Python基础语法

目录

Contents

学习目标

Learning Objectives

1. 掌握字面量的含义
2. 了解常见的字面量类型
3. 基于print语句完成各类字面量的输出

什么是字面量

字面量：在代码中，被写下来的的固定的值，称之为字面量

Python中有哪些值可以被写下来？

如何在代码中写它们呢？

常用的值类型

Python中常用的有6种值（数据）的类型

类型	描述	说明
数字（Number）	支持 • 整数（int） • 浮点数（float） • 复数（complex） • 布尔（bool）	整数（int），如：10、-10
		浮点数（float），如：13.14、-13.14
		复数（complex），如：4+3j，以j结尾表示复数
		布尔（bool）表达现实生活中的逻辑，即真和假，True表示真，False表示假。 True本质上是一个数字记作1，False记作0
字符串（String）	描述文本的一种数据类型	字符串（string）由任意数量的字符组成
列表（List）	有序的可变序列	Python中使用最频繁的数据类型，可有序记录一堆数据
元组（Tuple）	有序的不可变序列	可有序记录一堆不可变的Python数据集合
集合（Set）	无序不重复集合	可无序记录一堆不重复的Python数据集合
字典（Dictionary）	无序Key-Value集合	可无序记录一堆Key-Value型的Python数据集合

字符串

字符串（string），又称文本，是由任意数量的字符如中文、英文、各类符号、数字等组成。所以叫做字符的串

如：

- ″黑马程序员″
- ″学Python来黑马″
- ″!@#$%^&″
- ″传智教育的股票代码是：003032″

好像有哪里不对

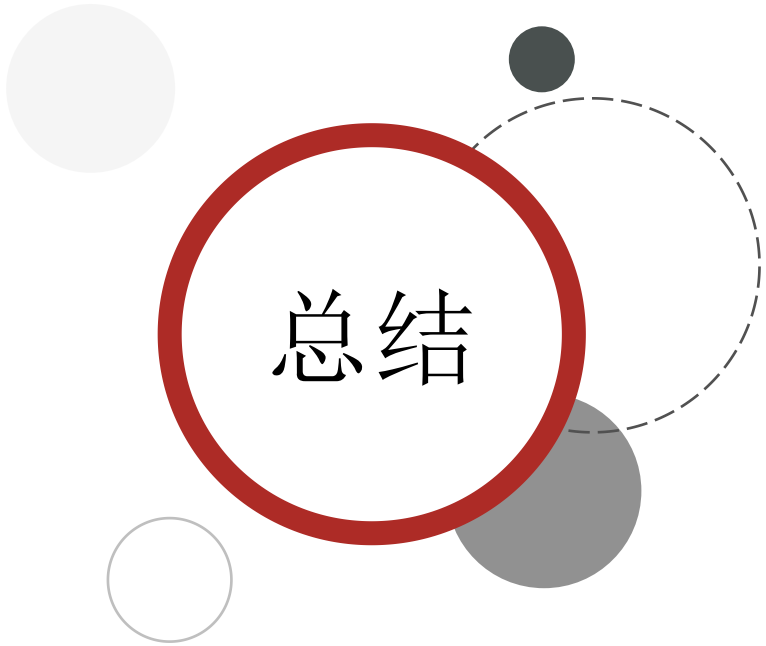
Python中，字符串需要用双引号（"）包围起来

被引号包围起来的，都是字符串

如何在代码中写它们

我们目前要学习的这些类型，如何在代码中表达呢？

类型	程序中的写法	说明
整数	666，-88	和现实中的写法一致
浮点数（小数）	13.14，-5.21	和现实中的写法一致
字符串（文本）	"黑马程序员"	程序中需要加上双引号来表示字符串



1. 掌握字面量的含义

代码中，被写在代码中的固定的值，称之为字面量

2. 常见的字面量类型

我们目前了解：整数、浮点数、字符串这三类即可

3. 如何基于print语句完成各类字面量的输出

print(字面量)，如：

- print(10)，输出整数10
- print(13.14)，输出浮点数13.14
- print("黑马程序员")，输出字符串：黑马程序员

目录

Contents

学习目标

Learning Objectives

1. 了解注释的作用

2. 能够使用单行注释和多行注释

注释的作用

```
666
13.14
"黑马程序员"

print(666)
print(13.14)
print("黑马程序员")
```

未使用注释代码

```
"""
演示：
- 各类字面量的写法
- 通过print语句输出各类字面量
"""
# 写一个整数字面量
666
# 写一个浮点数字面量
13.14
# 写一个字符串字面量
"黑马程序员"

# 通过print语句输出各类字面量
print(666)
print(13.14)
print("黑马程序员")
```

使用注释的代码

注释：在程序代码中对程序代码进行解释说明的文字。

作用：注释不是程序，**不能被执行**，只是对程序代码进行解释说明，让别人可以看懂程序代码的作用，能够大大增强程序的可读性。

注释的分类

- 单行注释：以 #开头，#右边 的所有文字当作说明，而不是真正要执行的程序，起辅助说明作用

```
1    # 我是单行注释
2    print("Hello World")
```

注意，#号和注释内容一般建议以一个空格隔开

- 多行注释： 以 一对三个双引号 引起来（"""注释内容"""）来解释说明一段代码的作用使用方法

```
1    """
2        我是多行注释
3        诗名:悯农
4        作者:李绅
5    """
6    print("锄禾日当午")
7    print("汗滴禾下土")
8    print("谁知盘中餐")
9    print("粒粒皆辛苦")
```

注释实战

按照如图所示，对代码添加

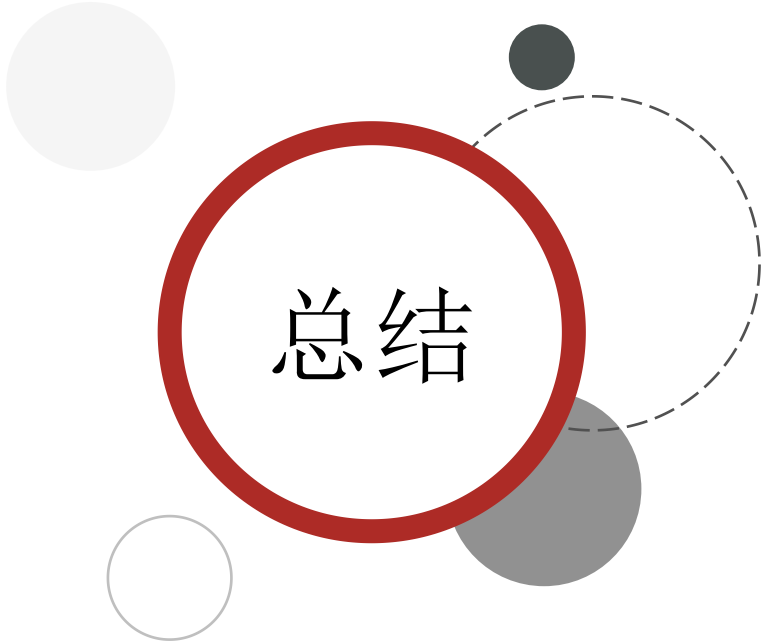
- 单行注释以及
- 多行注释

添加完成注释后，执行程序

验证注释是否对程序产生影响

```
"""
演示：
- 各类字面量的写法
- 通过print语句输出各类字面量
"""
# 写一个整数字面量
666
# 写一个浮点数字面量
13.14
# 写一个字符串字面量
"黑马程序员"

# 通过print语句输出各类字面量
print(666)
print(13.14)
print("黑马程序员")
```



1. 注释的作用是？

注释是代码中的解释型语句，用来对代码内容进行注解

注释不是代码，不会被程序执行

2. 单行注释如何定义？

通过 # 号定义，在#号右侧的所有内容均作为注释

建议在#号和注释内容之间，间隔一个空格

单行注释一般用于对一行或一小部分代码进行解释

3. 多行注释如何定义？

通过一对三个引号来定义(""”注释内容”""），引号内部均是注释，可以换行

多行注释一般对：Python文件、类或方法进行解释

思考

1. 思考第二个print语句会执行吗？

会执行的弹幕扣1，不会执行的弹幕扣2

```
print("Hi")  # print("Hello")
```

这是注释
不会执行

拓展

关于注释的面试题

1. 单行注释中能否使用多行注释？

2. 多行注释中能否使用单行注释？

3. 多行注释中能否使用多行注释？

目录

Contents

学习目标

Learning Objectives

理解变量的作用及特征

掌握变量的定义方式

什么是变量

变量：**在程序运行时**，能储存计算结果或能表示值的抽象概念。

简单的说，变量就是在程序运行时，记录数据用的

变量的定义格式

```
变量名称 = 变量的值
```

每一个变量都有自己存储的值（内容），称之为：**变量值**

赋值，表示将等号**右侧的值，赋予左侧的变量**

每一个变量都有自己的名称，称之为：**变量名，也就是变量本身**

变量就像盒子
可以存放内容

案例 模拟钱包

变量的特征

变量，从名字中可以看出，表示“量”是可变的。

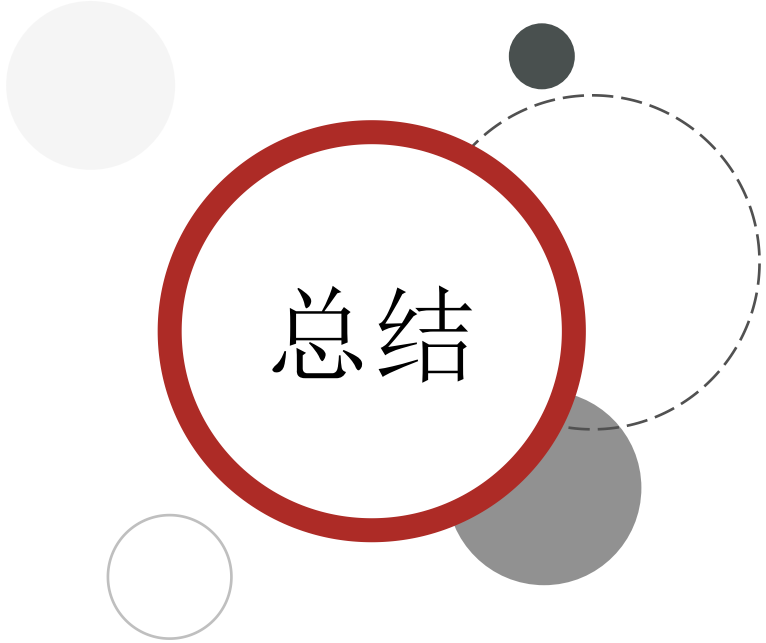
所以，变量的特征就是，**变量存储的数据，是可以发生改变的。**

为什么必须要使用变量？

都是输出内容，直接输出不行吗？

变量的目的是存储运行过程的数据

存储的目的是为了：重复使用



1. 变量是什么，有什么作用？

变量就是在程序运行时，记录数据用的

2. 变量的定义格式是？

变量名 = 变量值

3. 变量的特征是？

变量的值可以改变

总结

4. print语句如何输出多份内容？

print(内容1, 内容2, ..., 内容N)

5. Python中如何做减法？

使用符号 - 即可完成减法运算

拓展：加（+）、减（-）、乘（*）、除（/）

练习

求钱包余额

请在程序中，定义如下变量：

- 钱包余额(变量名：money)，初始余额50

请通过程序计算，在购买了：

- 冰淇淋10元
- 可乐5元

后，钱包余额还剩余多少元。请通过print语句按照下图所示，进行输出：

目录

Contents

学习目标

Learning Objectives

掌握使用type()语句查看数据的类型

理解变量无类型而数据有类型的概念

数据类型

在学习字面量的时候，我们了解到：数据是有类型的。

目前在入门阶段，我们主要接触如下三类数据类型：

类型	描述	说明
string	字符串类型	用引号引起来的数据都是字符串
int	整型（有符号）	数字类型，存放整数 如 -1,10, 0 等
float	浮点型（有符号）	数字类型，存放小数 如 -3.14, 6.66

string、int、float这三个英文单词，就是类型的标准名称。

type()语句

那么，问题来了，如何验证数据的类型呢？

我们可以通过type()语句来得到数据的类型：

语法：

type(被查看类型的数据)

type()语句的使用方式

1. 在print语句中，直接输出类型信息：

```
print(type("黑马程序员"))
print(type(666))
print(type(11.345))
```

```
test ×
D:\dev\Python\Python3.10
<class 'str'>
<class 'int'>
<class 'float'>
```

str是string的缩写

2. 用变量存储type()的结果（返回值）

```
1  string_type = type("黑马程序员")
2  int_type = type(666)
3  float_type = type(11.345)
4  print(string_type)
5  print(int_type)
6  print(float_type)
```

```
Run: test ×
D:\dev\Python\Python3.10.4\pytho
<class 'str'>
<class 'int'>
<class 'float'>
```

type()语句的使用方式

查看的都是<字面量>的类型，能查看变量中存储的数据类型吗？

那当然：可以

```
name = "黑马程序员"
name_type = type(name)
print(name_type)
```

```
test
D:\dev\Python\Python3.10
<class 'str'>
```

变量有类型吗？

我们通过type(变量)可以输出类型，这是查看变量的类型还是数据的类型？

查看的是：变量存储的数据的类型。因为，变量无类型，但是它存储的数据有。

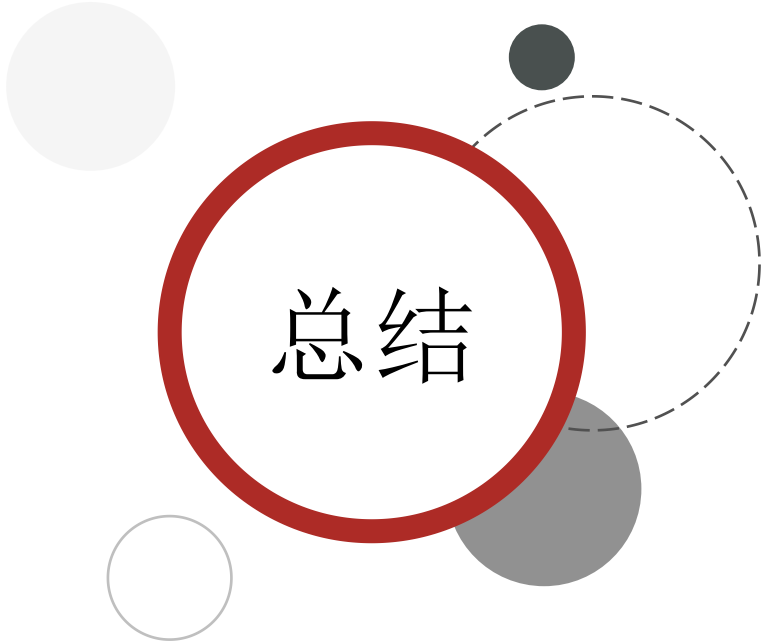
足球盒子

篮球盒子

盒子是足球、篮球吗？

我们可能会说：字符串变量

但要知道，不是变量是字符串，而是它存储了：字符串



1. 使用什么语句可以查看数据的类型？

type()

2. 如下代码，name_type变量可以存储变量name的类型信息，是因为？

```
name = "黑马程序员"
name_type = type(name)
```

因为type()语句会给出结果（返回值）

3. 变量有没有类型？

没有，字符串变量表示变量存储了字符串而不是表示变量就是字符串

字符串类型的不同定义方式

字符串有3种不同的定义方式：

双引号定义法 "字符串"

单引号定义法 '字

三引号定义法 """字符串"""

```
text1 = "我是字符串（文本）数据"
text2 = '我也是字符串（文本）数据哦'
text3 = """没想到吧，我即能做注释，也是字符串哦"""
```

三引号定义法，表示在一堆三个双引号的范围内，均是字符串，如下：

```
text = """
在三个引号的包围圈内
全部都是
字符串哦
"""
```

要注意的是，包含范围是：从三个引号开始，到下一个三个引号结

目录

Contents

学习目标

Learning Objectives

掌握如何在字符串、整数、浮点数之间进行相互转换

了解转换的注意事项

为什么要转换类型

数据类型之间，在特定的场景下，是可以相互转换的，如字符串转数字、数字转字符串等

那么，我们为什么要转换它们呢？

数据类型转换，将会是我们以后经常使用的功能。

如：

- 从文件中读取的数字，默认是字符串，我们需要转换成数字类型
- 后续学习的input()语句，默认结果是字符串，若需要数字也需要转换
- 将数字转换成字符串用以写出到外部系统
- 等等

用途很多，那么让我们来学习一下如何转换吧。

常见的转换语句

语句(函数)	说明
int(x)	将x转换为一个整数
float(x)	将x转换为一个浮点数
str(x)	将对象 x 转换为字符串

同前面学习的type()语句一样，这三个语句，都是带有结果的（返回值）
我们可以用print直接输出
或用变量存储结果值

类型转换注意事项

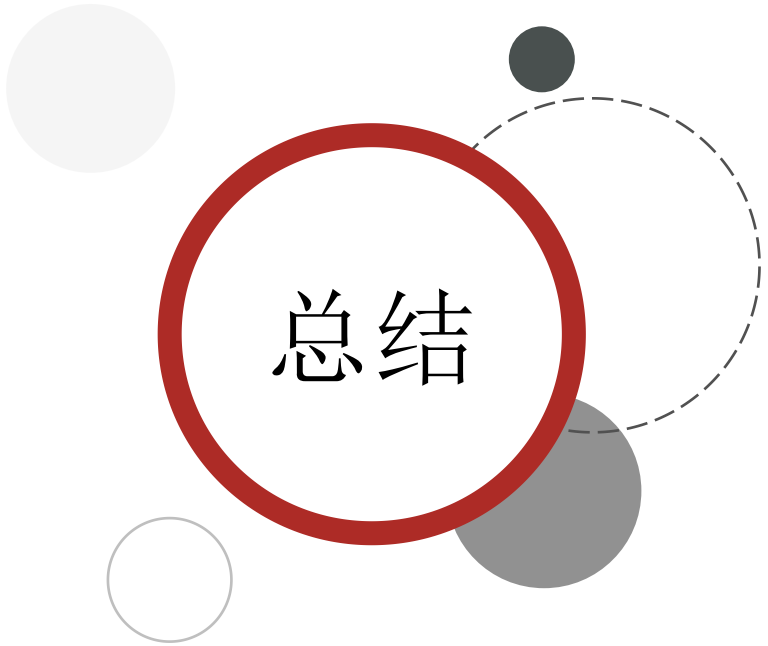
类型转换不是万能的，毕竟强扭的瓜不会甜，我们需要注意：

1. 任何类型，都可以通过str()，转换成字符串
2. 字符串内必须真的是数字，才可以将字符串转换为数字

```
v_str = "我不是数字"
num = int(v_str)
```

字符串内不是数字，是无法完成到数字的转换的

```
est (1)
D:\dev\Python\Python3.10.4\python.exe D:/python-learn/01_Python基础知识/
Traceback (most recent call last):
  File "D:\python-learn\01_Python基础知识\test.py", line 2, in <module>
    num = int(v_str)
ValueError: invalid literal for int() with base 10: '我不是数字'
```



1. 字符串、整数、浮点数类型转换的语句是？

语句(函数)	说明
int(x)	将x转换为一个整数
float(x)	将x转换为一个浮点数
str(x)	将对象 x 转换为字符串

2. 任何类型都可以转换成字符串，对不对？

正确

3. 字符串可以随意转换成数字，对不对？

错误，字符串内必须只有数字才可以

4. 浮点数转整数会丢失什么？

丢失精度，也就是小数部分

1. 描述什么是变量以及变量命名规范。
2. 熟悉Python中的7种数据类型以及定义方式。
3. 定义4个变量，需求：姓名：孙悟空，年龄：600岁，技能：筋斗云、72变，主要战绩：大闹天宫
4. 定义变量，c1 = '可乐'，c2 = '牛奶'，通过Python代码把c1内容调整为牛奶，c2调整为可乐。（提示：两个数的交换）

目录

Contents

学习目标

Learning Objectives

理解什么是标识符

掌握标识符的命名规则

掌握变量的命名规范

什么是标识符

在Python程序中，我们可以给很多东西起名字，比如：

- 变量的名字
- 方法的名字
- 类的名字, 等等

这些名字，我们把它统一的称之为标识符，用来做内容的标识。

所以，标识符：

是用户在编程的时候所使用的一系列名字，用于给变量、类、方法等命名。

既然要起名字，就会有对应的限制

标识符命名规则

Python中，标识符命名的规则主要有3类：

- 内容限定
- 大小写敏感
- 不可使用关键字

标识符命名规则 - 内容限定

标识符命名中，只允许出现：

- 英文
- 中文
- 数字
- 下划线（_）

这四类元素。

其余任何内容都不被允许。

注意

1. 不推荐使用中文
2. 数字不可以开头

a
a_b
_a
_a_b
a1
a_b_1

1
1_
1_a

标识符命名规则 - 大小写敏感

以定义变量为例：

Andy = “安迪1”

andy = “安迪2”

字母a的大写和小写，是完全能够区分的。

```
Andy = "安迪1"
andy = "安迪2"
print(Andy)
print(andy)

est (1)
D:\dev\Python\Python3.10.
安迪1
安迪2
```

标识符命名规则 - 不可使用关键字

Python中有一系列单词，称之为关键字
关键字在Python中都有特定用途
我们不可以使用它们作为标识符

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

变量命名规范

学完了标识符（变量、类、方法）的命名**规则**后，我们在来学习标识符的命名**规范**。

- 变量名
- 类名
- 方法名

不同的标识符，有不同的规范。

我们目前只接触到了：变量。所以，目前学习：变量的命名规范。

- 见名知意
- 下划线命名法
- 英文字母全小写

变量命名规范 - 见名知意

变量的命名要做到：

- 明了：尽量做到，看到名字，就知道是什么意思

```
a = "张三"
b = 11
```

```
name = "张三"
age = 11
```

- 简洁：尽量在确保“明了”的前提下，减少名字的长度

```
a_person_name = "张三"
```

```
name = "张三"
```

变量命名规范 - 下划线命名法

多个单词组合变量名，要使用下划线做分隔。

```
firstnumber = 1
studentnickname = "小明"
```

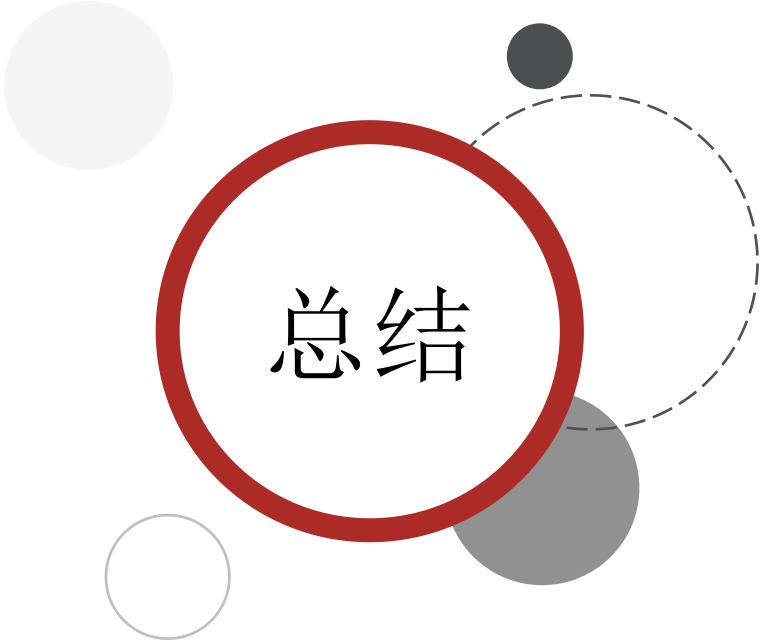
```
first_number = 1
student_nickname = "小明"
```

变量命名规范 - 英文字母全小写

命名变量中的英文字母，应全部小写：

```
Name = "张三"
Age = 11
```

```
name = "张三"
age = 11
```



不遵守规则：会出现问题

不遵守规范：不太高级

1. 什么是标识符？

1. 用户编写代码时，对变量、类、方法等编写的名字，叫做标识符。

2. 标识符的命名规则？

- 内容限定
  - （中文、英文、数字、下划线）
- 大小写敏感
- 不可使用关键字

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

3. 变量的命名规范？

- 见名知意
- 下划线命名法
- 英文字母全小写

目录

Contents

学习目标

Learning Objectives

了解Python中常见

- 算术（数学）运算符
- 赋值运算符

算术（数学）运算符

运算符	描述	实例
+	加	两个对象相加 a + b 输出结果 30
-	减	得到负数或是一个数减去另一个数 a - b 输出结果 -10
*	乘	两个数相乘或是返回一个被重复若干次的字符串 a * b 输出结果 200
/	除	b / a 输出结果 2
//	取整除	返回商的整数部分 9//2 输出结果 4 , 9.0//2.0 输出结果 4.0
%	取余	返回除法的余数 b % a 输出结果 0
**	指数	a**b 为10的20次方， 输出结果 100000000000000000000

算术运算符的演示

加减乘除和求平方，我们在前面已经使用过啦。

现在在带上：整除以及求余数，一起试一试吧。

```
print("1 + 1结果是：%d" % (1 + 1))
print("2 - 1结果是：%d" % (2 - 1))
print("1 * 3结果是：%d" % (1 * 3))
print("9 / 3结果是：%d" % (9 / 3))
print("9 // 2（9整除2）结果是：%d" % (9 // 2))
print("9 %% 2（9余2的结果是）结果是：%d" % (9 % 2))
print("2 的 6 次方是：结果是：%d" % (2 ** 6))
```

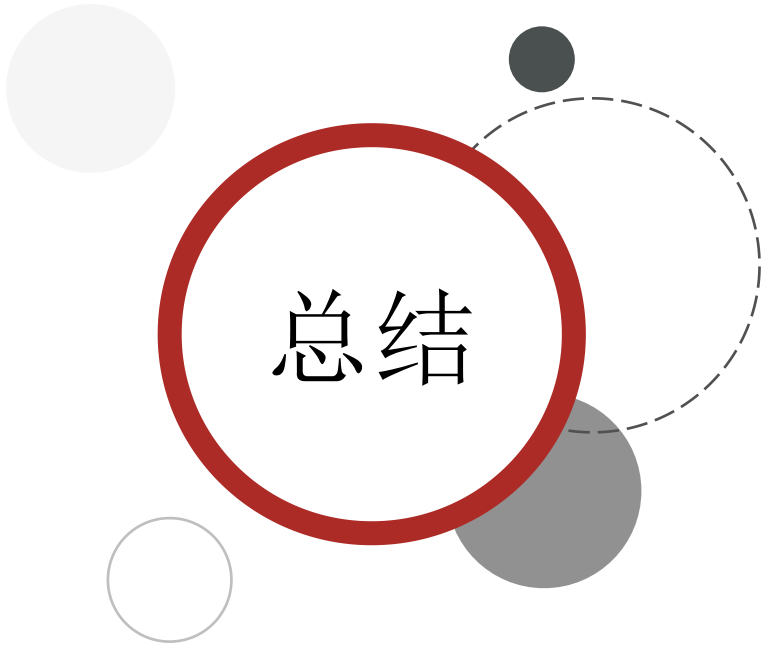
```
test (1)
D:\dev\Python\Python3.10.4\python.exe D:/pytho
1 + 1结果是：2
2 - 1结果是：1
1 * 3结果是：3
9 / 3结果是：3
9 // 2（9整除2）结果是：4
9 % 2（9余2的结果是）结果是：1
2 的 6 次方是：结果是：64
```

赋值运算符

运算符	描述	实例
=	赋值运算符	把 = 号右边的结果 赋给 左边的变量，如 num = 1 + 2 * 3，结果num的值为7

复合赋值运算符

运算符	描述	实例
+=	加法赋值运算符	c += a 等效于 c = c + a
-=	减法赋值运算符	c -= a 等效于 c = c - a
*=	乘法赋值运算符	c *= a 等效于 c = c * a
/=	除法赋值运算符	c /= a 等效于 c = c / a
%=	取模赋值运算符	c %= a 等效于 c = c % a
**=	幂赋值运算符	c **= a 等效于 c = c ** a
//=	取整除赋值运算符	c //= a 等效于 c = c // a



1. 常见的算术（数学）运算符有：

   加（+）、减（-）、乘（*）、除（/）、整除（//）、取余（%）、求平方（）**

2. 赋值运算符有：

   - **标准赋值： =**
   - **复合赋值：+=、-=、*=、/=、//=、%=、**=**

目录

Contents

目录

Contents

- 字符串扩展
  - 字符串的三种定义方式
  - 字符串拼接
  - 字符串格式化
  - 格式化的精度控制
  - 字符串格式化方式2
  - 对表达式进行格式化

字符串的三种定义方式

- 字符串在Python中有多种定义形式：

1. 单引号定义法：`name = '黑马程序员'`
2. 双引号定义法：`name = "黑马程序员"`
3. 三引号定义法：`name = """黑马程序员"""`

三引号定义法，和多行注释的写法一样，同样支持换行操作。

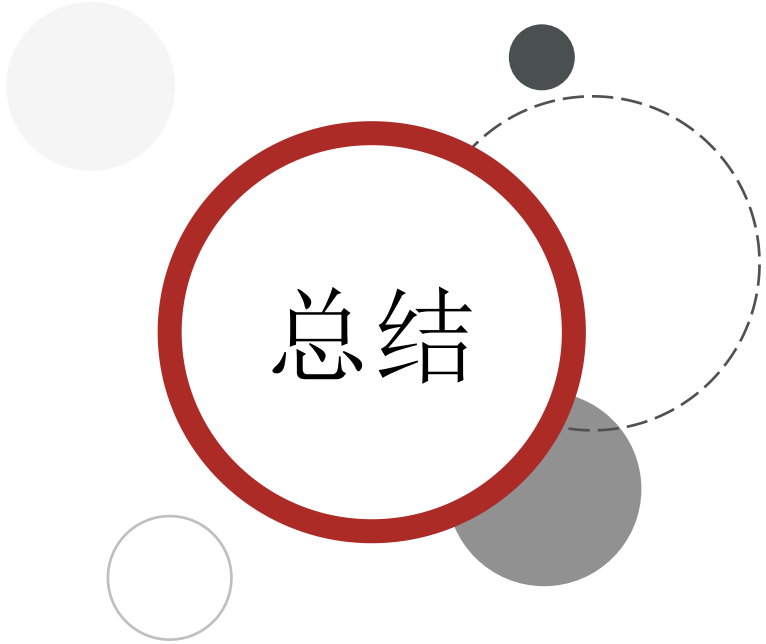
使用变量接收它，它就是字符串

不使用变量接收它，就可以作为多行注释使用。

字符串的引号嵌套

思考：如果我想要定义的字符串本身，是包含：单引号、双引号自身呢？如何写？

- 单引号定义法，可以内含双引号
- 双引号定义法，可以内含单引号
- 可以使用转移字符（\）来将引号解除效用，变成普通字符串



1. 字符串的三种定义方式：
   - **单引号方式**
   - **双引号方式**
   - **三引号方式**
2. 引号的嵌套
   - **可以使用：\来进行转义**
   - **单引号内可以写双引号或双引号内可以写单引号**

目录

Contents

学习目标

Learning Objectives

1. 掌握如何拼接字符串

字符串拼接

如果我们有两个字符串（文本）字面量，可以将其拼接成一个字符串，通过+号即可完成，如：

```
print("学IT来黑马" + "月薪过万")
```

输出结果：

```
学IT来黑马月薪过万
```

不过一般，单纯的2个字符串字面量进行拼接显得很呆，一般，字面量和变量或变量和变量之间会使用拼接，如：

```
name = "黑马程序员"
print("我的名字是：" + name + "，我可以教大家IT技能")
```

```
D:\dev\Python\Python3.10.4\python.exe
我的名字是：黑马程序员，我可以教大家IT技能
```

字符串拼接

既然，字符串和变量进行拼接，我们来完成一个小需求：

定义2个变量：

- name，内容是“黑马程序员”
- address，内容是“建材城东路9号”

要求写程序通过print语句以及字符串拼接的方式输出：

“我是：黑马程序员，我的地址是：建材城东路9号院“

```
1  name = "黑马程序员"
2  address = "建材城东路9号院"
3  print("我是：" + name + "，我的地址是：" + address)
```

代码

```
Run:  test (1)
D:\dev\Python\Python3.10.4\python.exe D:/python-le
我是：黑马程序员，我的地址是：建材城东路9号院
```

运行结果

字符串拼接

既然可以和字符串变量完成拼接，那么，是否可以和其它变量类型如数字类型完成拼接呢？

让我们试一试。

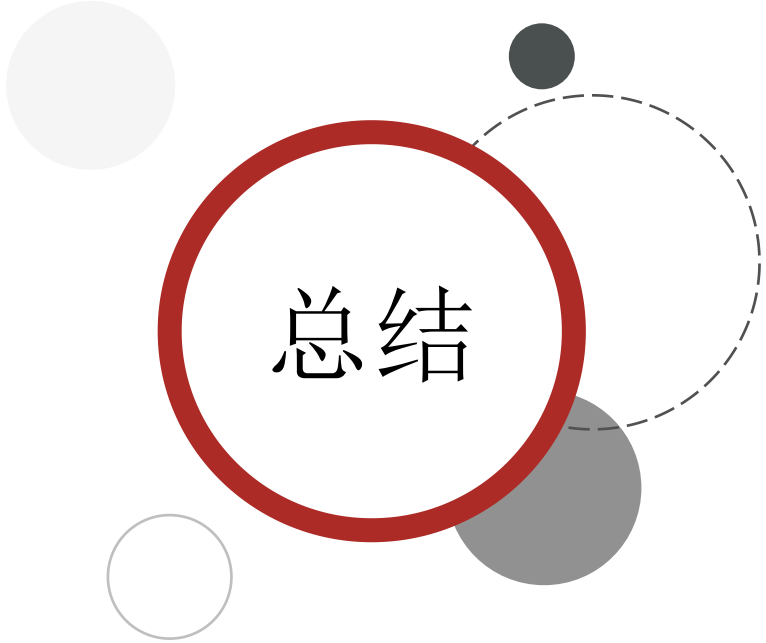
```
name = "传智播客"
set_up_year = 2006
print("我是：" + name + "，我成立于：" + set_up_year)
```

```
D:\dev\Python\Python3.10.4\python.exe D:/python-learn/01_Python基础知识/
Traceback (most recent call last):
  File "D:\python-learn\01_Python基础知识\test.py", line 3, in <module>
    print("我是：" + name + "，我成立于：" + set_up_year)
TypeError: can only concatenate str (not "int") to str
```

字符串无法和非字符串变量进行拼接

因为类型不一致，无法接上

就像接力赛一样，不是队友，不能接力的哦



1. 如何完成字符串拼接？

使用“+”号连接字符串变量或字符串字面量即可

2. 有哪些注意事项？

无法和非字符串类型进行拼接

目录

Contents

学习目标

Learning Objectives

1. 掌握通过占位的形式拼接字符串（字符串格式化）

字符串格式化

我们会发现，这个拼接字符串也不好用啊

1. 变量过多，拼接起来实在是太麻烦了

```
print("我是" + name + "，我的性别是：" + sex + "，我住在：" + address + "，我的爱好是：" + hobby)
```

2. 字符串无法和数字或其它类型完成拼接。

所以，有没有其它方式，即方便又支持拼接其它类型呢？

这个方式，就是字符串的格式化

字符串格式化

我们可以通过如下语法，完成字符串和变量的快速拼接。

```
name = "黑马程序员"
message = "学IT就来 %s" % name
print(message)
```

```
test ×
D:\dev\Python\Python3.10.4\python.
学IT就来 黑马程序员
```

其中的，%s

- % 表示：我要占位
- s 表示：将变量变成字符串放入占位的地方

所以，综合起来的意思就是：我先占个位置，等一会有个变量过来，我把它变成字符串放到占位的位置

字符串格式化

那，数字类型呢？可以不可以占位？

那必须可以，我们来尝试如下代码：

```
1  class_num = 57
2  avg_salary = 16781
3  message = "Python大数据学科，北京%s期，毕业平均工资：%s" % (class_num, avg_salary)
4  print(message)
```

```
Run:  test
D:\dev\Python\Python3.10.4\python.exe D:/python-learn/01_你好Python/test.py
Python大数据学科，北京57期，毕业平均工资：16781
```

多个变量占位

变量要用括号括起来

并按照占位的顺序填入

数字也能用%s占位吗？

可以的哦，这里是将数字 转换成了 字符串哦

也就是数字57，变成了字符串"57"被放入占位的地方

其中的，%s

- % 表示：我要占位
- s 表示：将变量变成字符串放入占位的地方

字符串格式化

数字类型，也太没有地位了吧，竟然要被转成字符串拼接。

有没有体面一点的方式，让数字以其原本的面貌拼接进去呢？

安排。

Python中，其实支持非常多的数据类型占位

最常用的是如下三类

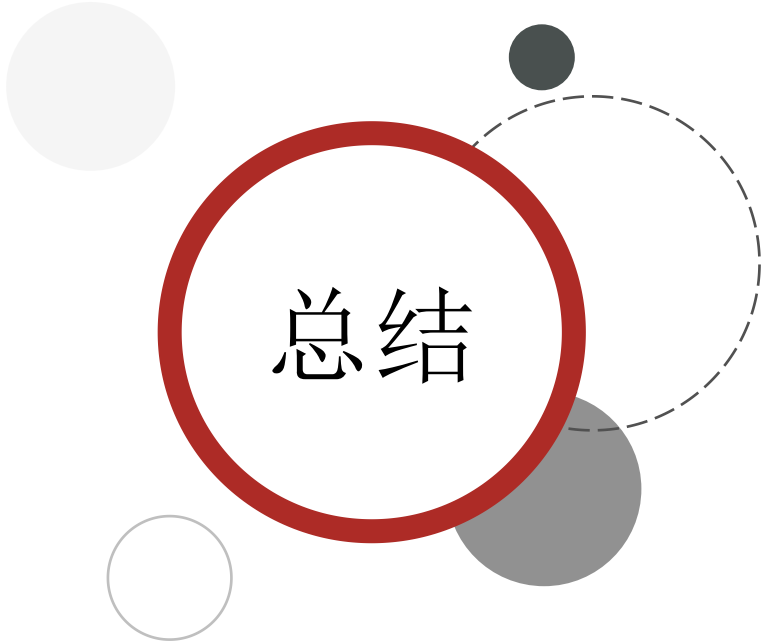
格式符号	转化
%s	将内容转换成字符串，放入占位位置
%d	将内容转换成整数，放入占位位置
%f	将内容转换成浮点型，放入占位位置

字符串格式化

如下代码，完成字符串、整数、浮点数，三种不同类型变量的占位

```
name = "传智播客"
set_up_year = 2006
stock_price = 19.99
message = "我是：%s，我成立于：%d，我今天的股价是：%f" % (name, set_up_year, stock_price)
print(message)
```

```
est (1) ×
D:\dev\Python\Python3.10.4\python.exe D:/python-learn/01_Python基础知识/test.py
我是：传智播客，我成立于：2006，我今天的股价是：19.990000
```



1. 字符串格式化的语法?

"%占位符" % 变量

2. 常用占位符有哪3个?

- 字符串： %s
- 整数: %d
- 浮点数: %f

目录

Contents

学习目标

Learning Objectives

1. 掌握格式化字符串的过程中做数字的精度控制

字符串格式化

如下代码，完成字符串、整数、浮点数，三种不同类型变量的占位

```
name = "传智播客"
set_up_year = 2006
stock_price = 19.99
message = "我是：%s，我成立于：%d，我今天的股价是：%f" % (name, set_up_year, stock_price)
print(message)
```

```
est (1) ×
D:\dev\Python\Python3.10.4\python.exe D:/python-learn/01_Python基础知识/test.py
我是：传智播客，我成立于：2006，我今天的股价是：19.990000
```

细心的同学可能会发现：

浮点数19.99，变成了19.990000输出

这里我们就要讲解一下，字符串格式化之“数字精度控制”

字符串格式化 - 数字精度控制

我们可以使用辅助符号"m.n"来控制数据的宽度和精度

- m，控制宽度，要求是数字（很少使用），设置的宽度小于数字自身，不生效
- .n，控制小数点精度，要求是数字，会进行小数的四舍五入

示例：

- %5d：表示将整数的宽度控制在5位，如数字11，被设置为5d，就会变成：[空格][空格][空格]11，用三个空格补足宽度。
- %5.2f：表示将宽度控制为5，将小数点精度设置为2

  小数点和小数部分也算入宽度计算。如，对11.345设置了%7.2f 后，结果是：[空格][空格]11.35。2个空格补足宽度，小数部分限制2位精度后，四舍五入为 .35
- %.2f：表示不限制宽度，只设置小数点精度为2，如11.345设置%.2f后，结果是11.35

字符串格式化 - 数字精度控制

体验一下如下代码的快乐吧。

```
num1 = 11
num2 = 11.345
print("数字11宽度限制5，结果：%5d" % num1)
print("数字11宽度限制1，结果：%1d" % num1)
print("数字11.345宽度限制7，小数精度2，结果 ：%7.2f" % num2)
print("数字11.345不限制宽度，小数精度2，结果 ：%.2f" % num2)
```

```
est (1)
D:\dev\Python\Python3.10.4\python.exe D:/python-learn/01_Python基础知识/test.py
数字11宽度限制5，结果：    11   宽度5，补了3个空格
数字11宽度限制1，结果：11      宽度小于数字本身，无影响
数字11.345宽度限制7，小数精度2，结果 ：  11.35   宽度7，补了2个空格，小数精度2，四舍五入后为.35
数字11.345不限制宽度，小数精度2，结果 ：11.35    不限制宽度，小数点后四舍五入后为.35
```

既快乐又神奇吧

总结

1. 精度控制的语法是：

m.n的形式控制，如%5d、%5.2f、%.2f

m和.n均可省略

2. 如果m比数字本身宽度还小，会发生什么事？

m不生效

3. .n会对小数部分做精度限制，同时：？

会对小数部分做四舍五入

目录

Contents

- 字符串扩展
  - 字符串的三种定义方式
  - 字符串拼接
  - 字符串格式化
  - 格式化的精度控制
  - 字符串格式化方式2
  - 对表达式进行格式化

学习目标

Learning Objectives

1. 掌握快速字符串格式化的方式

字符串格式化 - 快速写法

目前通过%符号占位已经很方便了，还能进行精度控制。

可是追求效率和优雅的Python，是否有更加优雅的方式解决问题呢？

那当然：有

通过语法：f"内容{变量}"的格式来快速格式化

看如下代码

```
name = "传智播客"
set_up_year = 2006
stock_price = 19.99
print(f"我是{name}，我成立于：{set_up_year}，我今天的股票价格是：{stock_price}")
```

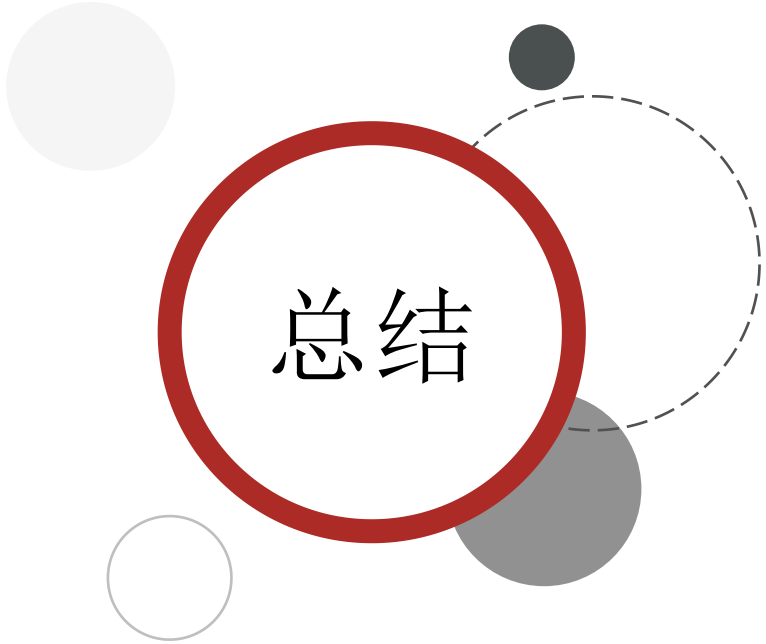
```
test (1)
D:\dev\Python\Python3.10.4\python.exe D:/python-learn/01_Python基础知识
我是传智播客，我成立于：2006，我今天的股票价格是：19.99
```

不做精度控制，原样输出

这种写法不做精度控制

也不理会类型

适用于快速格式化字符串



1. 可以通过

f”{变量} {变量}”的方式进行快速格式化

2. 这种方式：

- 不理会类型
- 不做精度控制

适合对精度没有要求的时候快速使用

目录

Contents

学习目标

Learning Objectives

1. 了解什么是表达式
2. 掌握对表达式进行字符串格式化

字符串格式化 - 表达式的格式化

刚刚的演示，都是基于变量的。

可是，我想更加优雅些，少写点代码，直接对“表达式”进行格式化是否可行呢？

那么，我们先了解一下什么是表达式。

表达式：一条具有明确执行结果的代码语句

如：

1 + 1、5 * 2，就是表达式，因为有具体的结果，结果是一个数字

又或者，常见的变量定义：

name = “张三”　　age = 11 + 11

等号右侧的都是表达式呢，因为它们有具体的结果，结果赋值给了等号左侧的变量。

字符串格式化 - 表达式的格式化

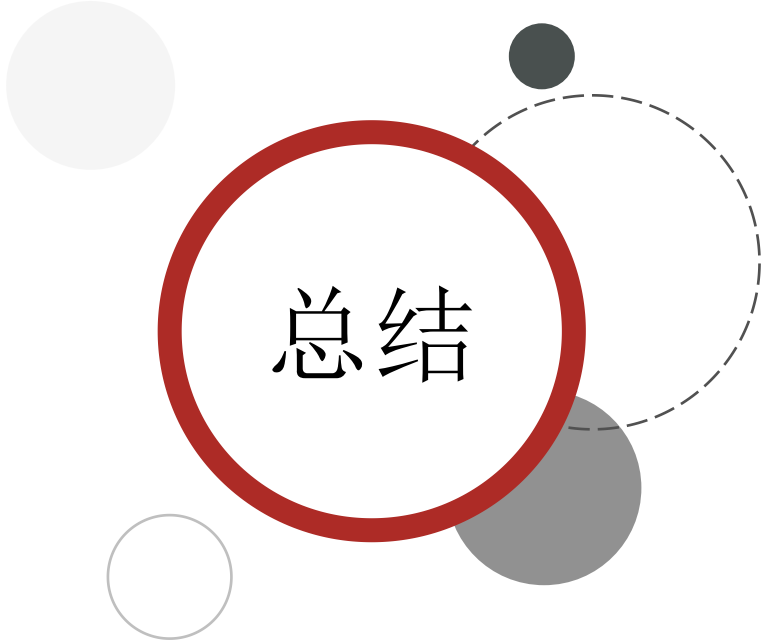
那么，对于字符串格式化，能否直接格式化一个表达式呢？

可以，上代码：

```
print("1 * 1的结果是：%d" % (1 * 1))
print(f"1 * 1的结果是：{1 * 1}")
print("字符串在Python中的类型是：%s" % type('字符串'))
```

```
test (1)
D:\dev\Python\Python3.10.4\python.exe D:/python-
1 * 1的结果是：1
1 * 1的结果是：1
字符串在Python中的类型是：<class 'str'>
```

在无需使用变量进行数据存储的时候，可以直接格式化表达式，简化代码哦



1. 表达式是什么？

表达式就是一个具有明确结果的代码语句，如 1 + 1、type(“字符串”)、3 * 5等

在变量定义的时候，如 age = 11 + 11，等号右侧的就是表达式，也就是有具体的结果，将结果赋值给了等号左侧的变量

2. 如何格式化表达式？

- f"{表达式}"
- "%s\%d\%f" %（表达式、表达式、表达式）

练习

股价计算小程序

定义如下变量：

- name，公司名
- stock_price，当前股价
- stock_code，股票代码
- stock_price_daily_growth_factor，股票每日增长系数，浮点数类型，比如1.2
- growth_days，增长天数

计算，经过growth_days天的增长后，股价达到了多少钱

使用字符串格式化进行输出，如果是浮点数，要求小数点精度2位数。

示例输出：

```
D:\dev\Python\Python3.10.4\python.exe D:/python-learn/01_Python基础知识/test.py
公司：传智播客，股票代码：003032，当前股价：19.99
每日增长系数是：1.2，经过7天的增长后，股价达到了：71.63
```

本行要求使用 f"{变量}"的方式输出

本行要求使用 % 占位符的方式输出

红色框框都是变量，要使用格式化的方式拼接进去

提示，可以使用： 当前股价 * 增长系数 ** 增长天数， 用来计算最终股价哦

如，股价19.99 * 系数1.2 ** 7天 = 71.62778419199998，小数点现在精度2位后结果：71.63

目录

Contents

学习目标

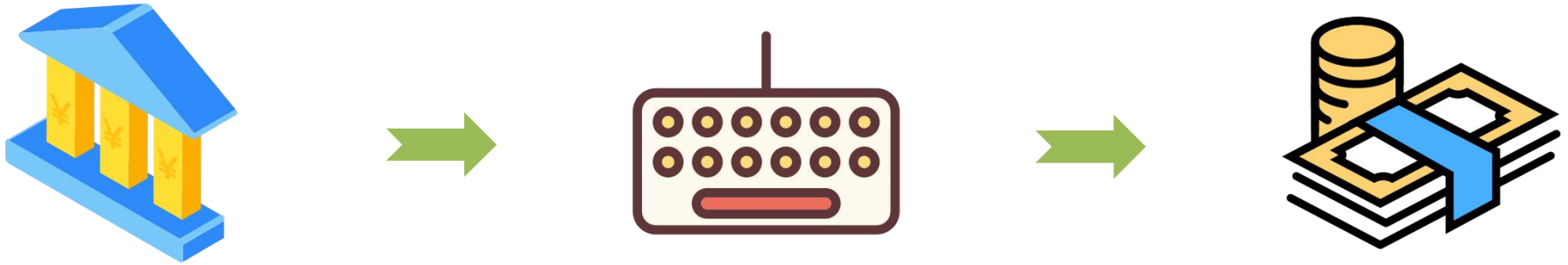
Learning Objectives

掌握input语句（函数）的使用

获取键盘输入

试想一下，我们经常遇到过程序需要我们输入信息的场景。

比如：银行取钱



如何在Python中做到读取键盘输入的内容呢？

这里就要请出input语句了

input语句（函数）

我们前面学习过print语句（函数），可以完成将内容（字面量、变量等）输出到屏幕上。

在Python中，与之对应的还有一个input语句，用来获取键盘输入。

- 数据输出：print
- 数据输入：input

使用上也非常简单：

- 使用input()语句可以从键盘获取输入
- 使用一个变量接收（存储）input语句获取的键盘输入数据即可

```
print("请告诉我你是谁？")
name = input()
print("Get！！！你是：%s" % name)
```

```
n:  test (1)
D:\dev\Python\Python3.10.4\python.ex
请告诉我你是谁？
黑马程序员
Get！！！你是：黑马程序员
```

键盘输入内容并回车

input语句（函数）

在前面的代码中，输出”请告诉我你是谁？“的print语句其实是多余的

```
print("请告诉我你是谁？")
name = input()
print("Get！！！你是：%s" % name)
```

input()语句其实是可以在要求使用者输入内容前，输出提示内容的哦，方式如下：

```
name = input("请告诉我你是谁？")
print("Get！！！你是：%s" % name)
```

```
test (1)
D:\dev\Python\Python3.10.4\py
请告诉我你是谁？黑马程序员
Get！！！你是：黑马程序员
```

如图，在input的括号内直接填入提示内容即可。

input语句获取的数据类型

我们刚刚试验的都是输入了字符串类型的数据。

那么如果我们输入数字类型或其它类型，结果会如何？

那么，让我们通过前面学习过的？？？

type()语句

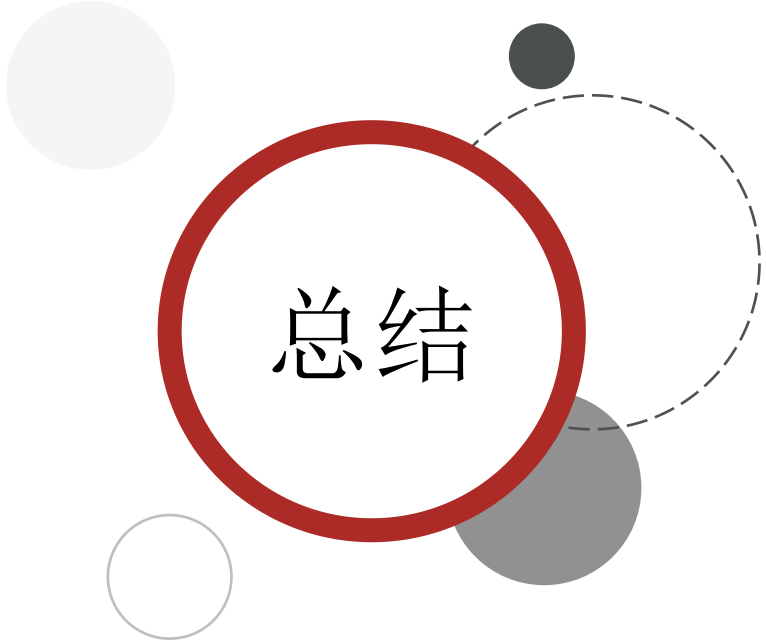
来验证一下输入内容的数据类型吧。

```
v1 = input("请输入一个字符串：")
v2 = input("请输入一个整数：")
v3 = input("请输入一个浮点数：")
v4 = input("请输入一个布尔类型：")
print(f"输入的是字符串，变量类型是{type(v1)}，内容是：{v1}")
print(f"输入的是整数，变量类型是{type(v2)}，内容是：{v2}")
print(f"输入的是浮点数，变量类型是{type(v3)}，内容是：{v3}")
print(f"输入的是布尔类型，变量类型是{type(v4)}，内容是：{v4}")
```

```
est (1)
D:\dev\Python\Python3.10.4\python.exe D:/python-learn
请输入一个字符串：你好
请输入一个整数：11
请输入一个浮点数：13.14
请输入一个布尔类型：True
输入的是字符串，变量类型是<class 'str'>，内容是：你好
输入的是整数，变量类型是<class 'str'>，内容是：11
输入的是浮点数，变量类型是<class 'str'>，内容是：13.14
输入的是布尔类型，变量类型是<class 'str'>，内容是：True
```

可以看到，无论键盘输入何种类型的数据

最终的结果都是：字符串类型的数据



1. input()语句的功能是，获取键盘输入的数据

2. 可以使用：input(提示信息)，用以在使用者输入内容之前显示提示信息。

3. 要注意，无论键盘输入什么类型的数据，获取到的数据**永远都是字符串类型**

练习

欢迎登陆小程序

定义两个变量，用以获取从键盘输入的内容，并给出提示信息：

- 变量1，变量名：user_name，记录用户名称
- 变量2，变量名：user_type，记录用户类型

并通过格式化字符串的形式，通过print语句输出欢迎信息，如下：

```
D:\dev\Python\Python3.10.4\python.exe D:/python-learn
您好：黑马程序员，您是尊贵的：SSSSSVIP 用户，欢迎您的光临。
```

变量：user_name　　变量：user_type

欢迎将练习作业，发布到评论区，看看谁的作业获得的点赞最多哟（最搞笑）

拓展-以专业视角看待变量

前言

本套Python视频教程，力求以最通俗易懂的方式来讲解知识点。

但是，对于一些知识点，我们以简单的方式学习过后呢，我们仍希望给同学们补充一下
从专业视角去看待它们的讲解。

这些高阶的拓展知识点，在刚开始大家可能听不太懂，所以我们对这些知识点不做强制要求。
同学们可以跳过这些知识点，等待以后学习更加深入后，再来回看即可。
这些高阶拓展都是独立的，跳过也不影响后面的学习。

传智教育旗下高端IT教育品牌