

python语言入门与实践

变量与数据类型

主讲人：吴陈炜 博士

复旦大学智能机器人研究院 工程博士

北京大学信息科学与技术学院 理学硕士

浙江大学信息与电子工程学系 工学学士

网易（杭州）网络有限公司 项目经理

美国项目管理协会 PMP（项目管理专业人士）

之江实验室 科研主管

2020 年 5 月 7 日

上节回顾

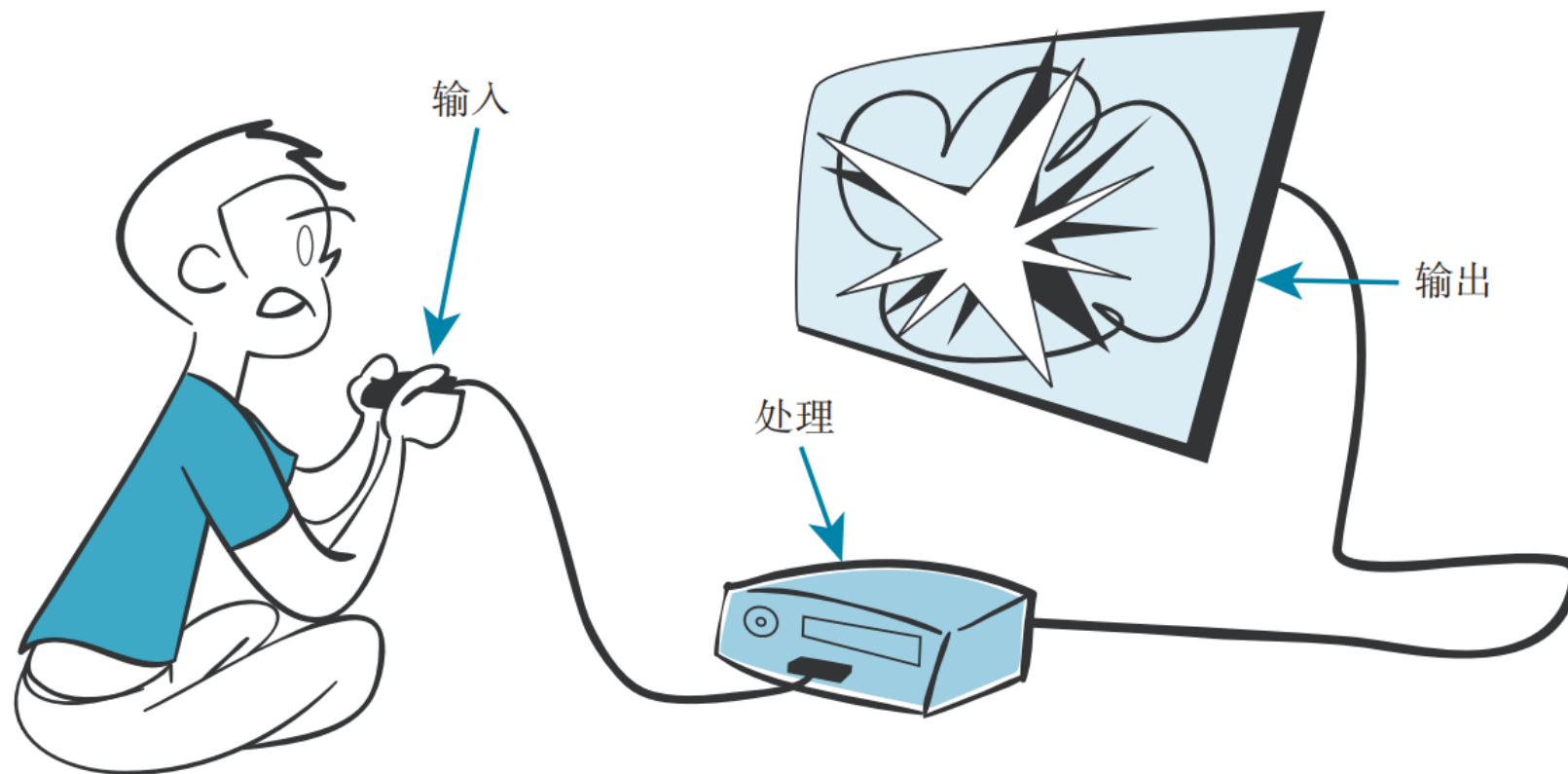
- 操作系统基础知识
- 使用import关键字调用python模块
- pip指令的使用
- 使用you-get下载视频

今日知识要点

- 变量的概念和使用
- python中的数据类型
- 字符串数据类型
- 数值型数据数据类型
- 简单的函数编写

变量与数据类型

程序的本质是数据变化的过程



经典的“输入 (input) 输出 (output)”模型，本质上就是数据流的变化

编程的“逻辑思维”

首先“把一件事分解到最细小的环节，然后按时间顺序或因果关联，组装成一条严密的逻辑链”。然后使用抽象出来的各个符号（变量），以及程序语言的各种语法（循环、判断），把这些逻辑链完整的描述清楚，得到的就是一个完整的计算机程序。

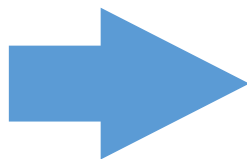
变量：数据的标签

- 世界是数字构成的，计算机在内存（Memory）中存放数据。
- 变量帮助人类标记（记忆）这些数据。
- 这就如同音乐家用音名和唱名来表示音调（频率）的高低一样。

| C调各音符频率对照表 | | |
|------------|-----|------|
| 音符 | 频率 | 周期 |
| 中音 1DO | 523 | 1912 |
| 中音 2RE | 587 | 1703 |
| 中音 3MI | 659 | 1517 |

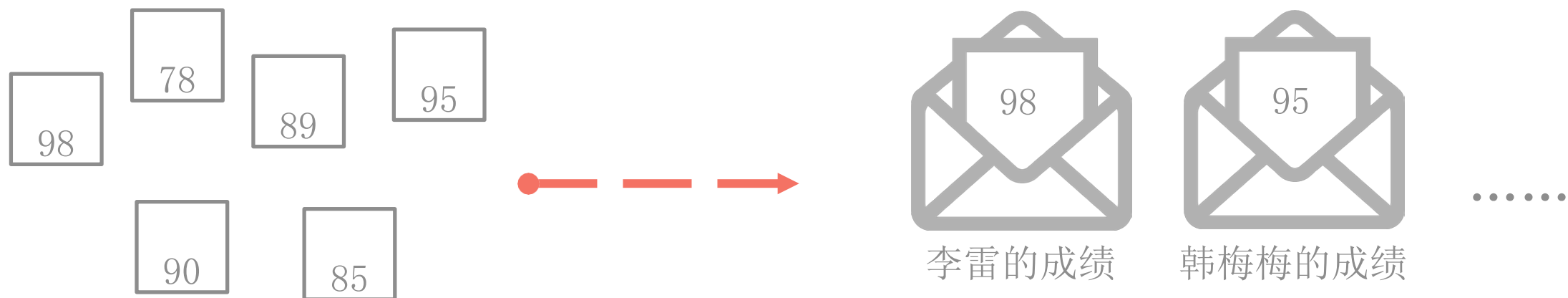
1 2 3-

转换为程序



```
1  import winsound
2  do = 523
3  re = 587
4  mi = 659
5  quad=1000 #1/4音符为一拍
6  half=2000
7  eigh=500
8  winsound.Beep(do,quad)
9  winsound.Beep(re,half)
10 winsound.Beep(mi,eigh)
```

理解变量——生活中的例子



$$\text{班级平均成绩} = \frac{(\text{李雷的成绩} + \text{韩梅梅的成绩} + \dots)}{\text{班级人数}}$$

如何创建变量——赋值语句

变量：通过变量名代表或引用某个值。

初始化赋值语句： 变量名 = 表达式

赋值运算符

示例：

- $x = 3$
- 李雷的成绩 = 98

变量赋值过程

```
x = 1  
x = x + 10
```

```
name1 = 'lilei'  
name2 = name1
```

```
name1 = 'lilei'  
name1 = 'hanmeimei'
```

进阶的赋值方法

同时给多个变量赋予同一个内容:

$a = b = c = 100$

同时给多个变量赋予不同的内容:

$a, b, c = 1, 2, 3$

变量的命名规则

- 起个好名字是编程成功的一半
- 尽可能使用有意义的文字给变量命名，比如 `power_level` , `tax_rate` 等。
- Python中变量名可以包含字母、下划线、数字等，但是不能以数字开头；
- 变量名不能与Python语法中规定的保留字相同，比如`import`等高亮字体；
- Python是大小写敏感语言，比如 `abc` 和 `aBc`是两个不同的变量；
- 不要使用`l/O/`等容易引起混淆的字母，尽管他们是合法的；
- 更多规则参考PEP（Python Enhancement Proposals）标准规范。

Python中的保留字

| | | | | |
|----------|---------|--------|--------|----------|
| and | elif | import | raise | global |
| as | else | in | return | nonlocal |
| assert | except | is | try | True |
| break | finally | lambda | while | False |
| class | for | not | with | None |
| continue | from | or | yield | |
| def | if | pass | del | |

(多选) 哪些可以做变量名:

A.as

B.As

C._as

D.01as

(多选) 哪些可以做变量名:

A.as

B.As



C._as



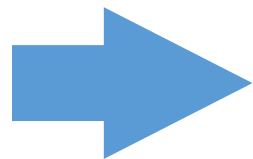
D.01as

字符串变量的使用

创建字符串：引号的使用

- 在python中我，们使用单引号、双引号、三引号来创建字符串
- 对于普通字符串，单引号就够了
- 如果我们的字符串中也包含引号，则使用双引号比较方便
- 我们也使用三引号来进行多行字符串的引用和注释

```
1 str_sgq = 'This Dr.Wu.'  
2 str_dbq = "I'm Dr.Wu."  
3 str_trq = '''Helle! everyone.  
4 I'm Dr.Wu  
5 '''  
6 print(str_sgq)  
7 print(str_dbq)  
8 print(str_trq)
```



```
This Dr.Wu.  
I'm Dr.Wu.  
Helle! everyone.  
I'm Dr.Wu  
  
[Finished in 0.1s]
```

Python中的转义字符 (escape)

Python和其他很多程序语言一样，定了许多转义字符，这些字符用反斜杠（\）开头，在字符串中表示一些特殊的字符，比如引号。Python中的转移字符如下表所示：

| 转义字符 | 描述 |
|----------|---------------|
| \\(在行尾时) | 续行符 |
| \\ | 反斜杠符号 |
| \' | 单引号 |
| \" | 双引号 |
| \a | 响铃 |
| \b | 退格(Backspace) |
| \e | 转义 |
| \000 | 空 |

| | |
|--------|---------------------------|
| \n | 换行 |
| \v | 纵向制表符 |
| \t | 横向制表符 |
| \r | 回车 |
| \f | 换页 |
| \oyy | 八进制数，yy代表的字符，例如：\o12代表换行 |
| \xyy | 十六进制数，yy代表的字符，例如：\x0a代表换行 |
| \other | 其它的字符以普通格式输出 |

SyntaxError:语法错误

- Syntax（语法）在编程语言是指一个句子中单词的排列和顺序
- 切记：字符串要放在成对的引号中如果出现了没有成对的情况，python就会报错，其中的EOL是End-Of-Line:

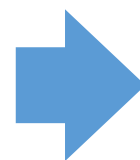
- `SyntaxError: EOL while scanning string literal`

- 解决方法就是把确实的引号补在合适的位置上
- 通俗的讲，也就是字符串的引号必须“成双成对出现”

在字符串中引用变量

- 如果字符串中需要包含变量，可以使用以下两种方法来实现。
 1. %方法：使用%s在字符串中占位，在字符串结尾用%引用；
 2. Fomat方法：使用花括号占位，在结尾用.format引用；
 3. f字符串方法。引号前加上字母f并使用花括号来引用变量。

```
name = "Allen"  
emotion = "happy"  
print("%s is %s" % (name,emotion))  
print("{} is {}".format(name,emotion))  
print(f"{name} is {emotion}.")
```

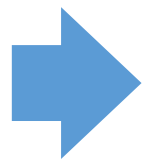


```
Allen is happy  
Allen is happy  
Allen is happy.  
[Finished in 0.1s]
```

字符串的乘法：重复

- 字符串的乘法（*），一般是字符串*整数，表示对该字符串进行重复操作，并返回重复之后的字符串。

```
string1='*'  
string2=string1*2  
print(string1)  
print(string2)
```



```
*  
**  
[Finished in 0.1s]
```

字符串的加法：拼接

- 我们可以使用+来拼接两个字符串，并且返回拼接之后的。我们来使用这种方法，配合字符串的乘法打印一条分割线。

```
16 string1='*'
17 string2=string1*20+'我是分割线'+string1*20
18 print(string1)
19 print(string2)
20 print(string1)
```



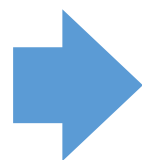
```
*
*****我是分割线*****
*
[Finished in 0.1s]
```

字符串的大小写转换

■ Python为字符串提供了三种方法来转换字符串中字母的大小写：

1. upper()方法：将字符串全部转换为大写字母
2. lower()方法：将字符串全部转换为小写字母
3. title()方法：将字符串转换成称呼模式

```
22 name = 'koBe bryant'  
23 print(name.upper())  
24 print(name.lower())  
25 print(name.title())
```



```
KOBE BRYANT  
kobe bryant  
Kobe Bryant  
[Finished in 0.1s]
```

更多字符串的用法

- **dir()函数**

用dir()函数来查看一下指定模块的功能列表。

- **help() 函数**

用于查看函数或模块用途的详细说明。

```
>>> dir(str)
['_add_', '__class__', '__contains__', '__delattr__', '__dir__', '__doc__', '__eq__', '__format__', '__ge__', '__getattribute__', '__getitem__', '__getnewargs__', '__gt__', '__hash__', '__init__', '__init_subclass__', '__iter__', '__le__', '__len__', '__lt__', '__mod__', '__mul__', '__ne__', '__new__', '__reduce__', '__reduce_ex__', '__repr__', '__rmod__', '__rmul__', '__setattr__', '__sizeof__', '__str__', '__subclasshook__', 'capitalize', 'casefold', 'center', 'count', 'encode', 'endswith', 'expandtabs', 'find', 'format', 'format_map', 'index', 'isalnum', 'isalpha', 'isdecimal', 'isdigit', 'isidentifier', 'islower', 'isnumeric', 'isprintable', 'isspace', 'istitle', 'isupper', 'join', 'ljust', 'lower', 'lstrip', 'maketrans', 'partition', 'replace', 'rfind', 'rindex', 'rjust', 'rpartition', 'rsplit', 'rstrip', 'split', 'splitlines', 'startswith', 'strip', 'swapcase', 'title', 'translate', 'upper', 'zfill']
>>> help(str.replace)
Help on method_descriptor:

replace(...)
    S.replace(old, new[, count]) -> str

    Return a copy of S with all occurrences of substring
    old replaced by new.  If the optional argument count is
    given, only the first count occurrences are replaced.

>>>
```


使用字符串传递命令

知识点

- Import os导入系统os库
- 变量的定义和使用
- 使用字符串拼接命令
- 使用os.system()向操作系统

发送命令（相当于在cmd

窗口中进行命令操作）

- you-get命令的写法

注：也可以直接在cmd中输入该命令

```
1 import os
2 url = 'https://www.bilibili.com/video/av1415480'
3 command = 'you-get --playlist -o F://bilibili/jaychou '+url
4 os.system(command)
5
```

```
site: Bilibili
title: 【经典】周杰伦MV 【185P】 (P1. 可爱女人【1st JAY】)
stream:
  - format: flv
    container: flv
    quality: 高清 1080P
    size: 17.4 MiB (18284207 bytes)
    # download-with: you-get --format=flv [URL]
```

Downloading 【经典】周杰伦MV 【185P】 (P1. 可爱女人【1st JAY】).flv ...

| | | | |
|----------------------|--|-------|----------|
| 0.0% (0.0/ 17.4MB) | | [1/1] | |
| 1.4% (0.2/ 17.4MB) | | [1/1] | 1 MB/s |
| 2.9% (0.5/ 17.4MB) | | [1/1] | 5 MB/s |
| 4.3% (0.8/ 17.4MB) | | [1/1] | 661 kB/s |
| 5.7% (1.0/ 17.4MB) | | [1/1] | 198 kB/s |
| 7.2% (1.2/ 17.4MB) | | [1/1] | 213 kB/s |
| 8.6% (1.5/ 17.4MB) | | [1/1] | 242 kB/s |
| 10.0% (1.8/ 17.4MB) | | [1/1] | 238 kB/s |
| 11.5% (2.0/ 17.4MB) | | [1/1] | 249 kB/s |

python数据类型总览

Python中常用的数据类型

以下这些类型是python编程基础中的基础，请务必识记：

- 布尔型 (bool)
- 整数型 (int)
- 浮点型 (float)
- 复数型 (complex)
- 字符串 (str)
- 列表型 (list)
- 元组型 (tuple)
- 字典型 (dict)
- 集合型 (set)



数据类型

Data Type

python3 常用数据类型

数值型

布尔型 (bool)

数字型

整数型(int)

浮点型(float)

(复数型(complex))

容器型

有序序列

字符串(str)

列表(list)

元组(tuple)

无序序列

字典(dict)

集合(set)

数据类型

Data Type

布尔型

```
a = True
b = False
print(type(a))
print(type(b))
```

```
<class 'bool'>
<class 'bool'>
```

数据类型

Data Type

数字型

```
a = 1  
print(type(a))
```

```
b = 2.0  
print(type(b))
```

```
<class 'int'>  
<class 'float'>
```

数据类型

Data Type

字符串

```
a = 'Hello Makerbean'  
print(type(a))
```

```
<class 'str'>
```

数据类型

Data Type

列表

```
a = [1, 'two', 3.0, 'four']  
print(type(a))
```

```
<class 'list'>
```


数据类型

Data Type

元组

```
a = (1, 'two', 3.0, 'four')  
print(type(a))
```

```
<class 'tuple'>
```

数据类型

Data Type

字典

```
a = {1: 'one', 2: 'two'}  
print(type(a))
```

```
<class 'dict'>
```

数据类型

Data Type

集合

```
a = set([1,2,2,3,2,6,2,1,13,5])  
print(type(a))
```

```
<class 'set'>
```

数据类型

Data Type

下列不是数据类型的是：

A.字符串

B.数字

C.数组

D.元组

答案

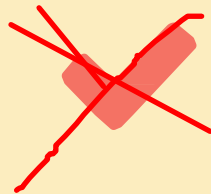
下列不是数据类型的是：

A.字符串

B.数字

C.数组

D.元组



数据类型

Data Type

以下哪一个数据类型是无序序列：

A.字典

B.字符串

C.列表

D.元组

答案

以下哪一个数据类型是无序序列：

A.字典



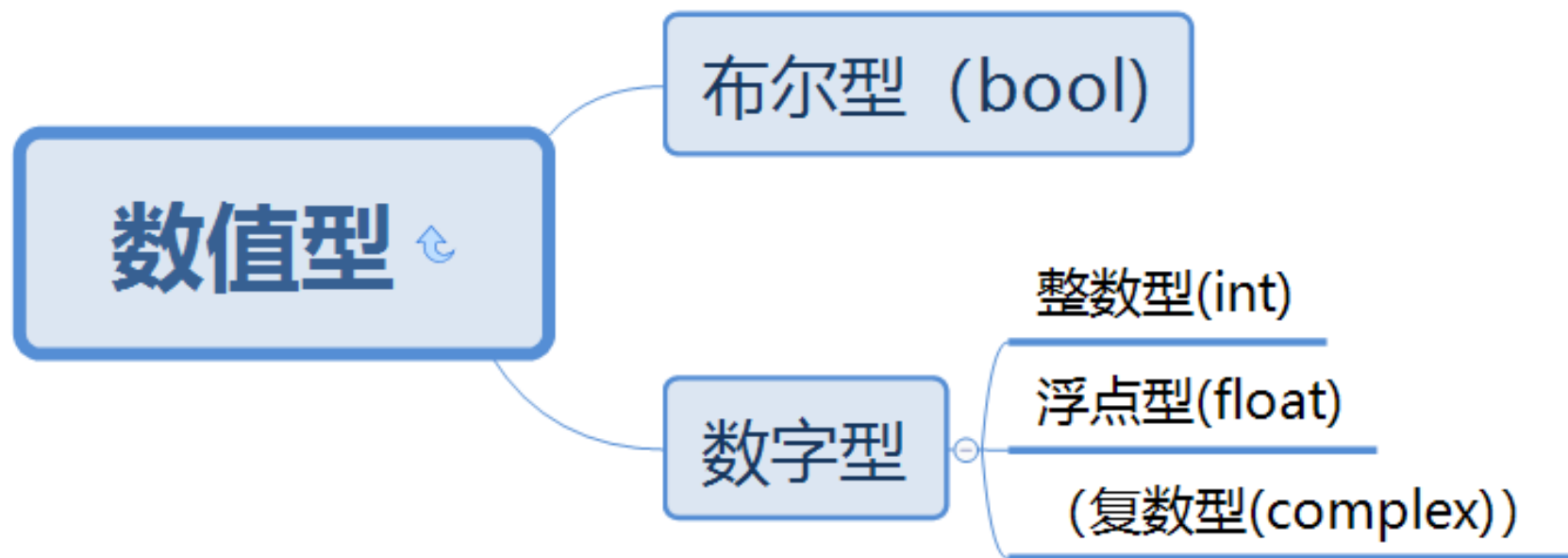
B.字符串

C.列表

D.元组

数值型数据

数值型数据分类



布尔值

意义：表示判断中的是与否。
一般用于条件测试中。

```
>>> a = True
>>> print(a)
True
```

```
>>> 10<5
False
>>> 10>8
True
```

bool { 是: True

否: False

- None
- 数值中的零，包括0，0.0
- 空序列，包括空字符串("")，空元组(()，空列表([])
- 空的字典{}

逻辑运算

逻辑运算符：用于检测两个或两个以上的条件是否满足。

逻辑运算只存在与布尔类型中。

| 运算符 | | 描述 |
|------|-------|--|
| and, | 逻辑“与” | 当运算符两边的两个运算对象都为true时，结果为true |
| or, | 逻辑“或” | 当运算符两边的两个运算对象其中有一个运算对象为true时， 结果 即为true |
| not, | 逻辑“非” | 用于反转运算对象的状态 |

布尔表达式

- 3 and 5 True and True True 5
- 3 or 5 True or True True 3
- 0 or 5 False or True True 5
- 3 and not 5 True and False False False

表达式的应用——条件测试

- 检查当前变量是否与一个特定值相等/不相等。
- 比较数字的大小。
- 检查特定值是否在某序列里。

表达式的应用——多条件检查

- 使用`and`检查多个条件

```
>>> age_lilei = 17
>>> age_hanmeimei = 18
>>> age_lilei >= 18 and age_hanmeimei >= 18
False
>>> age_lilei >= 15 and age_hanmeimei >= 15
True
```

- 使用`or`检查多个条件

```
>>> age_lilei >= 18 or age_hanmeimei >= 15
True
>>> age_lilei >= 20 or age_hanmeimei >= 20
False
```

Test

(多选) 以下哪些值可以被当作布尔值中的False?

- A. 0
- B. None
- C. 空序列
- D. 空字典

Answer

(多选) 以下哪些值可以被当作布尔值中的False?

A. 0



B. None



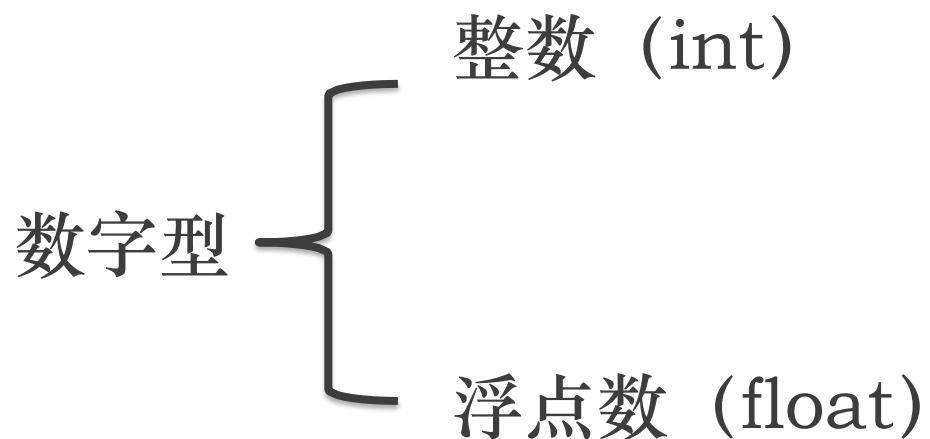
C. 空序列



D. 空字典



数字型



```
>>> num1 = 2
>>> num2 = 7.0
>>> print(num2/num1)
3.5
>>> print(num2*num1)
14.0
>>> print(num2+num1)
9.0
>>> print(num2-num1)
5.0
```

数字型-运算

算术运算符：用于算术计算。

| 运算符 | 描述 |
|-----|---------------------|
| + | 加法运算符 |
| - | 减法运算符 |
| * | 乘法运算符 |
| / | 除法运算符 |
| ** | 指数运算符 |
| % | 取余运算符，计算余数 |
| // | 除法取整运算符，计算商并去除其小数部分 |

```
>>> x = 9
>>> y = 2
>>> print(x**y)
81
>>> print(x%y)
1
>>> print(x//y)
4
```

数字型-运算

比较运算符：比较值的大小。

| 运算符 | 描述 |
|-----|-------------------------|
| > | 判断第一个运算对象是否大于第二个运算对象 |
| < | 判断第一个运算对象是否小于第二个运算对象 |
| >= | 判断第一个运算对象是否大于或等于第二个运算对象 |
| <= | 判断第一个运算对象是否小于或等于第二个运算对象 |
| == | 判断两个运算对象是否相同 |
| != | 判断两个运算对象是否不相同 |

```
>>> x = 9
>>> y = 2
>>> if(x==y):
...     print('yes')
... else:
...     print('no')
...
no
```

```
>>> if(x = y):
    File "<stdin>", line 1
        if(x = y):
            ^
SyntaxError: invalid syntax
```

数字型-运算

赋值运算符：把运算符右侧的值赋给左侧的数据。

| 运算符 | 描述 |
|-----|----------------------------|
| = | 把右侧的运算对象赋值给左侧的运算对象 |
| += | $a += b$ 等同于 $a = a + b$ |
| -= | $a -= b$ 等同于 $a = a - b$ |
| *= | $a *= b$ 等同于 $a = a * b$ |
| /= | $a /= b$ 等同于 $a = a / b$ |
| **= | $a **= b$ 等同于 $a = a ** b$ |
| //= | $a //= b$ 等同于 $a = a // b$ |

数字型-运算

```
x = 4.5  
y = 2  
print(x//y)
```

下列哪一项会被打印出来?

A.2.0

B.2.25

C.0.25

D.0.5

答案

```
x = 4.5  
y = 2  
print(x//y)
```

下列哪一项会被打印出来?

A.2.0



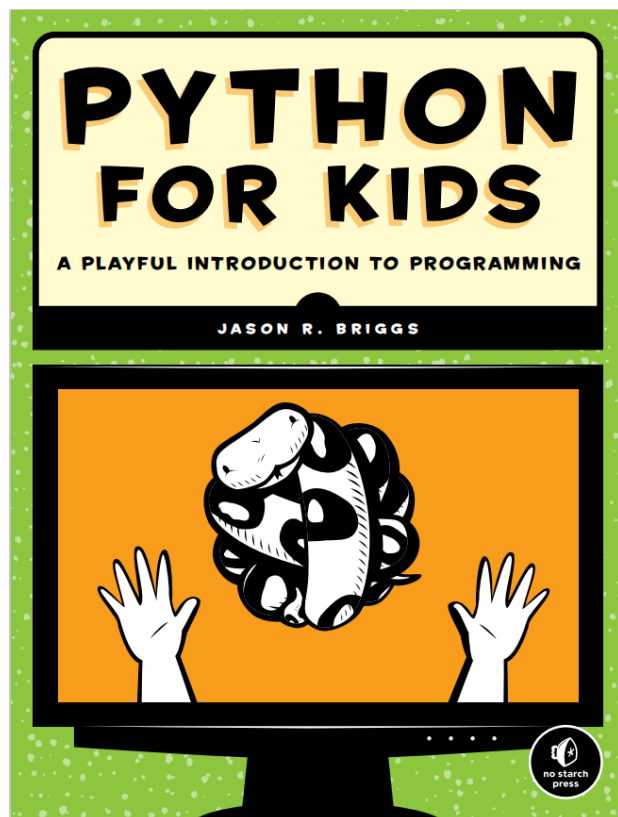
B.2.25

C.0.25

D.0.5

拓展阅读

■ Python for kid 第2章、第3章字符串部分

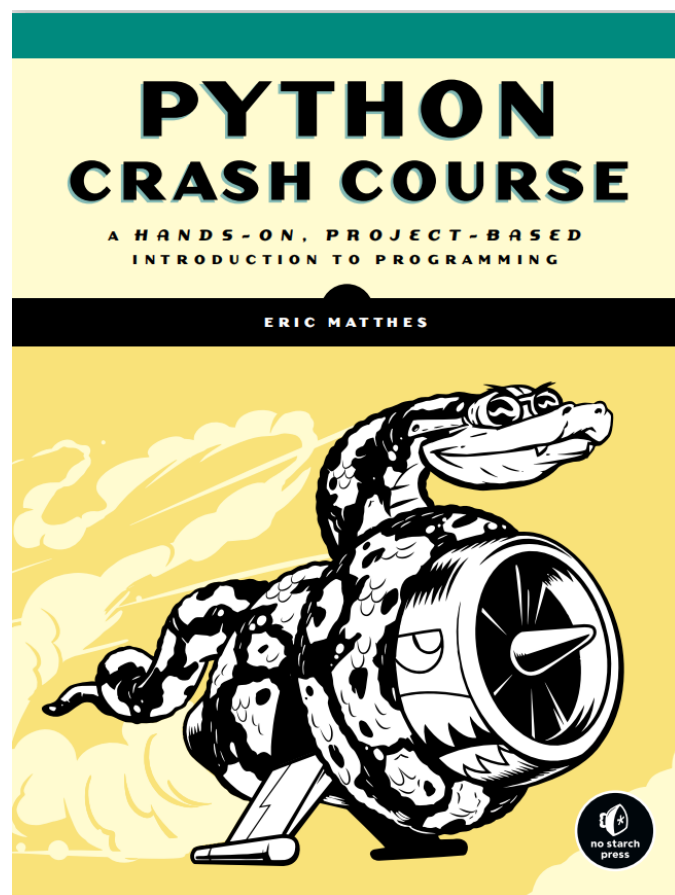


PART I: LEARNING TO PROGRAM

| | |
|---|-----|
| Chapter 1: Not All Snakes Slither | 3 |
| Chapter 2: Calculations and Variables | 15 |
| Chapter 3: Strings, Lists, Tuples, and Maps | 25 |
| Chapter 4: Drawing with Turtles | 43 |
| Chapter 5: Asking Questions with if and else | 53 |
| Chapter 6: Going Loopy | 67 |
| Chapter 7: Recycling Your Code with Functions and Modules | 81 |
| Chapter 8: How to Use Classes and Objects | 93 |
| Chapter 9: Python's Built-in Functions | 109 |

拓展阅读

■ Python Crash Course 第2章



| | |
|--|----------|
| PART I: BASICS | 1 |
| Chapter 1: Getting Started | 3 |
| Chapter 2: Variables and Simple Data Types | 19 |
| Chapter 3: Introducing Lists | 37 |
| Chapter 4: Working with Lists | 53 |
| Chapter 5: if Statements | 75 |
| Chapter 6: Dictionaries | 95 |
| Chapter 7: User Input and while Loops | 117 |
| Chapter 8: Functions | 133 |
| Chapter 9: Classes | 161 |
| Chapter 10: Files and Exceptions | 189 |
| Chapter 11: Testing Your Code | 215 |

作业

■ 完成Python Crash Course 第二章中的习题 (Exercises)

| | | |
|--|-----------|--|
| 2 | | |
| VARIABLES AND SIMPLE DATA TYPES | 19 | |
| What Really Happens When You Run <code>hello_world.py</code> | 19 | |
| Variables | 20 | |
| Naming and Using Variables | 21 | |
| Avoiding Name Errors When Using Variables | 21 | |
| <i>Exercise 2-1: Simple Message</i> | 23 | |
| <i>Exercise 2-2: Simple Messages</i> | 23 | |
| Strings | 23 | |
| Changing Case in a String with Methods | 24 | |
| Combining or Concatenating Strings | 25 | |
| Adding Whitespace to Strings with Tabs or Newlines | 26 | |
| Stripping Whitespace | 26 | |
| Avoiding Syntax Errors with Strings | 28 | |
| Printing in Python 2 | 29 | |
| <i>Exercise 2-3: Personal Message</i> | 29 | |
| <i>Exercise 2-4: Name Cases</i> | 29 | |
| <i>Exercise 2-5: Famous Quote</i> | 29 | |
| <i>Exercise 2-6: Famous Quote 2</i> | 29 | |
| <i>Exercise 2-7: Stripping Names</i> | 29 | |
| Numbers | 30 | |
| Integers | 30 | |
| Floats | 30 | |
| Avoiding Type Errors with the <code>str()</code> Function | 31 | |
| Integers in Python 2 | 32 | |
| <i>Exercise 2-8: Number Eight</i> | 33 | |
| <i>Exercise 2-9: Favorite Number</i> | 33 | |
| Comments | 33 | |
| How Do You Write Comments? | 33 | |
| What Kind of Comments Should You Write? | 33 | |
| <i>Exercise 2-10: Adding Comments</i> | 34 | |
| The Zen of Python | 34 | |
| <i>Exercise 2-11: Zen of Python</i> | 36 | |
| Summary | 36 | |

思考题

- 根据王者荣耀伤害结算公式，制作一个普攻伤害计算函数，并且计算李白在物攻834、物理穿透300|45%的状态下普攻物抗为1588的项羽时造成的伤害。并且打印伤害提示“李白对项羽造成了XX点伤害”

- 参考文献：

<https://zhuanlan.zhihu.com/p/68485973>

