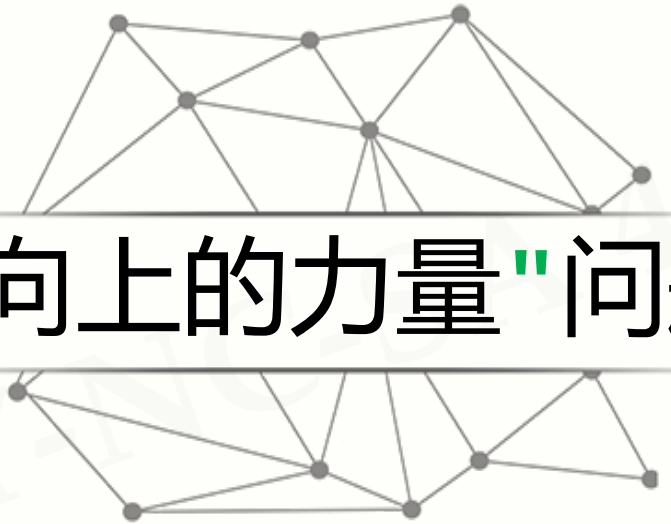

python
import turtle
turtle.setup(650,350,200,200)
turtle.penup()
turtle.fd(-250)
turtle.pendown()
turtle.pensize(25)
turtle.pencolor("purple")
turtle.seth(-40)
for 1 in range(4):
turtle.circle(40, 80)
turtle.circle(-40, 80)
turtle.circle(40, 80/2)
turtle.fd(40)
turtle.circle(16, 180)
turtle.fd(40 * 2/3)

Python语言程序设计

实例3: 天天向上的力量

嵩 天
北京理工大学



"天天向上的力量"问题分析

天天向上的力量

基本问题：持续的价值

- 一年365天，每天进步1%，累计进步多少呢?

$$1.01^{365}$$

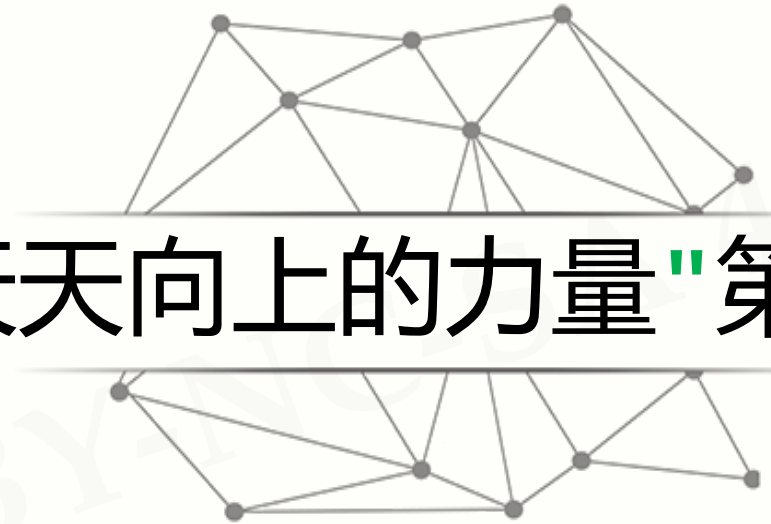
- 一年365天，每天退步1%，累计剩下多少呢?

$$0.99^{365}$$

需求分析

天天向上的力量

- 数学公式可以求解，似乎没必要用程序
- 如果是"三天打鱼两天晒网"呢?
- 如果是"双休日又不退步"呢?



"天天向上的力量"第一问

天天向上的力量

问题1： 1‰的力量

- 一年365天，每天进步1‰，累计进步多少呢?

$$1.001^{365}$$

- 一年365天，每天退步1‰，累计剩下多少呢?

$$0.999^{365}$$

天天向上的力量

问题1： 1‰的力量

```
#DayDayUpQ1.py
dayup = pow(1.001, 365)
daydown = pow(0.999, 365)
print("向上：{:.2f}，向下：{:.2f}".format(dayup, daydown))
```

编写上述代码，并保存为DayDayUpQ1.py文件

天天向上的力量

问题1： 1‰的力量

>>>（运行结果）

向上：1.44，向下：0.69

$1.001^{365} = 1.44$ $0.999^{365} = 0.69$

1‰的力量，接近2倍，不可小觑哦

"天天向上的力量"第二问

天天向上的力量

问题2： 5‰和1%的力量

- 一年365天，每天进步5‰或1%，累计进步多少呢?

1.005^{365} 1.01^{365}

- 一年365天，每天退步5‰或1%，累计剩下多少呢?

0.995^{365} 0.99^{365}

天天向上的力量

问题2： 5‰和1%的力量

```
#DayDayUpQ2.py
dayfactor = 0.005
dayup = pow(1+dayfactor, 365)
daydown = pow(1-dayfactor, 365)
print("向上：{:.2f}，向下：{:.2f}".format(dayup, daydown))
```

使用变量的好处：一处修改即可

编写上述代码，并保存为DayDayUpQ2.py文件

天天向上的力量

问题2： 5‰和1%的力量

>>>（5‰运行结果）

向上：6.17，向下：0.16

$$1.005^{365} = 6.17$$

$$0.995^{365} = 0.16$$

5‰的力量，惊讶！

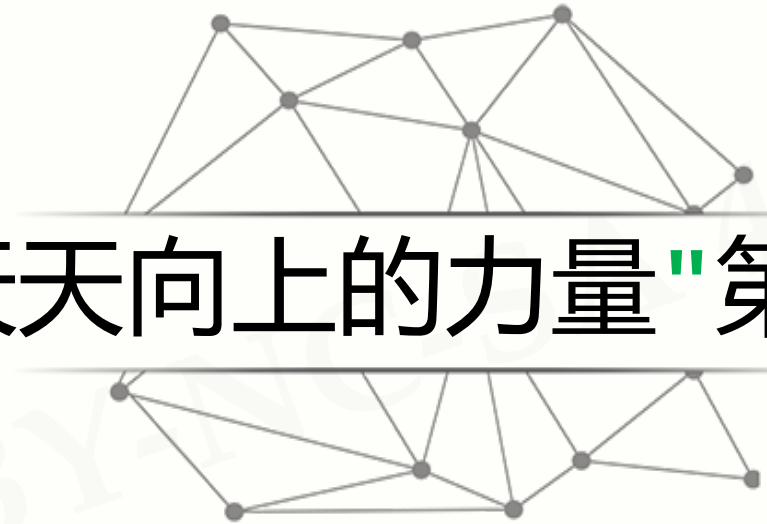
>>>（1%运行结果）

向上：37.78，向下：0.03

$$1.01^{365} = 37.78$$

$$0.99^{365} = 0.03$$

1%的力量，惊人！



"天天向上的力量"第三问

天天向上的力量

问题3： 工作日的力量

- 一年365天，一周5个工作日，每天进步1%
- 一年365天，一周2个休息日，每天退步1%
- 这种工作日的力量，如何呢?

1.01^{365} (数学思维) ➡ for..in.. (计算思维)

天天向上的力量

```
#DayDayUpQ3.py
dayup = 1.0
dayfactor = 0.01
for i in range(365):
    if i % 7 in [6,0]:
        dayup = dayup*(1-dayfactor)
    else:
        dayup = dayup*(1+dayfactor)
print("工作日的力量：{:.2f}".format(dayup))
```

采用循环模拟365天的过程

抽象 + 自动化

天天向上的力量

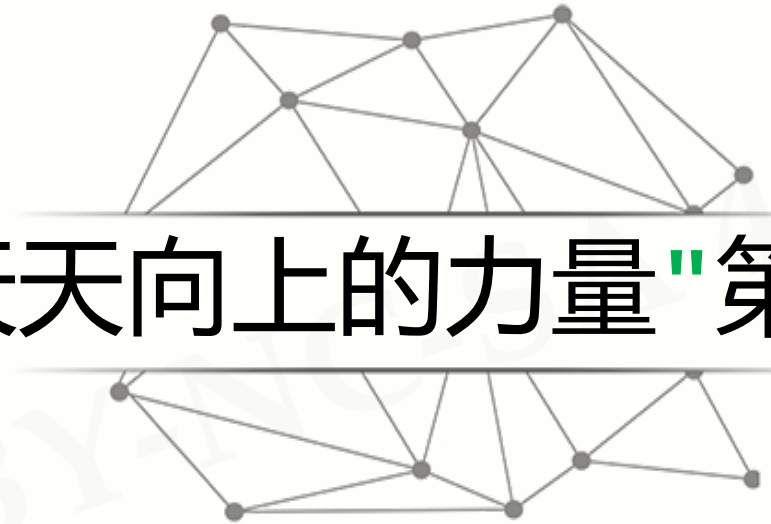
问题3： 工作日的力量

>>>（运行结果）

工作日的力量：4.63

$1.001^{365} = 1.44$　　$1.005^{365} = 6.17$　　$1.01^{365} = 37.78$

尽管工作日提高1%，但总体效果介于1‰和5‰的力量之间



"天天向上的力量"第四问

天天向上的力量

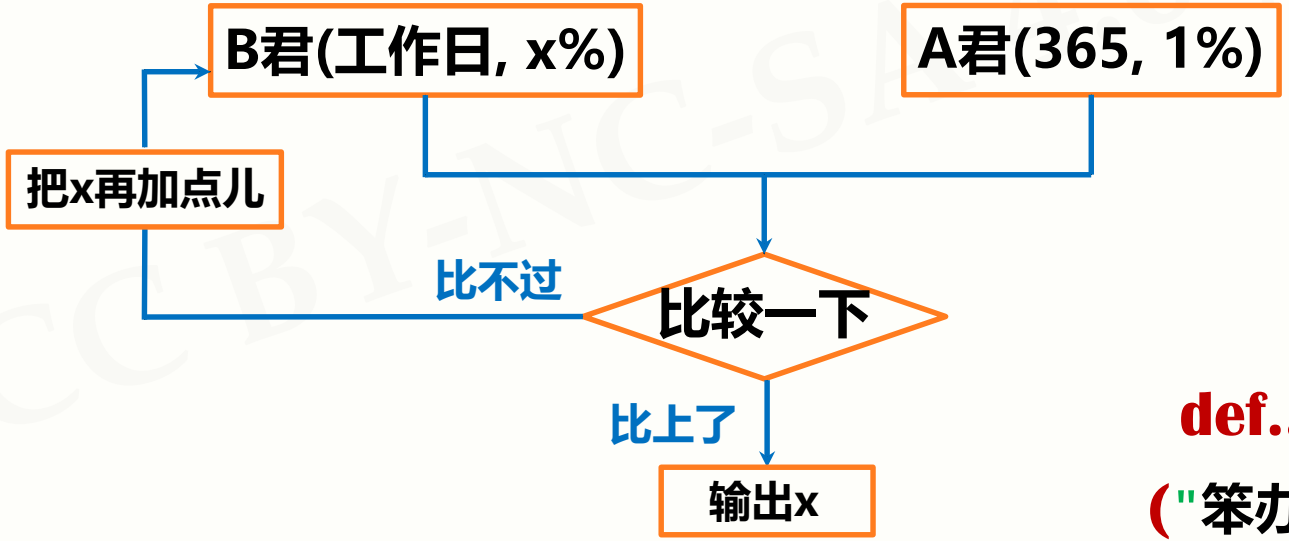
问题4： 工作日的努力

- 工作日模式要努力到什么水平，才能与每天努力1%一样?
- A君: 一年365天，每天进步1%，不停歇
- B君: 一年365天，每周工作5天休息2天，休息日下降1%，要多努力呢?

for..in.. (计算思维) ➡ **def..while.. ("笨办法"试错)**

天天向上的力量

问题4： 工作日的努力



def..while..

("笨办法"试错)

天天向上的力量

```
#DayDayUpQ4.py
def dayUP(df):
    dayup = 1
    for i in range(365):
        if i % 7 in [6,0]:
            dayup = dayup*(1 - 0.01)
        else:
            dayup = dayup*(1 + df)
    return dayup
dayfactor = 0.01
while dayUP(dayfactor) < 37.78:
    dayfactor += 0.001
print("工作日的努力参数是：{:.3f} ".format(dayfactor))
```

根据df参数计算工作日力量的函数

参数不同，这段代码可共用

def保留字用于定义函数

while保留字判断条件是否成立

条件成立时循环执行

准备好电脑，与老师一起编码吧！

天天向上的力量

问题4： 工作日的努力

>>>（运行结果）

工作日的努力参数是：0.019

$1.01^{365} = 37.78$　　$1.019^{365} = 962.89$

工作日模式，每天要努力到1.9%，相当于365模式每天1%的效果！

天天向上的力量

GRIT：perseverance and passion for long-term goals

$1.01^{365} = 37.78$　　$1.019^{365} = 962.89$

- GRIT，坚毅，对长期目标的持续激情及持久耐力
- GRIT是获得成功最重要的因素之一，牢记天天向上的力量

"天天向上的力量"举一反三

```
#DayDayUpQ3.py
dayup = 1.0
dayfactor = 0.01
for i in range(365):
    if i % 7 in [6,0]:
        dayup = dayup*(1-dayfactor)
    else:
        dayup = dayup*(1+dayfactor)
print("工作日的力量：{:.2f}".format(dayup))
```

for..in.. (计算思维)

```
#DayDayUpQ4.py
def dayUP(df):
    dayup = 1
    for i in range(365):
        if i % 7 in [6,0]:
            dayup = dayup*(1 - 0.01)
        else:
            dayup = dayup*(1 + df)
    return dayup
dayfactor = 0.01
while dayUP(dayfactor) < 37.78:
    dayfactor += 0.001
print("工作日的努力参数是：{:.3f} ".format(dayfactor))
```

def..while..

("笨办法"试错)

python

举一反三

天天向上的力量

- 实例虽然仅包含8-12行代码，但包含很多语法元素
- 条件循环、计数循环、分支、函数、计算思维
- 清楚理解这些代码能够快速入门Python语言

举一反三

问题的变化和扩展

- 工作日模式中，如果休息日不下降呢?
- 如果努力每天提高1%，休息时每天下降1‰呢?
- 如果工作3天休息1天呢?

举一反三

问题的变化和扩展

- "三天打鱼，两天晒网"呢?
- "多一份努力"呢？（努力比下降多一点儿）
- "多一点懈怠"呢？（下降比努力多一点儿）

小花絮

全国计算机等级考试二级 Python科目

http://ncre.neea.edu.cn

全国计算机等级考试（简称NCRE）是教育部批准，由教育部考试中心主办，面向社会，用于考查应试人员计算机应用知识与技能的全国性计算机水平考试体系。

二级Python语言科目在 2018年9月 首考，异常火爆！