

基础知识

王肖燕





➔ 内容介绍

一. 数据类型

- ✓ 整型、浮点型、布尔型、字符串

二. 类型转换

三. 常用运算符

- ✓ 算术、关系、逻辑、优先级

四. 基本应用

➡ 数据类型是什么？

- “1314”和1314一样吗？
 - 带双引号是字符串，不带引号的，是数字
- In [1]: "1314" + "520"
 - 结果？
- In [2]: 1314+520
 - 结果？

➡ 类型

- 解决问题之前，需要搞清楚使用什么**类型**的数据？
- **类型**是编程语言对数据的一种划分
- 课程涉及的类型：
 - ✓ **基本数据类型**
 - ✓ 数字类型、字符串类型
 - ✓ **组合数据类型**
 - ✓ 元组类型、列表类型、集合类型、字典类型（后续讲）

➔ 数字类型

- 整型

- 1, 123, 0, -2, -111
- 长度不受限制, 易进行大数计算

- 浮点型——小数

- 1.2, 0.1234, 0.015, 1.5e-2

- 布尔型

- True、False (注意首字母大写)
- 特殊的整型: True (1)、False (0)



运算符

1. 算术运算符

① **

② *, /, %, //

③ +, -

2. 字符串连接

✓ +

3. 关系运算符

✓ <, <=, >, >=, ==, !=

4. 逻辑运算符

✓ and, or, not

```
In [1]: 2**3  
Out[1]: 8
```

```
In [2]: 5%3  
Out[2]: 2
```

```
In [3]: 5//3  
Out[3]: 1
```

```
In [4]: "abc"+"ABC"  
Out[4]: 'abcABC'
```

```
In [5]: 1<2  
Out[5]: True
```

```
In [6]: 1<=2  
Out[6]: True
```

```
In [7]: 1==2  
Out[7]: False
```

```
In [8]: 1!=2  
Out[8]: True
```

```
In [9]: 1<2 and 0<1  
Out[9]: True
```

```
In [10]: 1<2 and 2>3  
Out[10]: False
```

```
In [11]: 1<2 or 2>3  
Out[11]: True
```

```
In [12]: not 1<2  
Out[12]: False
```

➡ 关系运算符

- 与其它程序设计语言不同
- Python关系运算符最大的特点是可以连用
- In [1]: $1 < 3 < 5$
- Out[1]: True
- In [2]: $3 < 5 > 2$
- Out[2]: True
- In [3]: $1 > 6 < 8$
- Out[3]: False
- In [4]: 'Hello' > 'world'
- Out[4]: False
- 操作数之间必须可比较大小，即必须有意义

➡ 变量

- 存放上述类型的数据的容器
- `x = 3`
 - 创建了一个整型变量x里面存放的值为3
- Python不需要事先声明变量名及其类型，直接赋值即可创建各种类型的对象变量
- 变量类型可以随时发生改变
- `x = 3`
- `x = 1.5`
- 解释器根据赋值或运算来自动推断变量类型

➡ 变量命名规则

- 变量名由数字、字母或下划线组成，必须以字母或下划线开头
- 区分大小写：student与Student不同
- 下面合法的变量名有：
abc、3abc、_3abc、__3abc、a,bc、a.b
Abc、b、Z、a_b

➡ 思考优先级

1. $30 - 3^{**}2 + 9 // 3^{**}2 * 10 = ?$

2. $3 * 4^{**}2 / 8 \% 5 = ?$

3. $2^{**}2^{**}3 = ?$

➤ 搞不清楚优先级时，怎么办？

➡ 思考优先级

1. $30 - 3^{**}2 + 9 // 3^{**}2 * 10 = ?$

2. $3 * 4^{**}2 / 8 \% 5 = ?$

3. $2^{**}2^{**}3 = ?$

➤ 搞不清楚优先级时，怎么办？

加括号！



字符串类型

- 文本

- 单引号或双引号内的everything
- 与数字截然不同，比如：相加
- 特殊情况：

➤ 字符串类型-特殊情况

- 字符串内容中需要出现单引号或双引号：打印'Let's go!'
 - ① 使用转义符号 (\) 对字符串中的引号进行转义：`print('Let\'s go!')`
 - ② 单引号和双引号相结合：`print("Let's go!")`
- 原始字符串
 - ① 转义字符\：打印"C:\now"→`print("C:\\now")`
 - ② 字符串前加r：`print(r"C:\now")`
- 长字符串
 - ✓ 一篇长文，包括格式，如何打印？

➔ 字符串操作-访问字符

- 字符串是一个字符序列：字符串最左端位置标记为0，依次增加。**字符串**中的编号叫做“**索引**”

H	e	l	l	o		J	o	h	n
0	1	2	3	4	5	6	7	8	9

- 单个索引辅助访问字符串中的特定位置
 - 格式为<string>[<索引>]

```
In [1]: s="Hello world!"
In [2]: s[0]
Out[2]: 'H'
In [3]: s[5]
Out[3]: '.'
In [4]: s[6]
Out[4]: 'w'
```

➡ 字符串操作-访问字符

1. Python中字符串索引从0开始，一个长度为L的字符串最后一个字符的位置是L-1
2. Python同时允许使用负数从字符串右边末尾向左边进行反向索引，最右侧索引值是-1

```
In [1]: s="Hello world!"
```

```
In [2]: s[-1]  
Out[2]: '!'
```

```
In [3]: s[-2]  
Out[3]: 'd'
```

```
In [4]: s[-3]  
Out[4]: 'l'
```

```
In [5]: s[0]  
Out[5]: 'H'
```

```
In [6]: s[11]  
Out[6]: '!'
```

```
In [7]: s[-1]  
Out[7]: '!'
```

```
In [8]: s[-12]  
Out[8]: 'H'
```

➡ 字符串操作-访问子串

- 两个索引值确定一个位置范围，返回子串
 - 格式： <string>[<start>:<end>]
- start和end都是整数型数值
- 子序列从索引start开始到end-1结束
- 但不包括end位置。

```
In [9]: s
Out[9]: 'Hello world!'

In [10]: s[0:3]
Out[10]: 'Hel'

In [11]: s[2:6]
Out[11]: 'llo '
```


➡ 字符串操作-连接及长度

- 字符串之间可以通过+或*进行连接
- +：将两个字符串连接成为一个新的字符串
- *：生成一个由其本身字符串重复连接而成的字符串

```
In [12]: "egg"+"plant"  
Out[12]: 'eggplant'  
  
In [13]: 3*"idiot"  
Out[13]: 'idiotidiotidiot'
```

- len()函数返回一个字符串的长度

```
In [14]: len("egg")  
Out[14]: 3  
  
In [15]: len("plant")  
Out[15]: 5
```

➡ 类型转换

- `int()`: 将一个字符串或浮点数转换为整数
 - `int(5.1)`, `int(5.99)`
 - 注意: 是直接去掉小数部分
- `float()`: 将一个字符串或整数转换为小数
 - `float('123')`, `float(123)` 增加小数部分
- `str()`: 将一个数或任何其他类型转换成字符串
 - `str(1.234)`

➡ 输入输出函数

- `input()`函数

- 从交互命令行获得用户输入：
- 用户在命令行里输入什么内容，`input`函数都以字符串类型返回结果，结果可以赋值给一个变量

```
In [1]: input("请输入一个数: ")
```

```
请输入一个数: 5
```

```
Out[1]: '5'
```

```
In [2]: a=input("请输入一个数: ")
```

```
请输入一个数: 6
```

```
In [3]: a
```

```
Out[3]: '6'
```

- `print()`函数

- 输出信息：
- 变量与字符串混合的形式输出

```
In [4]: print("Hello!")  
Hello!
```

```
In [5]: a=7
```

```
In [6]: print(a)  
7
```

- 槽格式

```
In [1]: a=3;b=2
```

- `format()`

```
In [2]: print("a的值是%d,b的值是%d"%(a,b))  
a的值是3,b的值是2
```

```
In [3]: print("a的值是{0},b的值是{1}".format(a,b))  
a的值是3,b的值是2
```

➡ eval函数

- 能够以Python表达式的方式解析并执行字符串，并返回执行结果

```
— In [17]: a=3  
In [18]: eval("a+2.2")  
Out[18]: 5.2
```

- 如何希望通过input函数输入一个数字，并用程序对这个数字进行计算：

```
In [19]: a=eval(input("请输入一个数: "))  
  
请输入一个数: 6  
  
In [20]: a  
Out[20]: 6  
  
In [21]: type(a)  
Out[21]: int
```

➡ 字符串格式化方式1——百分号

格式：'%[对齐方式][加正号][指定空位填充][指定最小宽度].[指定精度]格式字

符'%待转换的表达式，其中

- %表示格式的开始，必选
- 对齐方式，+右对齐，-左对齐
- 加正号，对正数加正号
- 指定空位填充，可以用字符对不够宽度的空位进行填充
- 指定最小宽度，数据的宽度设定
- 指定精度，小数点后保留的位数

格式字符	描述
%d	十进制整数
%i	十进制整数
%o	八进制整数
%x	十六进制整数
%e	指数（基底为e）
%E	指数（基底为E）
%f,%F	浮点数
%g	指数（e）或浮点数
%G	指数（E）或浮点数
%s	字符串
%r	字符串
%c	单个字符
%%	字符%

例如：print("")

➡ 字符串格式化方式2——format

- 位置映射

```
In [1]: print("{}:{}".format("192.168.0.100", "8080"))  
192.168.0.100:8080
```

- 关键字映射

```
In [2]: print("{server}{0}:{1}".format('192.168.0.100', '8080', server='Web  
server Info:'))  
Web server Info:192.168.0.100:8080
```

- 元素访问

```
In [4]: print("{0}.{1}.{2}".format('www', 'baidu', 'com'))  
www.baidu.com
```

```
In [3]: print("{0[0]}.{1[0]}.{1[1]}".format(('www', 'sss'), ('baidu', 'com')))  
www.baidu.com
```

➡ 字符串格式化方式2——format

- 填充对齐

```
In [5]: print("{0}*{1}={2:0>2}".format(3,2,2*3))  
3*2=06
```

```
In [6]: print("{:*^30}".format("centered"))  
*****centered*****
```

- 精度设置

```
In [7]: print("{:.3f}".format(2.1415))  
2.142
```

```
In [8]: print("{:10f}".format(3.1415))  
3.141500
```

```
In [9]: print("{:<10f}".format(3.1415))  
3.141500
```



表达式练习

- 闰年判断
 - 能被400整除
 - 能被4整除，并且不能被100整除
 - 假设年存放在year变量中，如何写判断表达式？
- 判断一个人的年龄是否在20到28岁之间
 - 假设年龄存放在age变量中
 - 如何写判断表达式？

➡ 算术运算和字符串练习

用Python程序计算：

1. $x = \frac{2^4 + 7 - 3 \times 4}{5}$

2. $x = (1 + 3^2) \times (16 \bmod 7) \div 7$

3. 输入一个月份数字，返回对应月份名称缩写

➡ 返回月份简写

```
>>> m = int(input("输入月份: "))
```

```
>>> months = "JanFebMarAprMayJunJulAugSepOctNovDec"
```

- 如果pos表示一个月份的第一个字母
 - 则months[pos:pos+3]表示这个月份的缩写

"JanFebMarAprMayJunJulAugSepOctNovDec"

- | | | | | | | | | | | | | |
|---------|-------|---|---|---|----|----|----|----|----|----|----|----|
| • 索引: | 0 | 3 | 6 | 9 | 12 | 15 | 18 | 21 | 24 | 27 | 30 | 33 |
| • 月份 m: | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| • 规律总结: | pos=? | | | | | | | | | | | |

- 如何实现?

Thank You!

