

# 第一节课 基础复习

## 主讲：雪梨

data type: int, float, string, list, tuple, set, dictionary  
selection: if  
iteration: while, for  
scope  
method  
class  
ADT: stack, queue

墨  
学FIT9133 期末复习班

---

memory

墨  
学墨  
学FIT9133 期末复习班

---

Python中的对象和变量  
Objects and Variables

**Object Definition:** Pieces of memory locations in the computer that containing (holding) a specific type of data value or literal

**Variable Definition:** Each object is associated with a specific data type (or object type)

墨  
学

理解:

我们可以把python理解成一个高级的计算器，并且电脑为python提供了一个非常大存储数据的仓库。当我们给python输入指令的时候（比如说运算一个加法： $x+y=z$ ），python便会在仓库中先储存x和y的值，然后运算加法得到z并把z的值也储存在仓库里。

对于每一个需要储存在这个仓库里的值，python都把它视为一个对象（object）。

给这个对象一个名字，以便python以后想用到这个对象，这称之为变量（Variable）。

Variable 命名规律

1. 只能用大写A-Z，小写a-z，数字0-9组成，或者下划线\_组成
2. 开头不能用数字
3. 变量最好用小写，常量用大写
4. 不能用python已经存在的keyword
5. 取名要有意义!!
6. 要么用下划线取名法，要么用大小写取名法

## Definition &amp; Initialization

variableName = value -> 第一次出现叫做initialization  
 一个variable只有先被initial之后才可以在其他的地方使用  
 之后这个variable的值被更新叫做assignment

这个概念非常的重要！！

## Python中的数据类型 Data Types

Atomic type: 在存储中，只占用一个格子。

Numeric:

int, float

Boolean:

Collective type: 在存储中，只占用多个格子

String

list, tuple, set, dictionary

`type()`这个函数可以返回出来一个具体的数据类型

比如:

`a = "123" b = 123`

`c = 123.0 d = True`

```
type(a)    >>> <class 'str'>
type(b)    >>> <class 'int'>
type(c)    >>> <class 'float'>
type(d)    >>> <class 'bool'>
```

数字类型数据类型 Numeric

整数类型: `int`    浮点类型: `float`

Numeric 数据类型运算符 (arithmetic operators): `+` `-` `*` `/` `%`

注意: 因为python版本的问题, 在python2.7里整数相除如果结果不是整数, 那么除法运算结果自动向下取整, (`5 / 2 = 2`); 而在python3.6.8里, 如果两个整数相除如果结果不是整数, 则运算结果自动变为float类型(`5 / 2 = 2.5`), 如果想要向下取整则用两个除号: `//`



int / int = float or int	int // int = int
float / int = float	float // int = int
int / float = float	int // float = int
float / float = float	float // float = int

\*\* 两个星号表示次方: 5 \*\* 2 -> 5的2次方 : 25  
// 两个除号表示向下取整: 25 // 2 -> 12  
% 表示余数  
运算法则依从先乘除后加减, 有括号先算括号



布尔类型数据类型 Boolean

其实就是代表对和错, 真和假  
真(对)值: True 假(错)值: False  
Boolean 运算符号(logical operators): and or not

	a = True b = True	a = False b = True	a = True b = False	a = False b = False
a and b	True	False	False	False
a or b	True	True	True	False
not a	False	True	False	True
not b	False	False	True	True



Collective type:

在存储中，一次性占用多个格子。从0开始是重点。

String, list, tuple, set, dictionary

String(str):

String储存了一串字符，可以是任意的字符串，在python里字符串用双引号或者单引号表示。

a = "hello"    b = "world"    c = "123"

转行: "\n"    tab键: "\t"

空的字符串: ""

Index的概念:

a = "hello, world!"

'h'	'e'	'l'	'l'	'o'	','	'w'	'o'	'r'	'l'	'd'	'!'
0	1	2	3	4	5	6	7	8	9	10	11

↑

这是第0位而不是第1位

长度: 12      index: 0-11

a[0] >>> 'h'  
a[5] >>> ','

Python中数据类型做对比 Data Types Comparison  
整数或者浮点数做对比用，返回结果是一个  
Boolean。

> 大于      >= 大于等于  
< 小于      <= 小于等于  
= 等于      != 不等于

>, >=, <, <=, ==, != 叫做 relational operators

Arithmetic operators

> Relational Operators

> Logical Operators

同理，字符串也可以用 relational operators做对比。  
大小对比采用a-z顺序。



```
'a' < 'b'      >>> True
'a' < 'aa'     >>> True
'a' == 'a'     >>> True
'a' == 'b'     >>> False
'a' != 'b'     >>> True
```

这里需要注意的重点是：

Relational expressions 返回出来的是一个 boolean 类型

Relational expression 中的等于是双等于号 == 而不是单等于号

### 关于String的几个方法调用 String Methods

String.upper() -- 全部大写 String.lower() -- 全部小写

String.count(a) -- 在这个string中有几个a

print(String) -- 打印字符串 len() -- string的长度

String[index] -- 返回string中的第几个元素

Slicing:

String[x:y] -- x,y是index, 返回string中从x到y的前一位

String[:y] -- y是index, 返回string中从头到y的前一位

String[x:] -- x是index, 返回string中从x到尾

String[-1] -- 返回string中最后一位

## Input/Output

```
print("")
print(" "+" ")
print(" ", " ", int())
print("", end = " ")
```

```
a = input(" ")
b = input(int(" "))
```

Selection -> if:

```
if boolean expression:
    body
else:
    body
```

```
if boolean expression:
    body
elif boolean expression:
    Body
else:
    body
```

墨  
学

FIT9133 期末复习班

Iterative -> loop:

```
for xxx in yyyy :  
    Body
```

```
while boolean expression:  
    Body
```

```
A_list = [a for a in range(10) if a % 2 == 0]
```

墨  
学

墨  
学

FIT9133 期末复习班

List [ ] ordered, changeable, duplicate allow  
 Tuple ( ) ordered, unchangeable, duplicate allow  
 Set { } unordered, unindex, no duplicate  
 Dictionary {key:value} unordered, changeable, index

墨  
学



FIT9133 期末复习班

List []



FIT9133 期末复习班

Tuple()





FIT9133 期末复习班

set{ }



FIT9133 期末复习班

dictionary {key:value}



墨学

FIT9133 期末复习班

Operation	Result
<code>x in s</code>	True if an item of <code>s</code> is equal to <code>x</code> , else False
<code>x not in s</code>	False if an item of <code>s</code> is equal to <code>x</code> , else True
<code>s + t</code>	the concatenation of <code>s</code> and <code>t</code>
<code>s * n</code> or <code>n * s</code>	equivalent to adding <code>s</code> to itself <code>n</code> times
<code>s[i]</code>	<code>i</code> th item of <code>s</code> , origin 0
<code>s[i:j]</code>	slice of <code>s</code> from <code>i</code> to <code>j</code>
<code>s[i:j:k]</code>	slice of <code>s</code> from <code>i</code> to <code>j</code> with step <code>k</code>
<code>len(s)</code>	length of <code>s</code>
<code>min(s)</code>	smallest item of <code>s</code>
<code>max(s)</code>	largest item of <code>s</code>
<code>s.index(x[, i[, j]])</code>	index of the first occurrence of <code>x</code> in <code>s</code> (at or after index <code>i</code> and before index <code>j</code> )
<code>s.count(x)</code>	total number of occurrences of <code>x</code> in <code>s</code>

墨学

墨学

FIT9133 期末复习班

Scope:  
一个variable永远只存在于自己本身的block之中。

```
a = 1
while (a <= 9):
    b = 1
    while (b <= 9):
        print(" ", end=" ")
        print(a * b, end=" ")
        b = b + 1
    print()
    a = a + 1
```

墨学



练习：

定义一个list，list里面包含1-100

定义一个list，list里面包含1-100的所有偶数

求1-100的所有数的和

求1-2+3-4+5 ... 99的所有数的和

用户登陆（三次机会重试）



练习：What are the values of the following z?

i = 1.0 j = 2 z = i * j	
i = 1 j = 5 z = i / j	
i = 1 j = 2 z = i % j	
i = 3 j = 5 z = i > j	
i = 1 * 3 > 5 % 2 j = not 3 < 2 z = i and j	
i = True j = False z = not i and not j	





练习: Is there an error on the following code? If yes, what is the error on the following code?  
If no, what is the result of the following code?

5 <= True	
8 = 8	
True > True	
True > False	



练习: What are the values of the following s?

s = "abc"	s.upper()	
s = "abc"	s = s.upper()	
s = "aaa"	s = s.count("a")	
s = " aa "	s = s.strip()	
s = "12345"	s = s.isdigit()	





墨  
学

FIT9133 期末复习班

练习：What are the values of the following s?

s = "abc"	s.upper()	
s = "abc"	s = s.upper()	
s = "aaa"	s = s.count("a")	
s = " aa "	s = s.strip()	
s = "12345"	s = s.isdigit()	

墨  
学

墨  
学

FIT9133 期末复习班

练习：What are the values of the following s?

s = "123456"	s[0]	
s = "123456"	s[-1]	
s = "123456"	s[0:]	
s = "123456"	s[1:]	
s = "123456"	s[:-1]	
s = "123456"	s[:-2]	
s = "123456"	s[2:4]	
s = "123456"	s[::2]	
s = "123456"	s[6]	
s = "123456"	len(s)	

墨  
学

墨  
学

FIT9133 期末复习班

练习：What are the values of the following s?

s = "1 2 3 4 5 6"	s = s.split(" ")	
s = "1 2 3 4 5 6"	s = s.split(" ", 1)	
s = ["1", "2", "3"] s2 = "-"	s = s2.join(s)	
s = ("1", "2", "3") s2 = "+"	s = s2.join(s)	

墨  
学

墨  
学

FIT9133 期末复习班

练习：What are the values of the following a?

a = [1, 2, 3]	a[1]	
a = [1, 2, 3] b = [4, 5, 6]	a.append(b)	
a = [1, 2, 3] b = [4, 5, 6]	a.extend(b)	
a = [1]	a = [1] * 3	
a = [1, 2, 3]	a.insert(0, "4")	
a = ["1", "2", 3, "Four"]	a = a[1:2]	
a = ["1", "2", 3, "Four"]	a = a.reverse	
a = [4, 2, 7, 5, 2, 1]	a = a.sort()	
a = [4, 2, 7, 5, 2, 1]	3 in a2 in a	
a = [4, 2, 7, 5, 2, 1]	len(a)	

墨  
学



练习：What are the values of the following expression?

a = {0:[1,2,3,4,5], 1:[6,7,8,9,10], 3:[11,12,13,14,15,16,17]}	len(a[3])	
a = {0:[1,2,3,4,5]}	a[3] = "hello"	



练习：complete the following code:

如果今天下雨，那么我打车上学 如果今天是晴天，那么我走路上学  
如果今天是阴天，那么我骑自行车上学 如果今天下冰雹，我不去上学





练习：以下循环循环了多少次？

<pre>i = 0 s = "1234567" while (i &lt; len(s)):     i += 1</pre>	
<pre>while True:     print("hello world")</pre>	
<pre>i = 0 while True:     if(i == 100):         break     i += 1</pre>	



练习：complete the following code:

打印出来一个50 \* 50 的星星表





练习：以下循环循环了多少次？

<pre>for a in [1,2,3]:     print("hello")</pre>	
<pre>for a in [(1,2), (3,4)]:     print("hello")</pre>	
<pre>for a in "hello":     print(a)</pre>	



练习：把以下while loop转换成for loop?

<pre>i = 0 s = "1234567" while (i &lt; len(s)):     i += 1</pre>	
<pre>while True:     print("hello world")</pre>	
<pre>i = 0 while True:     if(i == 100):         break     i += 1</pre>	



墨  
学

FIT9133 期末复习班

练习：用for loop define出来一个list 里面的数字是1到100的所有偶数？

```
A_list = [a for
```

墨  
学墨  
学

FIT9133 期末复习班

Scope:

一个variable永远只存在于自己本身的block之中。

```
a = 1
while (a <= 9):
    b = 1
    while (b <= 9):
        print(" ", end=" ")
        print(a * b, end=" ")
        b = b + 1
    print()
    a = a + 1
```

墨  
学

Decomposition:

把复杂的问题分解成一些小问题。对于不同的小问题有不同的解决办法。在程序之中我们用function/methods/class/modules来代表这些小问题。

一个module里面可以有不同的class，一个class里面有不同的method/function。

Re-usability: 由于同一个method可以调用多次，那么就大大的减少了代码量。

Function/Method

Key word: def

Header: def methodName (parameters)

这里def是我们的我们关键词，说明下面的这一个block是一个method。methodName是这个method的名字。这个名字是我们自己去define的，取method的名字的时候，永远小写开头，并且是有意义的。

Parameters的个数可以是0个，也可以是1个，也可以是多个。在method的header里面，我们的parameter只是一个variable name。也可以附上一个default value。

墨  
学

FIT9133 期末复习班

```
def calculator():
    a = 1
    while (a <= 9):
        b = 1
        while (b <= 9):
            print(" ", end=" ")
            print(a * b, end=" ")
            b = b + 1
        print()
        a = a + 1

def calculator2(x,y):
    calculator()
    return x * y

def main():
    calculator()
    a = calculator2(1,2)
    print(a)
```

Calling Functions  
 methodName(real parameters)  
 这里的real parameters是给  
 parameters附上了一个真实的值。

墨  
学墨  
学

FIT9133 期末复习班

练习：写一个method，有两个parameter a 和b，  
 求a到b的所有数的和？

练习：这个method会让用户输入一行字符，分别  
 统计出其中英文字母、空格、数字和其它字符的  
 个数，把这些字符的个数print出来。

墨  
学



Lambda

只能做简单的工作 Anonymous functions, only one expression/parameter

$x = \text{lambda } a : a + 10$  -> 让a成为一个method/function

`print(x(5))`

Lambda

只能做简单的工作 Anonymous functions, only one expression/parameter

$x = \text{lambda } a : a + 10$  -> 让a成为一个method/function

`print(x(5))`

map

`map(function, iterables)`

`def myfunc(a):`

`return len(a)`

`x = map(myfunc, ('apple', 'banana', 'cherry'))`

`print(list(x))`

墨学

FIT9133 期末复习班

Main function: 程序的起始位置

```
def main():
    calculator()
    a = calculator2(1,2)
    print(a)

if __name__ == "__main__":
    main()
```

墨学

墨学

FIT9133 期末复习班

练习：以下code中哪些是global variable，哪些是local variable？有Output吗？Output是什么？如果没有Output，error是什么？

x = "global" def foo(): print("x inside :",x) foo() print("x outside:", x)		def foo(): y = "local" print(y) foo()	
x = "global" def foo(): x = x * 2 print(x) foo()		x = "global" def foo(): y = "local" x = x * 2 print(x) print(y) foo()	
x = 5 def foo(): x = 10 print("local x:", x) foo() print("global x:", x)			

墨学

## Class VS Object

在python中，类(class)是一种对象(object)的模板，对象(object)也是类(class)的实例化。Class不代表具体的事物，只是一个模板，而object才是表示具体的事物。

```
class cat  
class food
```

我们通过描述特征和行为来描述一个object，比如猫有自己的品种，颜色，大小，年龄；猫也会喵喵叫，追玩具老鼠，这些是它的行为。猫，只是用来概括猫这个动物，而当具体到自己家的猫的时候，便是指代具体的事物了。

在Python中，一个object的特征成为attribute，它所拥有的行为叫做method。

Object = attribute + method

Class: 具有相同attribute和method的object归为一个class。

## Class Header

class ClassName:      注意 Class 的名字永远大写

```
1 class Cat:
```

## Constructor

Def \_\_init\_\_(self, parameters):

```
1 class Cat:
2     def __init__(self, name):
3         self.name = name
```

创建一个新的 object 通过调用这个  
constructor

```
7 myCat = Cat("XiaoMao")
```

self

每一个属于 class 的 method 的 parameter 的第一位，都必须有一个 self。  
当外部 call 到这个 method 的时候，我们在 call 这个 method 的时候不需要加上 self。Self 本身的意思是指我这个 object 本身。

## Class Header

class ClassName:      注意 Class 的名字永远大写

```
1 class Cat:
```

## Constructor

Def \_\_init\_\_(self, parameters):

```
1 class Cat:
2     def __init__(self, name):
3         self.name = name
```

## Method

每个class可以有很多属于自己的method去自己define。

```
1 class Cat:
2     def __init__(self, name):
3         self.name = name
4     def catJump(self):
5         print("jump")
```

调用通过 variable.method()

```
8 myCat.catJump()
```

墨  
学

FIT9133 期末复习班

---

Overloading

\_\_eq\_\_

\_\_str\_\_

墨  
学