

信息技术

上节内容

- 关键字，运算符和句式
- 变量，赋值，`print()`
- Replit编程

本节内容

- 变量类型
- 运算符
- 类型转换

Python数据类型

我们前四周需要掌握的4种数据类型：

- ◆ **整数 (int)**

- e.g. 732 or -5

- ◆ **浮点数 (float)**

- e.g. 3.14 or 2.0

- ◆ **字符串 (str)**

- e.g. 'hello' or 'C'

- ◆ **布尔值 (boolean)**

- True/False

字符串

```
name = "Python"  
print(name)  
message = "I love Python."  
print(message)
```

练习

请指出下列变量的类型：

◆ $a = 5$

◆ $t = 5.0$

◆ $b = "10.24"$

◆ $z = 3.14159$

◆ $x = 3.5 + 5$

Python是一门弱类型语言

弱类型语言：声明变量时不需要指明类型，变量类型可以随时变化

```
var = 1
```

```
var = "Hi There"
```

变量类型取决于最新的
赋值数据的类型

Java:

```
int var1 = 1;
```

```
String var2 = "Hi There";
```

Java是强类型语言：变量类型
需要在声明时确定。

而且一旦确定，“几乎”不
能更改

Python算术运算符

Op	Meaning	Example
+	Addition	9+2 is 11
		9.1+2.0 is 11.1
-	Subtraction	9-2 is 7
		9.1-2.0 is 7.1
*	Multiplication	9*2 is 18
		9.1*2.0 is 18.2
/	Division	9/2 is 4.5
		9.1/2.0 is 4.55
//	Integer Division	9//2 is 4
%	Remainder	9%2 is 1

The // operator

/ 和//运算的区别

- ◆ **/ operator**: 普通除法, 得到结果为浮点数 (即使可以整除)
- ◆ **// operator**: 向下取整除法, 返回两个操作数相除的结果, 但结果总是向下取最接近的整数。

Example:

9//4 is 2	(-1)//2 is -1	(-1)//(-2) is 0
1//2 is 0	1//(-2) is -1	10//2 is 5

The % operator

% operator: 取余符

- ◆ The remainder operator % returns the remainder of the result of dividing its first operand by its second.

Example:

9%4 is 1	1%2 is 1	2% 5 is 2
13%5 is 3	10%5 is 0	1%0.6 is 0.4

Python逻辑运算符

and	True	False
True	True	False
False	False	False

and: 同真即真

or	True	False
True	True	True
False	True	False

or: 同假即假

	True	False
not	False	True

not: 非真即假, 非假即真

数据类型

Python提供六种数据类型：

- 数字
- 字符串
- 列表
- 元组
- 集合
- 字典

数据类型

Python提供六种数据类型：

- 数字
- 字符串

- 列表
- 元组
- 集合
- 字典

数据结构



数字类型

- 整数 (int) : 5, -81, 19452
- 浮点数 (float) : 0.5, -5.2, 2016.2435
- 布尔数 (boolean): True, False
- 复数: 3+5j

字符串类型

- 用单引号或者双引号括起来，例如“Good morning”，‘hello world’
- 最好遵循统一的标准，不要将单引号和双引号混合使用
- 字符串常用来表示文本，是我们用到最多的数据类型之一

type(): 查看数据类型

```
main.py
1 print(type(12))
2
3 print(type(5.0))
4
5 print(type(True))
6
7 print(type(4+3j))
```

<class 'int'>
<class 'float'>
<class 'bool'>
<class 'complex'>

变量

- 用来保存数据，方便以后使用
- 变量必须有名字

变量

- 用来保存数据，方便以后使用
- 变量必须有名字

```
print("Good morning")
```

```
a = "Good morning"  
print(a)
```


Python是一门弱类型的语言

- 定义变量无须声明类型
- 变量类型可以随时改变

```
main.py
1  a = 1
2  print(type(a))
3  a = "Hi there!"
4  print(type(a))

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

弱类型不等于没有类型！

书写代码时不用刻意关注类型，但是编程语言内部仍然有类型



Tea, Coffee, Beer, Popcorn, Icecream, Goldfish.....

变量命名

问题：所有变量都有自己的名字。那么我们可以随便对变量进行命名吗？

变量的名字需不需要满足一定的规则呢？

变量命名规则

1. 名称只能由数字、字母和下划线组成
2. 第一个字符不能用数字
3. 只要符合上述两条规则，你就可以随意命名，但还要避开Python关键字（keyword）

关键字：具有特殊含义的单词

指出下列合法的变量名

`int = 35`

`7c = 20*4`

`_python = "this is python"`

变量命名规范

规则(rule)是强制性的, 规范(convention)是约定俗称的习惯。

- 如果你遵循这些规范, 它会大大提高你的代码的可读性, 也会让你显得更为专业。

变量命名规范

1. 变量要可以自我描述(self-descriptive)

```
x = 40
```

```
student_count = 40
```

变量命名规范

2. 字母全部小写，单词中间用下划线隔开

```
good_weather = True
```

```
purchase_price = 10.5
```


变量命名规范

3. 不要过长，一般不超过三个单词

`this_is_a_really_long_word = 0`



使用变量前要先赋值

```
a  
print(a)  #error
```

运算符

Python使用标准数学运算符来进行四则运算

```
a = a+a
```

```
a = a+a
```

```
a = a+a
```

运算符细节

1. 当一个运算数为浮点数的时候，运算结果也是浮点数

```
print(1.0*2+5)
```

运算符细节

2. Python中除法的结果是浮点数

```
print(10/2)
```

```
print(-5/3)
```

运算符细节

3. `//`: 除法取整, 取整方式为向下取整

```
print(15//2)
```

```
print(-4//3)
```

模运算符 (modulo operator)

取余

```
a = 5%2
```

```
b = 15%4
```

```
print(a)
```

```
print(b)
```

优先级

括号 > 乘、除、模 > 加、减

```
a = 5
```

```
b = 2
```

```
c = 3
```

```
print(5 + a * b - c + 1 % 4)
```


小结

- 变量的命名需要符合相应的规则
- 变量的命名最好符合相应的规范
- Python的运算需要满足优先级，括号优先级永远最高
- /的结果是浮点数，//是返回商的向下取整
- %是取模符号