课程名称 python实验

实验成绩

指导教师



实 验 报 告

院系 工学部

班级 软件工程B182   
学号 1801120080

姓名 常自昂

日期 2020/10/17

# Python语言实验报告



## 实验二: python的基本数据类型

## 一 实验目的

1) 熟悉type()的用法

2) 锻炼从已有代码改出来新的代码的能力

3) 熟悉字符串操作

## 二 实验环境

计算机：CPU四核i7 6700处理器；内存8G； SATA硬盘2TB硬盘； Intel芯片组

主板；集成声卡、千兆网卡、显卡； 20寸液晶显示器。

环境：windows7 64位、python3.5.3、pycharm教育版

## 三 实验内容

### 3.1 实验内容一

#### 3.1.1实验题目

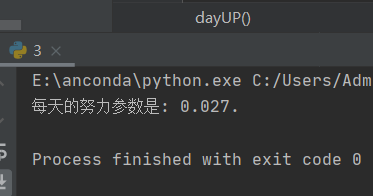
修改天天向上例子(课本76-77页)计算每周工作4天,周五,周六,周日休息,如果想要一年内能力达到原有能力的50倍,每个工作日需要进步的系数.

体会函数的封装思路

#### 3.1.2实验步骤

def dayUP(df):  
 dayup = 1.0  
 for i in range(365):  
 if i % 7 in [5,6,0]:  
 dayup = dayup \* (1 - 0.01)  
 else:  
 dayup = dayup \* (1 + df)  
 return dayup  
dayfactor = 0.01  
while (dayUP(dayfactor)<50):  
 dayfactor += 0.001  
print("每天的努力参数是: %.3f."%dayfactor)

#### 3.1.3实验结果(截图)

·

#### 3.1.4实验结论

### 3.2 实验内容二

#### 3.2.1实验题目

从书上91页示例改造出来一个如下形式的文本进度条:

----执行开始----

100%[$$$$$$$$$$$$$$$$$$$$$$$$$->]19.81s

----执行结束----

注意:

1是25个$

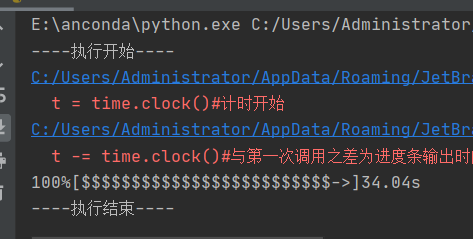
2 最终时间不一定是19.81s

要求:写出来每一行的注释,解释当前行代码在做什么

#### 3.2.2实验步骤

import time#引入time库  
scale = 25#输出进度条精度  
print("执行开始".center(scale//2,'-'))#输出“----执行开始----”  
t = time.clock()#计时开始  
for i in range(scale+1):#i的值为从0到25，每次循环加1  
 a = '$' \* i #每次输出i个$  
 b = '.' \* (scale - i)#$每多一个，‘.’减少一个  
 c = (i/scale)\*100#每次输出的$所占进度条百分比  
 t -= time.clock()#与第一次调用之差为进度条输出时间  
 print("\r{:^3.0f}%[{}->{}]{:.2f}s".format(c,a,b,-t),end='')  
 time.sleep(1)#将当前程序挂起0.1秒，  
print("\n"+"执行结束".center(scale//2,'-'))

#### 3.2.3 实验结果(截图)



#### 3.2.4 实验结论

实验本身理解课本并不难，但是在参考课本程序时，由于解释器版本不一样，clock函数报警，还没有找到替代它的方法

### 3.3 实验内容三

#### 3.3.1实验题目

实现一个回文判断程序，写出代码,要求能同时处理字符串和数字,连续输入3次之后程序退出

#### 3.3.2实验步骤

for i in range(3):  
 a=input("输入:")  
 b=a[::-1]  
 i+=1  
 if a==b:  
 print("{}".format(b))  
 else:  
 print("{}不是回文数".format(a))

#### 3.3.3 实验结果(截图)



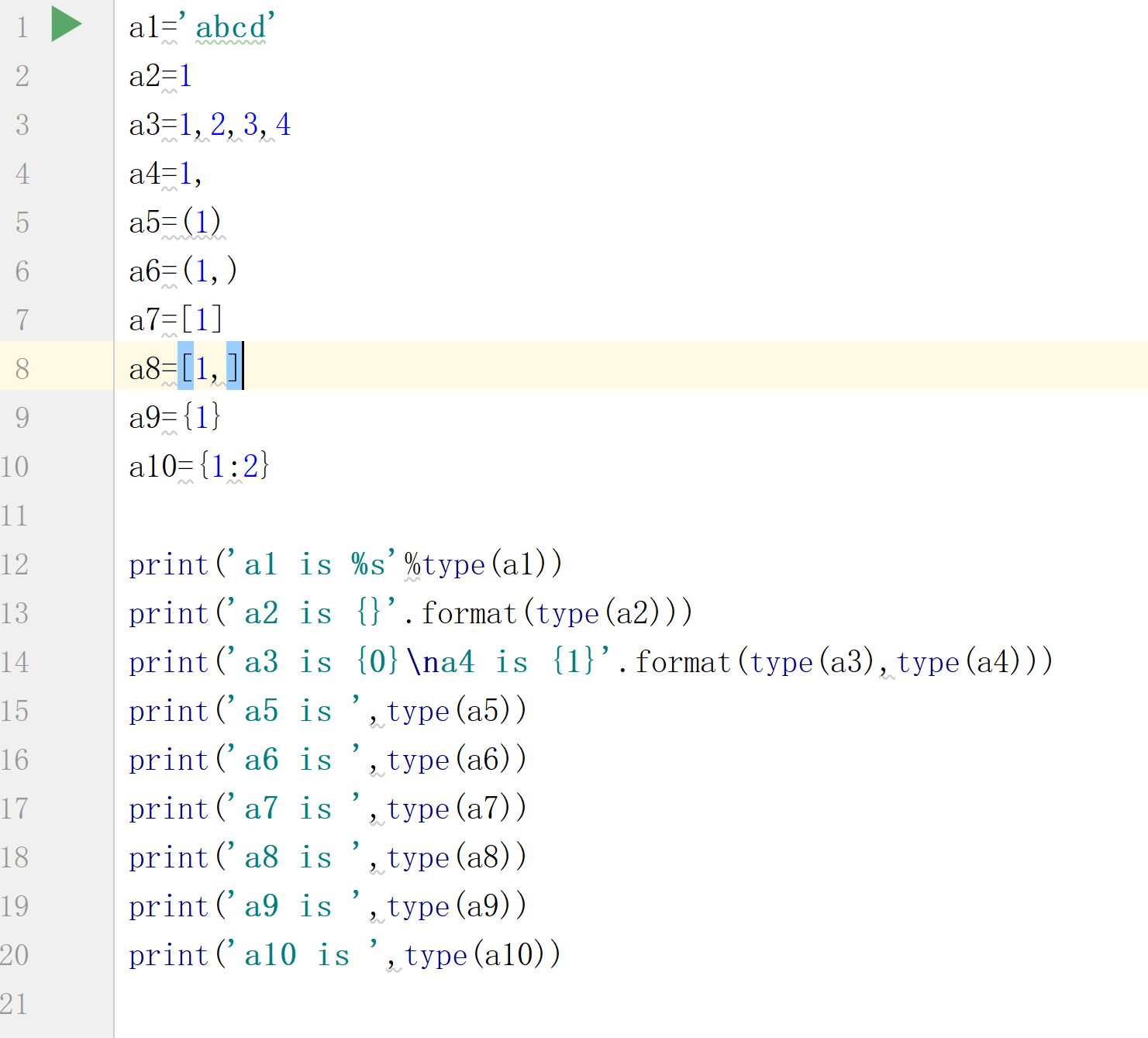
#### 3.3.4 实验结论

主要用format函数

### 3.4 实验内容四

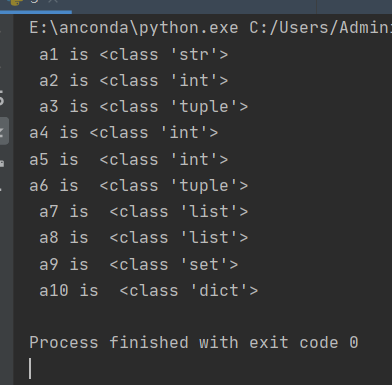
#### 3.4.1实验题目

### 写出下面程序的运行结果



#### 3.4.2实验步骤

#### 3.4.3 实验结果(截图)



#### 3.4.4 实验结论

Type函数可以返回类型

## 四 实验感想

本次实验对之前的学习进行了回顾，很多以为会的知识到了实际编程序的时候就无从下手，还是应该多练习。