi = 'https://wiki.52poke.com/wiki/' + mingcheng
    p[mingcheng] = [bianhao, riwen, yingwen, wanzhi]
```

拼音大神

```
# 按\n分割content为line_list
line_list = content.split('\n')
# 遍历line_list中的每一行
for line in line_list:
    # 按\t分割line为field_list
    field_list = line.strip().split('\t')
# 如果field_list有四个字段
if len(field_list) == 4:
    # 解包field_list到具体的变量
    id, name, japan_name, english_name = field_list
    # 拼装网址url
    url = 'https://wiki.52poke.com/wiki/' + name
    # 存入poke_dict, key为name, value为所有字段
    poke_dict[name] = [id, japan_name, english_name, url]
```

注释不花钱

代码可读性：什么是好代码？

- 好代码的关键词
 - 高内聚、低耦合、可读性、可测试、可复用等
- 我的理解
 - 目标：可维护、可复用
 - 可读、可理解是先决条件
 - 手段：高内聚、低耦合、测试、继承、封装等
- 我们该如何做？
 - 写烂代码
 - 多吐槽、改进自己的代码
 - 遵循规范
 - Python风格指南
 - https://zh-google-styleguide.readthedocs.io/en/latest/google-python-styleguide/python_style_rules/#
 - 参考优秀代码
 - GitHub
 - <https://github.com/>





函数：一种代码收纳方式

用过的函数：open

```
file = open('poke.txt', 'r', encoding='utf-8')
```

用函数改写FindPoke

```
poke_dict = read_poke_from_data('poke.txt')
poke_name = get_poke_name(sys)
if poke_name:
    show_poke(poke_dict, poke_name)
else:
    show_error()
```

使用了函数

```
# 将宝可梦数据导入字典
line_list = content.split('\n')
for line in line_list:
    field_list = line.strip().split('\t')
    if len(field_list) == 4:
        id, name, japan_name, english_name = field_list
        url = 'https://wiki.52poke.com/wiki/' + name
        poke_dict[name] = [id, japan_name, english_name, url]
```

未使用函数



未尽事宜： CMD、输入结果

```
C:\Windows\system32\cmd.exe
Microsoft Windows [版本 10.0.18362.2951]
(c) 2019 Microsoft Corporation。保留所有权利。

C:\Users\family>cd C:\Users\family\Documents\FindPoke\stage1

C:\Users\family\Documents\FindPoke\stage1>python FindPoke.py 小火龙
编号: #004
名称: 小火龙
日文: Charmander
英文: https://wiki.52poke.com/wiki/小火龙

C:\Users\family\Documents\FindPoke\stage1>python FindPoke.py 皮卡丘
编号: #025
名称: 皮卡丘
日文: Pikachu
英文: https://wiki.52poke.com/wiki/皮卡丘

C:\Users\family\Documents\FindPoke\stage1>python FindPoke.py 妙蛙
没有找到宝可梦: 妙蛙
```



下一步：多写代码

练习网站

- lintcode.com
- leetcode.com

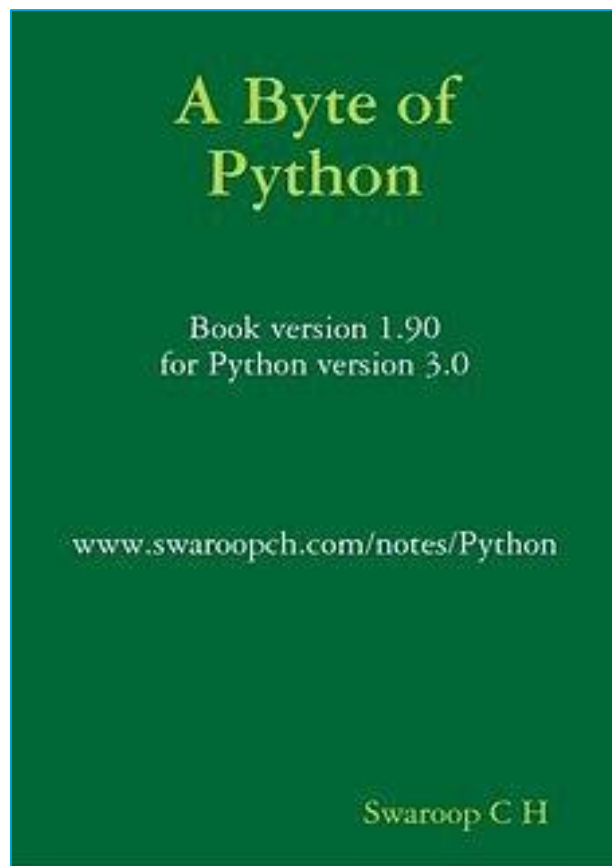
The screenshot shows the LintCode website interface. At the top, there's a navigation bar with 'LintCode', '主页', '算法', '人工智能', 'CAT', and 'VIP'. A search bar is present. Below the navigation bar, there's a filter section with various categories: '公司', '难度' (Difficulty), '提交状态' (Submission Status), '题目类型' (Problem Type), '算法' (Algorithms), and '数据结构' (Data Structures). The '难度' filter is expanded, and the '入门' (Beginner) option is highlighted with a red box. Below the filter section, a list of problems is shown, with the problem '145. 大小写转换' (145. Size Case Conversion) highlighted with a red box.

难度	题目	通过率
入门	37. 反转一个3位整数	63%
入门	145. 大小写转换	65%
入门	146. 大小写转换 II	45%
入门	214. 数组的最大值	49%
入门	222. Getter与Setter	62%
入门	225. 在链表中找节点	55%
入门	228. 链表的中点	47%

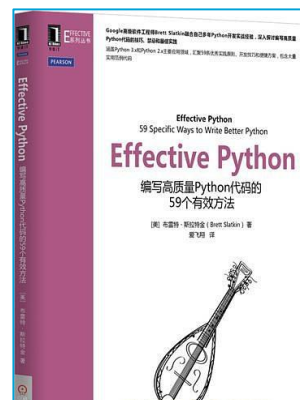
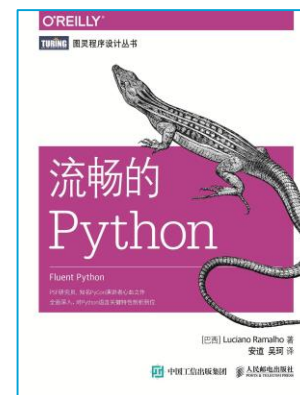
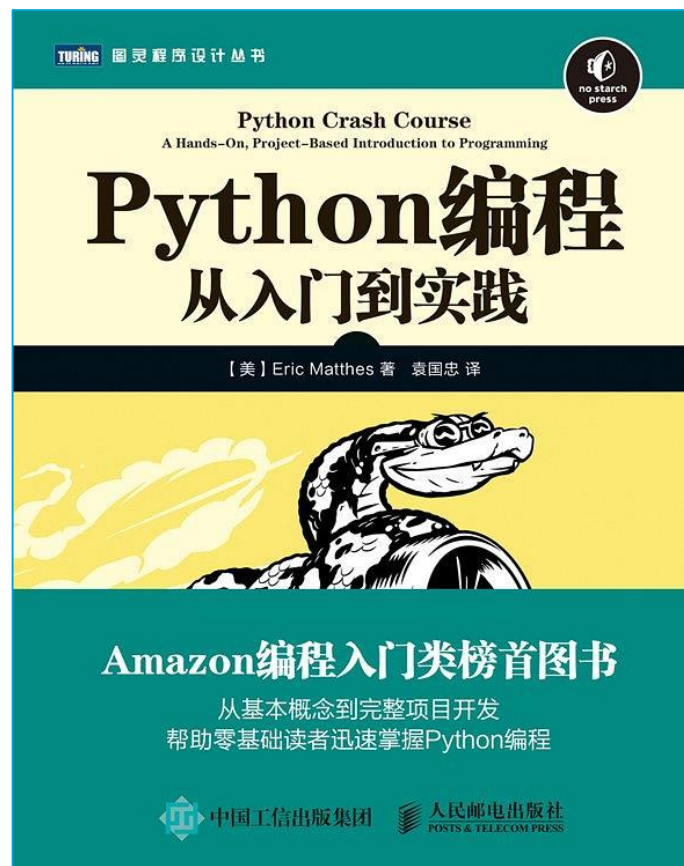
The screenshot shows the LintCode problem page for '145. 大小写转换' (145. Size Case Conversion). The page is in Chinese. The problem description is: '将一个字符由小写字母转换为大写字母' (Convert a character from lowercase to uppercase). The examples are: '输入: 'a', 输出: 'A' and '输入: 'b', 输出: 'B'. The code editor shows a Python3 solution template. The 'Python3' language is selected in the bottom left corner, highlighted with a red box. The bottom right corner has buttons for '我的历史提交' (My History Submissions), '运行测试数据' (Run Test Data), and '提交' (Submit).

```
1 class Solution:
2     """
3     @param character: a character
4     @return: a character
5     """
6     def lowercaseToUppercase(self, character):
7         # write your code here
8
```

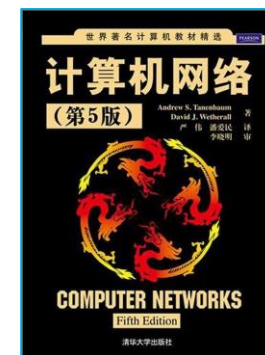
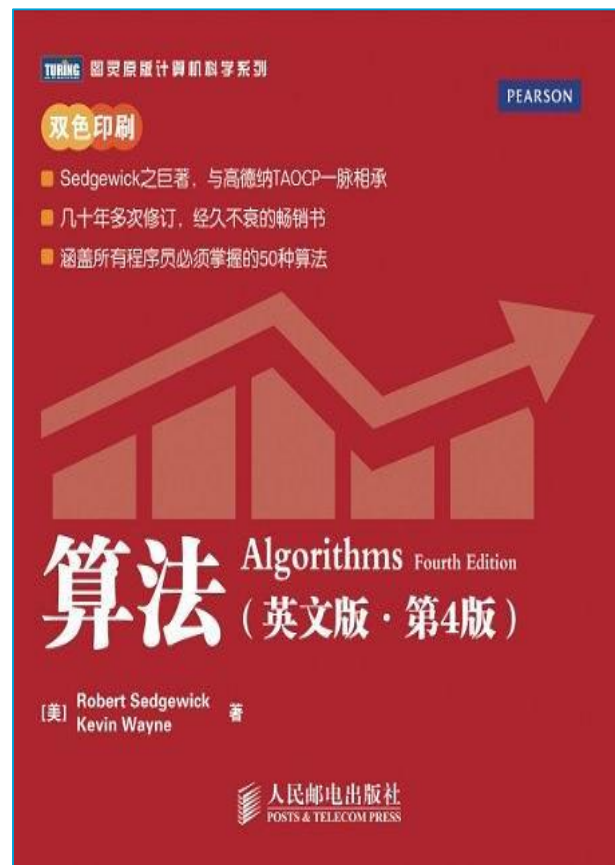
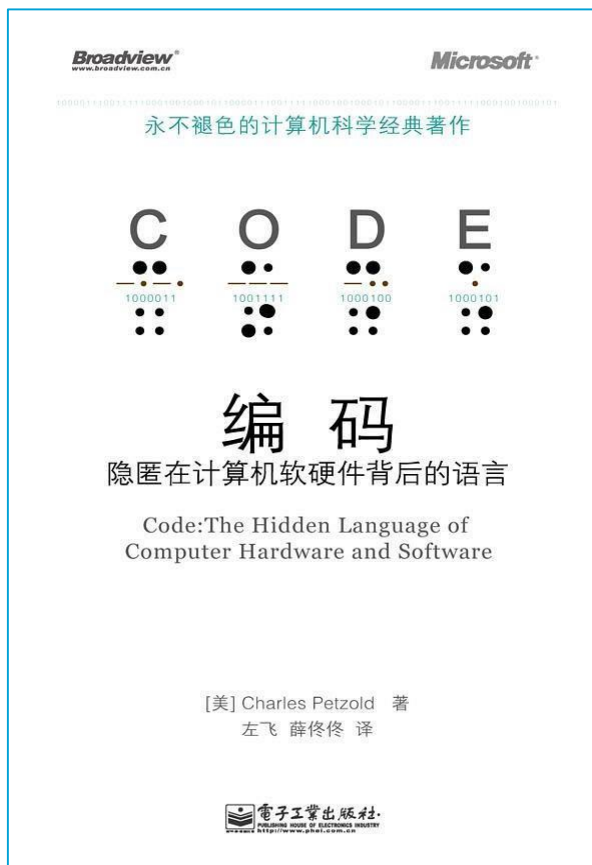
下一步： 补齐Python基础



中文: <https://bop.mol.uno/>
英文: <https://python.swaroopch.com/>



下一步： 补齐计算机基础



高等数学

线性代数

概率论

