

The background of the slide is a light gray gradient. It is decorated with numerous realistic water droplets of various sizes. Some droplets are large and prominent, while others are small and subtle. They are scattered across the slide, with a higher concentration in the top-left and bottom-right corners. Each droplet has a soft highlight and a subtle shadow, giving it a three-dimensional appearance.

# CS001 编程零基础 Python语言入门

第二讲

# 本讲内容

名字和变量

数据类型

运算和表达式

程序输入

## 数据在内存中的存储



内存空间，可能有1G字节或者更多。



一个数据，可能有若干字节。



一个字节，但内容可能各不相同。

内存地址编号从0开始，一直到“容量-1”。  
假如你的机器只有1024字节内存，那么内存地址范围就是  
0~1023

# 如何描述和引用数据？

- 字面值

- 表示某个固定不变的值。例如：

- 字符串字面值

`'Hello World'`

`'123'`

`'上海'`

- 整数字面值：

`2017` （默认是十进制）

`0B10010111` （二进制）

- 变量

- 如果你希望Python记住某个数据在内存中某处，以便你以后使用，你需要给这个数据起个名字。
  - 这个名字就是变量名。

# 名字

```
#Name.py  
Teacher = 'Mr. Zhang'  
print(Teacher)  
print('Teacher')
```

在内存中创建了一个字符串数据'Mr. Zhang'，并且给它起了个名字Teacher。

这里的=，不是表示相等，而是赋值。

你不需要知道这个数据在内存中的具体位置（地址），计算机知道的。

从现在开始，你就可以使用Teacher来引用这个数据了。

Teacher两边如果加上引号，那就不是名字了，python会当做字符串面值来处理。



## 名字

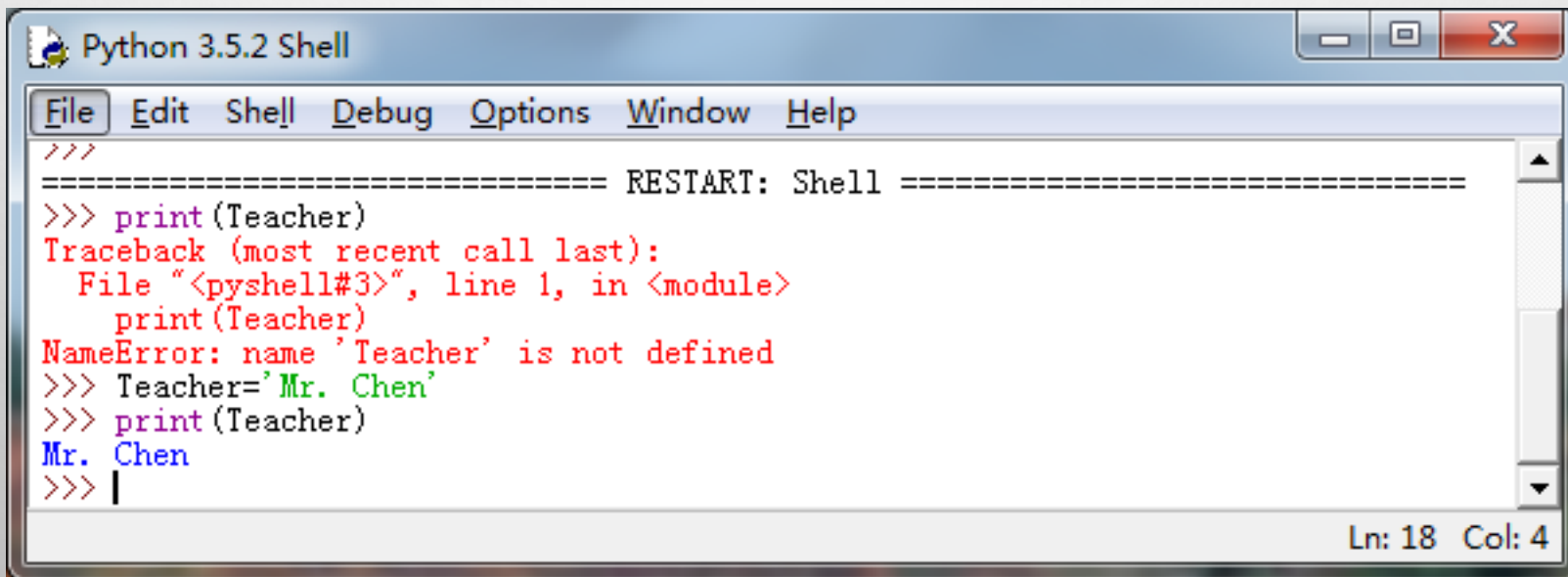
Name2.py

```
#Name2.py  
someone = 'Sam'  
print(someone, "is very smart!")  
print(someone, "is a genius!")
```

你可以修改赋值的内容

# 变量

- Python中，变量表示对于某个数据的引用
- 变量在使用前必须有定义（赋值）
- 对变量的第一次赋值，相当于对变量的定义。从此刻起，这个变量才存在于内存里。

A screenshot of a Python 3.5.2 Shell window. The window has a menu bar with 'File', 'Edit', 'Shell', 'Debug', 'Options', 'Window', and 'Help'. The main text area shows the following code and output:

```
>>>  
===== RESTART: Shell =====  
>>> print(Teacher)  
Traceback (most recent call last):  
  File "<pyshell#3>", line 1, in <module>  
    print(Teacher)  
NameError: name 'Teacher' is not defined  
>>> Teacher='Mr. Chen'  
>>> print(Teacher)  
Mr. Chen  
>>> |
```

The status bar at the bottom right indicates 'Ln: 18 Col: 4'.

## 名字的命名规则

- 名字也叫标识符，是变量、函数和其他对象的名字。
- 标识符必须以字母或下划线开头，其后的字符可以是字母、下划线或数字。
  - 使用汉字也是可以的，但考虑语言的适用和兼容性，一般不用。
  - 以下划线开头的变量在PYTHON中有特殊含义，一般不用。
  - 变量名中不能有空格以及标点符号（括号、引号、逗号、斜线、反斜线、冒号、句号、问号等等）
  - 错误示例：

99VAR

good boy

first-class



## 名字的命名规则

- 一些特殊的字符组合保留为Python关键字，不能用作变量名。  
例如：if、elif等。
- 标识符对英文字母的大小写敏感，例如student和STUDENT是不同的变量。
- 建议使用有意义的名字。例如：sum、count、average等。
- 建议使用小写字母表示变量名，多个单词时以下划线隔开。例如：  
student\_number。

# 下面的名字是合法的标识符吗？

Bill

Bill Gates

Bill-Gates

Bill\_Gates

18monks

monks18

Tom's

if

## 内置数据类型1：整数（int）

- 整数是不带小数部分的数字。
- 整数字面量，例如：25、-86、0
- 整数变量，例如：x = 1

## 内置数据类型2：浮点数（float）

- 浮点数是带小数的数字，或者是使用科学记数法的数字。
- 浮点字面量，例如：
  - -3.1      2.999    -4.0      3.      .5
  - 2.3e02（相当于230.0）      1.5e-2（相当于0.015）
- 浮点变量，例如：x = 1.5

## 内置数据类型3：字符串（str）

- 字符串是一系列字符。
- 字符包括：字母、数字、标点符号、汉字等
- 字符串字面量，例如：'CAT!'                      '021-65640000'
- 字符串字面量的界定符，可以使用单引号、双引号和三引号
  - 例如："It's OK. "                      """This is a comment"""
  - 其中三引号字符串可以跨多行
- 字符串变量，例如：s = 'abc'



## 内置数据类型4：布尔类型（bool）

- 布尔类型，也叫逻辑类型，只有两种取值：**True**（真）、**False**（假）
- 布尔类型非常重要，条件表达式和控制语句一定会用到它。

## 类型转换

- 将其它类型转换为整数，`int()`
  - 并非四舍五入
  - 也可用于将其它进制转换为十进制

如果要实现四舍五入该怎么办？

```
>>> int(1.4)
1
>>> int(1.5)
1
>>> int('123')
123
>>> int('abc')
Traceback (most recent call last):
  File "<pyshell#3>", line 1, in <module>
    int('abc')
ValueError: invalid literal for int() with base 10: 'abc'
```

## 类型转换

- 将其它类型转换为浮点类型，float()
- 将其它类型转换为字符串，str()

```
>>> float(1)
1.0
>>> float('1.5')
1.5
>>> str(1)
'1'
>>> str(1.5)
'1.5'
```

## 四大基本运算

除法的除数不能是0

加

- $3 + 5$
- $1.2 + 3.4$
- `'class' + 'room'`

减

- $5 - 3$
- $1.2 - 3.4$

乘

- $5 * 3$
- $1.2 * 3.4$
- `'Ha' * 5`

除

- $5 / 3$
- $1.2 / 3.4$

## 整数除法

- 使用 //

整数和浮点数计算的结果是浮点数

```
>>> 5 // 3
1
>>> 3.4 // 1.2
2.0
>>> 4.5 // 2
2.0
>>> 4 // 2
2
>>> 4 // 2.0
2.0
>>> 4.0 // 2
2.0
>>>
```



## 整数除法求余数

- 使用%

```
>>> 5 % 3
```

```
2
```

```
>>> 6 % 3
```

```
0
```

```
>>> 7 % 2
```

```
1
```

```
>>> x = 234
```

```
>>>
```

如何求一个三位数的百位数字、  
十位数字和个位数字？

```
>>> x = 234
```

```
>>> x // 100
```

```
2
```

```
>>> x // 10 % 10
```

```
3
```

```
>>> x % 10
```

```
4
```

## 乘方

- 使用\*\*

```
>>> 3 ** 2
```

```
9
```

```
>>> 3 ** 3
```

```
27
```

```
>>> 4 ** 0.5
```

```
2.0
```

```
>>> 2.5 ** -1
```

```
0.4
```

```
>>>
```

## 表达式

- 字面量或变量使用各种运算，形成表达式
- 例如：  $1 + 3.5 * 2$                    $4.2 * 5 ** 2$
- 不同运算混合计算时，要遵守运算的优先级



# input函数

- Input函数可以得到用户在键盘输入的字符串。
- 一般，我们总是把input的结果赋值给一个变量，来保存这次输入的内容。
- 在input之前可以先print输出提示信息。

```
#SayHi.py  
print("Enter your name:")  
name = input()  
print("Hi", name, "how are you today?")
```

```
Enter your name:  
Sam  
Hi Sam how are you today?  
>>> |
```

## input函数

- 如果希望用户键入和提示在同一行，print函数要修改。

```
#SayHi2.py  
print("Enter your name:", end=' ')  
name = input()  
print("Hi ", name, " how are you today?", sep=' ')
```

```
Enter your name: Sam  
Hi Sam how are you today?  
>>> |
```

Print函数默认的结束符(end)是换行，默认的分隔符(sep)是空格。



## input函数

- 也可以用input函数直接打印提示信息。

```
#SayHi3.py  
name = input("Enter your name:")  
print("Hi", name, "how are you today?")
```

```
Enter your name: Sam  
Hi Sam how are you today?  
>>> |
```

# input函数

- 函数结果总是得到字符串，如果需要的是数字，要使用类型转换。

```
#Square.py  
s = input("Enter a number:")  
x = int(s)  
print('Square of', x, 'is', x**2)
```

```
Enter a number:-2  
Square of -2 is 4
```

## input函数

Square2.py

- 也可以将int和input结合使用。

```
#Square2.py  
x = int(input("Enter a number:"))  
print('Square of', x, 'is', x**2)
```

```
Enter a number:-2  
Square of -2 is 4
```

# 货币兑换问题

Exchange.py

```
#Exchange.py
rate = 6.78
dollar = float(input("Dollars:$"))
rmb = dollar * rate
print('RMB:¥%.2f' % rmb, sep='')
```

```
#1美元兑换多少人民币
#请用户键入欲兑换的美元数量，转换为浮点数
#计算兑换的人民币数量
#输出小数点后2位
```

%.2f表示在字符串此处嵌入一个浮点数，保留两位小数（四舍五入）

注意：这里只是输出的数值四舍五入，**rmb**并没有四舍五入。

# 复习阅读

- 课本第2章的2.2、2.3、2.5、2.8、2.9
- 课本第3章的3.3

