

实验二 Python基础

ANDYWWW

LAST MODIFIED: 14/3/2019

碰到不会用的命令怎么办

- ① 在交互式界面输入help(函数名) ② 百度大法好

```
In [1]: help(print)
Help on built-in function print in module builtins:
```

```
print(...)
    print(value, ..., sep=' ', end='\n', file=sys.stdout, flush=False)
```

```
Prints the values to a stream, or to sys.stdout by default.
Optional keyword arguments:
file:  a file-like object (stream); defaults to the current sys.stdout.
sep:   string inserted between values, default a space.
end:   string appended after the last value, default a newline.
flush: whether to forcibly flush the stream.
```

函数语法

自变量含义

初步介绍“函数”

数学中的函数：一元函数 $f(x) = x + 1$ ； 二元函数 $f(x, y) = x + y$

一般而言一个函数是作用在实数域的“操作符”。

推广一下，我们可以让函数作用在其他类型的自变量上，以完成同一类操作。

例如：实现将一个字符串一个一个字符print的函数 $f(\text{str})$

```
def f(str):  
    for i in str:  
        print(i)
```

调用

```
In [3]: f('我是大笨蛋')  
我  
是  
大  
笨  
蛋
```

初步介绍“函数”

例1. 实现某一操作的函数;

def f(水果):
 切水果

函数f(x)的作用
为切x

则f(苹果)就是
切苹果

def f(水果, 炸弹):
 切水果
 扔炸弹

函数f(x, y)的作
用为切x扔y

则f(苹果, 香蕉)
就是切苹果扔
香蕉

初步介绍“函数”

```
def f(str):  
    for i in str:  
        print(i)
```

第1行的str是为了后续表示而出现的，作用跟x相似

调用

```
In [3]: f('我是大笨蛋')
```

我
是
大
笨
蛋

```
for i in '我是大笨蛋':  
    print(i)
```

函数可以有运算结果，上文的例子只是实现一类功能，因此没有返回值，如果有返回值，需要利用a=f(...)将运算结果赋值给a。

初步介绍“函数”

例2. 实现某一操作并返回值的函数;

```
def f(水果):  
    切水果  
    return 水果片
```

函数f(x)的作用
为切x, 返回值
为x片

则a = f(苹果)就
是切苹果, a代
表苹果片

```
def f(水果 = 苹果):  
    try:  
        切水果  
        return 0  
    except 爆炸:  
        return -1
```

如果明确了要
切x, 如果切成
功返回0, 如果
切炸了, 返回-1;
如果没明确,
则尝试切苹果

则a = f(炸弹)代
表切炸弹, 如
果切成功a = 0;
如果炸弹爆炸a
= -1

初步介绍“函数”

使用的input、print都是函数，其中print实现一个操作，没有返回值；input也是实现一个操作，但是有返回值。

使用函数时，注意：

- ① 形参（相当于x）的数值类型、默认值。
- ② 返回值的类型（如果有返回值的话）。

初步介绍“函数”

```
In [1]: help(print)
Help on built-in function print in module builtins:
```

带有默认值的自变量

```
print(...)
print(value, ..., sep=' ', end='\n', file=sys.stdout, flush=False)
```

自变量（形参）

```
Prints the values to a stream, or to sys.stdout by default.
Optional keyword arguments:
file: a file-like object (stream); defaults to the current sys.stdout.
sep:  string inserted between values, default a space.
end:  string appended after the last value, default a newline.
flush: whether to forcibly flush the stream.
```

常用的变量就是value、sep、end；这个print函数的作用为：在file中(默认为屏幕)打印value们的值，其中各个value之间用sep(默认为' ')分隔，输出以end(默认为换行符)结束。

1. 编写程序，输入一个9位的长整数，将其分解为三个三位的基本整数并输出。

方法I: 数学方法（类似问题用循环语句后更好处理）

例：对1234而言，求其后两位的数字，

$$1234 \div 100 = 12 \dots 34$$

余数34即后两位，商12即剩下的数字，重复此步骤即可。

方法I:

```
new_num = int(input("Please input the number:\n"))
a_1 = int(new_num % 1000)
new_num = (new_num - a_1) / 1000
a_2 = int(new_num % 1000)
new_num = (new_num - a_2) / 1000
a_3 = int(new_num)
print(a_3, ' ', a_2, ' ', a_1)
```

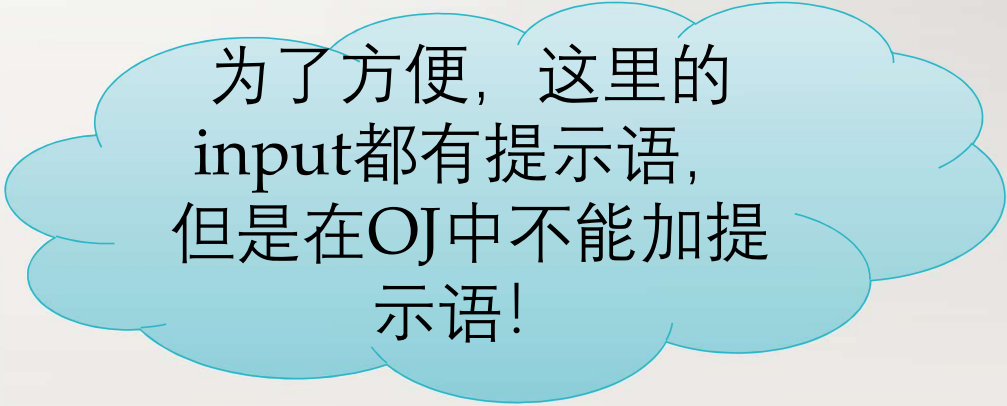
方法I: 优化

```
new_num = int(input("Please input the number:\n"))
a_1 = int(new_num % 1000)
new_num //= 1000
a_2 = int(new_num % 1000)
new_num //= 1000
a_3 = int(new_num)
print(a_3, ' ', a_2, ' ', a_1)
```

1. 编写程序，输入一个9位的长整数，将其分解为三个三位的基本整数并输出。

方法II：利用String进行切片

```
num = int(input("Please input the number:\n"))
num_str = str(num)
b_1 = int(num_str[0:3])
b_2 = int(num_str[3:6])
b_3 = int(num_str[6:])
print(b_1, ' ', b_2, ' ', b_3)
```



为了方便，这里的
input都有提示语，
但是在OJ中不能加提示
语！

2. 编写程序，从键盘输入一个4位正整数，输出该数的反序数。反序数即原数各位上的数字颠倒次序形成的另一个整数

方法I：利用String切片

```
num = input("Please enter the number:\n")
num_ans = int(num[::-1])
print(num_ans)
```

OJ要求整型输出，如果不在OJ上做这里可以不转成int

方法II：直接输出

```
num = input("Please enter the number:\n")
num_new = num[3] + num[2] + num[1] + num[0]
num_new = int(num_new)
print(num_new)
```

2. 编写程序，从键盘输入一个4位正整数，输出该数的反序数。反序数即原数各位上的数字颠倒次序形成的另一个整数

方法III：数学方法

```
num = input()
rev_num = 0
a = num % 10
rev_num = (rev_num + a) * 10
num //= 10
a = num % 10
rev_num = (rev_num + a) * 10
num //= 10
a = num % 10
rev_num = (rev_num + a) * 10
num //= 10
```

```
a = num % 10
rev_num += a
print(int(rev_num))
```

注：循环语句

```
num = input()
rev_num = 0
while(num > 0):
    rev_num *= 10
    a = num % 10
    rev_num += a
    num //= 10
print(int(rev_num))
```

注意：输出的应该是一个完整的数字，而不是一个个分隔开的倒序的数。

```
num='1234'  
print(num[3],num[2],num[1],num[0])
```

output:

4 3 2 1



```
num='1234'  
print(num[3],num[2],num[1],num[0],sep='')
```

- 注：
- ① 这里为了方便省略了input语句，直接定义num='1234'。
 - ② 更改后虽然也是一个个print字符，但是直观上输出的是一个四位整数。

因为OJ要求整型输出，所以在OJ中后一种办法也是错误的

3. 编写程序，输入分钟数，输出这些分钟代表多少年零多少天零多少小时零多少分钟。为了简化问题，假设一年有365天。

参考答案：

```
time = int(input("Please input the minutes:\n"))
```

```
all_minutes = time
```

```
a_year = 60*24*365
```

```
a_day = 60*24
```

```
an_hour = 60
```

```
year = time // a_year
```

```
all_minutes -= year * a_year
```

```
day = all_minutes // a_day
```

```
all_minutes -= day * a_day
```

```
hour = all_minutes // an_hour
```

```
minute = all_minutes - hour * an_hour
```

```
print(time, '分钟代表', year, '年', day, '天', hour, '小时', minute, '分钟')
```

为了增加程序可读性，
可以对比一下如下代码。

```
time=int(input("Please input the minutes:\n"))
all_minutes=time
year=time//525600
all_minutes-=year*525600
day=all_minutes//1440
all_minutes-=day*1440
hour=all_minutes//60
minute=all_minutes-hour*60
print(time, '分钟代表', year, '年', day, '天', hour, '小时', minute, '分钟')
```

这样可以得到正确结果，但是可读性降低

3. 编写程序，输入分钟数，输出这些分钟代表多少年零多少天零多少小时零多少分钟。为了简化问题，假设一年有365天。

参考答案：

```
time = int(input("Please input the minutes:\n"))
```

```
all_minutes = time
```

```
a_year = 60*24*365
```

```
a_day = 60*24
```

```
an_hour = 60
```

```
year = time // a_year
```

```
all_minutes -= year * a_year
```

```
day = all_minutes // a_day
```

```
all_minutes -= day * a_day
```

```
hour = all_minutes // an_hour
```

```
minute = all_minutes - hour * an_hour
```

```
print(time, '分钟代表', year, '年', day, '天', hour, '小时', minute, '分钟')
```

在print函数里，尽量避免复杂表达式

```
a=12;b=13;c=14  
print((a+b)*c/2+b*c,a-b**2+c**3)
```

```
a=12;b=13;c=14  
d=(a+b)*c/2+b*c  
e=a-b**2+c**3  
print(d,e)
```

4. 编写程序，输入水的重量（以千克为单位）以及水的初始温度和最终温度，计算并输出将水从初始温度加热到最终温度所需的能量。

参考答案：

```
t_start = float(input("Please enter the initial temperature:\n"))
t_end = float(input("please enter the final temperature:\n"))
m = float(input("Please enter the mass of the water:\n"))
Q = m * (t_end - t_start) * 4184.0
print('We need',Q,'Joule energy.')
```

注意：根据条件温度、质量可能不是整数，因此用float而不用int。

输入部分用OJ的提示代替即可，学习完列表后可以再理解一下OJ给的提示