CS001编程零基础 Python语言入门

第五讲

作业回顾

第一题：中文和Python语言间的翻译

答案文件名week04solution01.py

```
a=int(input("Please input a:"))
b=int(input("Please input b:"))
if a > b:
    print(a,b)
elif a < b:
    print(b,a)
else:
    print("What a coincidence!")
```

1）利用标准输入请用户输入一个整数a

2）利用标准输入请用户输入一个整数b

3）如果a大于b，那么输出两个整数a b

4）如果a小于b，那么输出两个整数b a

5）如果a等于b，那么输出字符串"What a coincidence!"

第二题：圆周率计算

$$\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \frac{1}{11} + \cdots$$

答案文件名pi.py

```
num=0
factor=1
for i in range(10000000):
    num = num + factor/(i+i+1)
    factor=-factor
print(num*4)
```

```
>>> import math
>>> math.pi
3.141592653589793
>>> math.e
2.718281828459045
>>> math.inf
inf
>>>
```

附加题：生成彩票中奖号码

答案文件名lottery.py

```
import random

seeds=list(range(1,34))
red=[]
for i in range(6):
    x=random.randrange(0,33-i)
    red.append(seeds[x])
    seeds.remove(seeds[x])

blue=random.randint(1,16)

print("RED: ",red,"\nBLUE:",blue)
```

引入随机工具模块

为了确保六个随机数不能重复，使用seeds种子列表

初始时，seeds列表里有1-33所有数字
为了生成6个红色球数字，共循环6次：
每次从seeds列表中随机取走一个数
取走的数字会从seeds列表移除

随机生成一个1-16的数字赋值给blue变量

输出结果

条件循环语句：while

while 语法说明

查看while 语法的方法

1. IDLE 中输入help()进入帮助界面

2. help>提示符后输入while查看while语句说明文档

3. help>提示符后输入quit退出帮助界面

```
help> while
The "while" statement
*********************

The "while" statement is used for repeated execution as long as an
expression is true:

   while_stmt ::= "while" expression ":" suite
                  ["else" ":" suite]

This repeatedly tests the expression and, if it is true, executes the
first suite; if the expression is false (which may be the first time
it is tested) the suite of the "else" clause, if present, is executed
and the loop terminates.

A "break" statement executed in the first suite terminates the loop
without executing the "else" clause's suite.  A "continue" statement
executed in the first suite skips the rest of the suite and goes back
to testing the expression.

Related help topics: break, continue, if, TRUTHVALUE

help>
```

while语法练习

无限循环/死循环的演示

```
>>> while True:
        print("I will never stop.")
```

以上循环语句可以翻译成如下中文

当 True这个条件成立时，反复执行以下命令：

输出"I will never stop."

```
>>> while True:
        print("I will never stop.")

I will never stop.
I will never stop.
I will never stop.
I will never stop.
I will never stop.
I will never stop.
I will never stop.
I will never stop.
I will never stop.
I will never stop.
I will never stop.
I will never stop.
I will never stop.
I will never stop.
I will never stop.
I will never stop.
I will never stop.
I will never stop.
I will never stop.
I will never stop.
I will never stop.
I will never stop.
I will never stop.
I will never stop.
I will never stop.
I will never stop.
I will never stop.
I will never stop.
I will never stop.
```

while语法练习

用while语法实现for循环的功能：固定次数循环

```
>>> x=0
>>> while x<10:
        print(x)
        x=x+1
```

以上循环语句可以翻译成如下中文

初始化把x赋值为0
当 x<10这个条件成立时，反复执行以下命令：
    输出x
    使x加1

```
>>> x=0
>>> while x<10:
        print(x)
        x=x+1
0
1
2
3
4
5
6
7
8
9
>>>
```

效果等同于如下for循环

```
>>> for x in range(10):
        print(x)
```

while语法练习

小心死循环！！

```
>>> x=0
>>> while x<10:
        print(x)
```

这段程序会发生死循环
因为x没有变化
x<10的条件永远满足

```
>>> x=0
>>> while x<10:
        print(x)
        x=x-1
```

这段程序会发生死循环
因为x每次都减1
x<10的条件永远满足

while语法举例：简易版猜数字游戏

程序文件名easy-guess.py

随机生成标准答案保存在x变量里

```
import random
x=random.randint(0,9)
CORRECT=False
while not CORRECT:
    guess=int(input("[0,9]What is your guess? "))
    if guess<x:
        print("Too small!")
    elif guess>x:
        print("Too big!")
    else:
        print("Smart kid!")
        CORRECT=True
```

初始化时，让CORRECT变量等于False

当CORRECT变量不是True就反复循环

用户输入猜的数字保存在guess变量里

判断猜的数字和标准答案是不是一致。如果猜中了，那么使CORRECT等于True

while语法举例：电脑随机猜数字

程序文件名random-guess.py

随机生成标准答案保存在x变量里

```
import random
x=random.randint(0,9)
CORRECT=False
while not CORRECT:
    guess=random.randint(0,9)
    print("The robot's guess is",guess)
    if guess<x:
        print("Too small!")
    elif guess>x:
        print("Too big!")
    else:
        print("Smart robot!")
        CORRECT=True
```

初始化时，让CORRECT变量等于False

当CORRECT变量不是True就反复循环

用户输入猜的数字保存在guess变量里

判断猜的数字和标准答案是不是一致。如果猜中了，那么使CORRECT等于True

while语法和for语法的关系

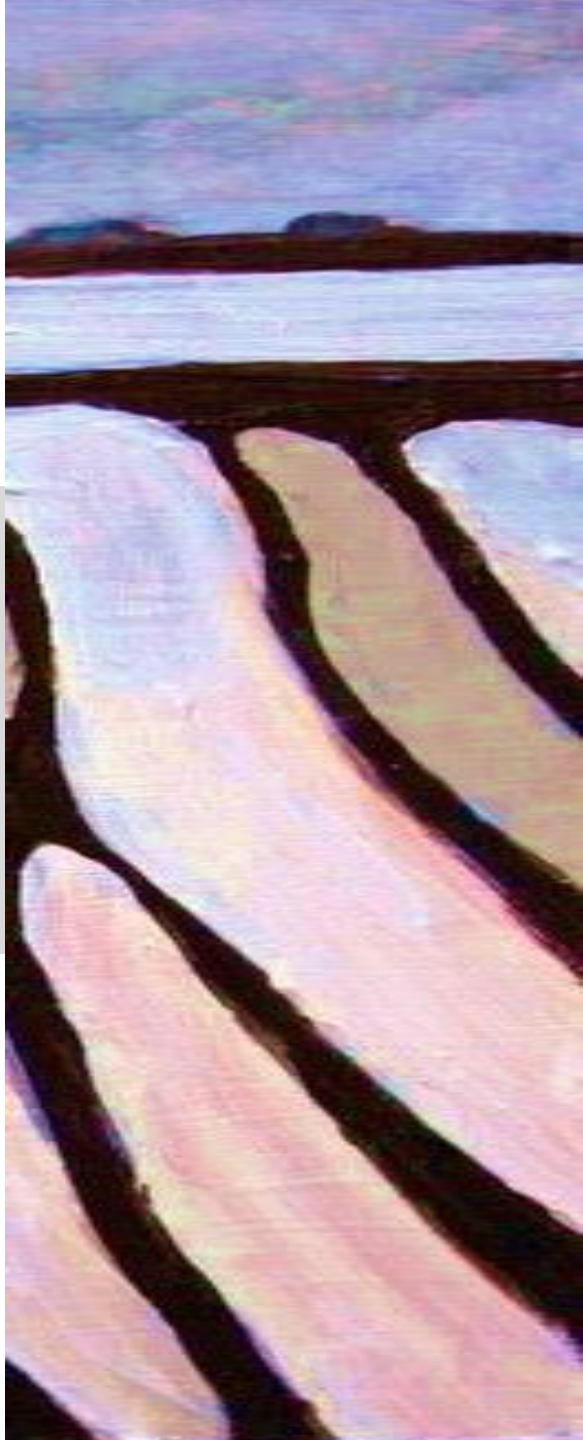
```
>>> for i in range(10):
        print(i)
```

```
>>> i=0
>>> while i<10:
        print(i)
        i=i+1
```

两段程序都是通过循环的方式输出0-9这十个数字

讨论题：在哪些情况下应该使用for语句，在哪些情况下应该使用while语句？

定义新函数/功能：def



Python内置函数功能

The Python interpreter has a number of functions and types built into it that are always available. They are listed here in alphabetical order.

除了Python自带的内置函数/功能以外，我们还可以通过def语法自定义新的函数/功能

		Built-in Functions		
abs()	dict()	help()	min()	setattr()
all()	dir()	hex()	next()	slice()
any()	divmod()	id()	object()	sorted()
ascii()	enumerate()	input()	oct()	staticmethod()
bin()	eval()	int()	open()	str()
bool()	exec()	isinstance()	ord()	sum()
bytearray()	filter()	issubclass()	pow()	super()
bytes()	float()	iter()	print()	tuple()
callable()	format()	len()	property()	type()
chr()	frozenset()	list()	range()	vars()
classmethod()	getattr()	locals()	repr()	zip()
compile()	globals()	map()	reversed()	__import__()
complex()	hasattr()	max()	round()	
delattr()	hash()	memoryview()	set()	

def语法说明

查看def语法的方法

1. IDLE中输入help()进入帮助界面

2. help>提示符后输入def查看def语句说明文档

3. help>提示符后输入quit退出帮助界面

```
help> def
Function definitions
********************

A function definition defines a user-defined function object (see
section The standard type hierarchy):

   funcdef                 ::= [decorators] "def" funcname "(" [parameter_list]
")" ["->" expression] ":" suite
   decorators              ::= decorator+
   decorator               ::= "@" dotted_name ["(" [argument_list [","]] ")"] N
EWLINE
   dotted_name             ::= identifier ("." identifier)*
   parameter_list          ::= defparameter ("," defparameter)* ["," [parameter_
list_starargs]]
                             | parameter_list_starargs
   parameter_list_starargs ::= "*" [parameter] ("," defparameter)* ["," ["**" pa
rameter [","]]]
                               | "**" parameter [","]
   parameter               ::= identifier [":" expression]
   defparameter            ::= parameter ["=" expression]
   funcname                ::= identifier

A function definition is an executable statement.  Its execution binds
the function name in the current local namespace to a function object
(a wrapper around the executable code for the function).  This
function object contains a reference to the current global namespace
as the global namespace to be used when the function is called.

The function definition does not execute the function body; this gets
executed only when the function is called. [3]

A function definition may be wrapped by one or more *decorator*
expressions. Decorator expressions are evaluated when the function is
                                the function definition.  The
                                invoked with the function object
```

A function definition defines a user-defined function object

用def语法定义新函数/新功能

定义一个新功能hello() 方便打招呼

```
>>> def hello(name):
        print("Hello",name,"long time no see.")

>>> hello("Mike")
Hello Mike long time no see.
>>> hello("Alice")
Hello Alice long time no see.
>>> hello("Superman")
Hello Superman long time no see.
```

定义一个新功能praise() 方便赞美

```
>>> def praise(name1,name2):
        print(name1,“的颜值爆表啦!")
        print(name2,"的智商上天啦!")

>>> praise ("Mike","Nana")
Mike 的颜值爆表啦!
Nana 的智商上天啦!
>>> praise ("妈妈","爸爸")
妈妈 的颜值爆表啦!
爸爸 的智商上天啦!
```

def语法细节

定义一个新函数功能praise() 方便赞美

新函数的申明需要用def语法开始申明

在def后面写上新函数的名称，例如praise

括号内的变量作为函数的输入信息。此例子中包含两个名字的字符串

```
def praise(name1,name2):
        print(name1,“的颜值爆表啦!")
        print(name2,"的智商上天啦!")

praise ("Mike","Nana")
praise ("妈妈","爸爸")
```

别忘了冒号哦！

该函数具体操作定义为打印两句话

对于已经定义过的函数，就可以直接使用了

用def语法+return语法增加返回值

定义一个新函数average() 返回三个数的平均数

```
>>> def average(a,b,c):
        return (a+b+c)/3

>>> average(1,2,3)
2.0
>>> average(10,100,1000)
370.0
>>> x=average(5,6,7)+average(8,88,888)
>>> print(x)
334.0
```

def语法细节：增加函数返回值

定义一个新函数average() 返回三个数的平均数

新函数的申明需要用def语法开始申明

在def后面写上新函数的名称，例如average

括号内的变量作为函数的输入信息。此例子中包含三个数字

```
def average(a,b,c):
        return (a+b+c)/3

x=average(5,6,7)+average(8,88,888)
print(x)
```

别忘了冒号哦！

该函数具体操作定义为计算三个数的平均数，然后返回这个数值

对于已经定义过的函数，就可以直接使用了

给函数起一个新名字：函数的赋值语句

第一步：定义一个新函数average() 返回三个数的平均数

第二步：把average赋值给mean, 比较使用的效果

```
>>> def average(a,b,c):
        return (a+b+c)/3
>>> mean=average
>>> mean(1,2,3)
2.0
>>> mean(10,100,1000)
370.0
>>> average(10,100,1000)
370.0
>>> if mean==average:
        print("same")
same
```

这句赋值语句给函数起了新名字

功能并没有变化

def函数定义举例：汇率转换

文件名RMBtoUSD.py

```
def RMBtoUSD(rmb):
    return rmb*6.8957

x=float(input("How much RMB do you have? "))
print(RMBtoUSD(x))
```

def函数定义举例：摄氏温度转换为华氏度

文件名fahrenheit.py

```
def fahrenheit(c):
    return 1.8*c+32

c=float(input("What is the Celsius degree? "))
print(fahrenheit(c))
```

def函数定义举例：计数函数

文件名countodd.py

```
def countodd(a):
    count=0
    for x in a:
        if x % 2==1:
            count=count+1
    return count

t=[1,2,3,4,5,11,12,13,14,15]
print(countodd(t))
t.remove(5)
print(countodd(t))
t.remove(3)
print(countodd(t))
t=t+[9,9,9]
print(countodd(t))
```

例题: 超市打折问题：if-else语句回顾

派尚超市举行打折促销，皮皮虾每斤30元。
如果一次性购买大于等于10斤皮皮虾，每斤只需要25元。
请写程序用标准输入让用户输入购买多少斤，标准输出为总费用为多少元。

输入样例：8
输出样例：240

输入样例：20
输出样例：500

输入样例：12
输出样例：300

输入样例：1
输出样例：30

超市打折问题答案 discount.py

```
#discount.py
n=int(input("How much mantis shrimp? "))
if n<10:
    money=n*30
else:
    money=n*25
print("The total payment is",money,"RMB.")
```

例题: 超市打折问题　利用函数改进程序

派尚超市举行打折促销，皮皮虾每斤30元。
如果一次性购买大于等于10斤皮皮虾，每斤只需要25元。
请写程序用标准输入让用户输入购买多少斤，标准输出为总费用为多少元。

输入样例: 8
输出样例: 240

输入样例: 20
输出样例: 500

输入样例: 12
输出样例: 300

输入样例: 1
输出样例: 30

超市打折问题答案 discount.py

```
#discount2.py
def money(n):
    if n<10:
        return n*30
    else:
        return n*25

n=int(input("How much mantis shrimp? "))
print("The total payment is",money(n),"RMB.")
```

回顾： 猜数字游戏II 高级版

文件名guess-game.py

```
import random
#生成4个随机的不重复数字
seeds=list(range(10))
answer=""
for i in range(4):
    x=random.randrange(0,10-i)
    answer=answer+str(seeds[x])
    seeds.remove(seeds[x])
print("Hi! We have generated a four-digit number. You have 10 chances to guess.")
for i in range(10):
    x=input("What is your guess #"+str(i+1)+"? ")
    A,B=0,0
    for j in range(4):
        if x[j]==answer[j]:
            A=A+1
        elif x[j] in answer:
            B=B+1
    print(A,"A",B,"B")
    if A==4:
        print("Bingo! You are a genius!")
        break
if A!=4:
    print("Sorry, the answer is"+str(answer))
```

游戏演示结果

```
Hi! We have generated a four-digit number.
You have 10 chances to guess.
What is your guess #1? 1234
0 A 1 B
What is your guess #2? 5678
0 A 2 B
What is your guess #3? 2349
0 A 2 B
What is your guess #4? 9340
1 A 0 B
What is your guess #5? 9562
1 A 2 B
What is your guess #6? 9827
3 A 0 B
What is your guess #7? 9826
3 A 0 B
What is your guess #8? 9825
4 A 0 B
Bingo! You are a genius!
>>>
```

用函数改进： 猜数字游戏II 高级版

文件名guess-game2.py

```
import random

def randomanswer(num):
#生成num个随机的不重复数字
    seeds=list(range(10))
    answer=""
    for i in range(num):
        x=random.randrange(0,10-i)
        answer=answer+str(seeds[x])
        seeds.remove(seeds[x])
    return answer

def countAB(x,y):
#统计答对情况
    countA,countB=0,0
    for j in range(4):
        if x[j]==y[j]:
            countA=countA+1
        elif x[j] in y:
            countB=countB+1
    print(countA,"A",countB,"B")
    return countA,countB
```

```
#主程序

y=randomanswer(4)
print("Hi! We have generated a four-digit number. You have 10 chances to guess.")
for i in range(10):
    x=input("What is your guess #"+str(i+1)+"? ")
    A,B=countAB(x,y)
    if A==4:
        print("Bingo! You are a genius!")
        break
if A!=4:
    print("Sorry, the answer is "+str(y))
```

回顾：质数判定

文件名prime1.py

```
n=int(input("Please input a number: "))
PRIME = True
for i in range(2,n):
    if n % i == 0:
        PRIME=False
        print(n,"=",i,"*",n//i)
        break
if PRIME:
    print(n,"is a prime number.")
```

标准输入：n为待判定整数

通过循环来寻找n的质因数
枚举所有的可能的i
判断n是不是i的倍数

如果能找到n的质因数i
那么n就不是质数

如果不能找到n的质因数
那么n就是质数

思考题: 质数判定函数

第一步：请定义一个函数isPrime()，这个函数的输入参数为一个正整数n，程序能通过这个函数的返回值判断n是不是质数。

第二步：请利用这个isPrime()函数，写程序可以自动判断输入的数是否为质数。用标准输入让用户输入一个正整数，判断结果使用print(isPrime(n)) 作为输出语句。

输入样例：8
输出样例：False

输入样例：1
输出样例：False

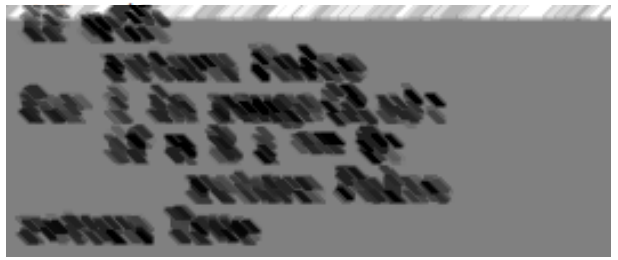
输入样例：2
输出样例：True

输入样例：37
输出样例：True

提示：使用第四周质数判断的程序，进行一些修改后，可以定义成isPrime()函数。程序框架如右图所示

File Edit Format Run Options Window Help

```
#isPrime.py
def isPrime(n):
```



```
n=int(input("Please input a number: "))
print(isPrime(n))
```