

day02 基础&运算符

今日概要

1. 循环
2. 字符串格式化
3. 运算符
4. 编码
5. 博客 & git

内容回顾 & 补充

内容回顾

- 计算机基础
- 安装解释器
 - py2
 - py3
- 语法
 - print / input
 - 整型 int / 字符串 str / 布尔值 boolen
 - 条件语句
 - and运算符
 - 变量
- 练习:

评分规则:

- A >=90
- B >=80
- C >=70
- D 其他

用户输入成绩，根据成绩的不同显示不同的级别。

```
score = input('请输入成绩')
score_int = int(score)
if score_int >= 90:
    print('A')
elif score_int >=80:
    print('B')
elif score_int >=70:
    print('C')
else:
    print('D')
```

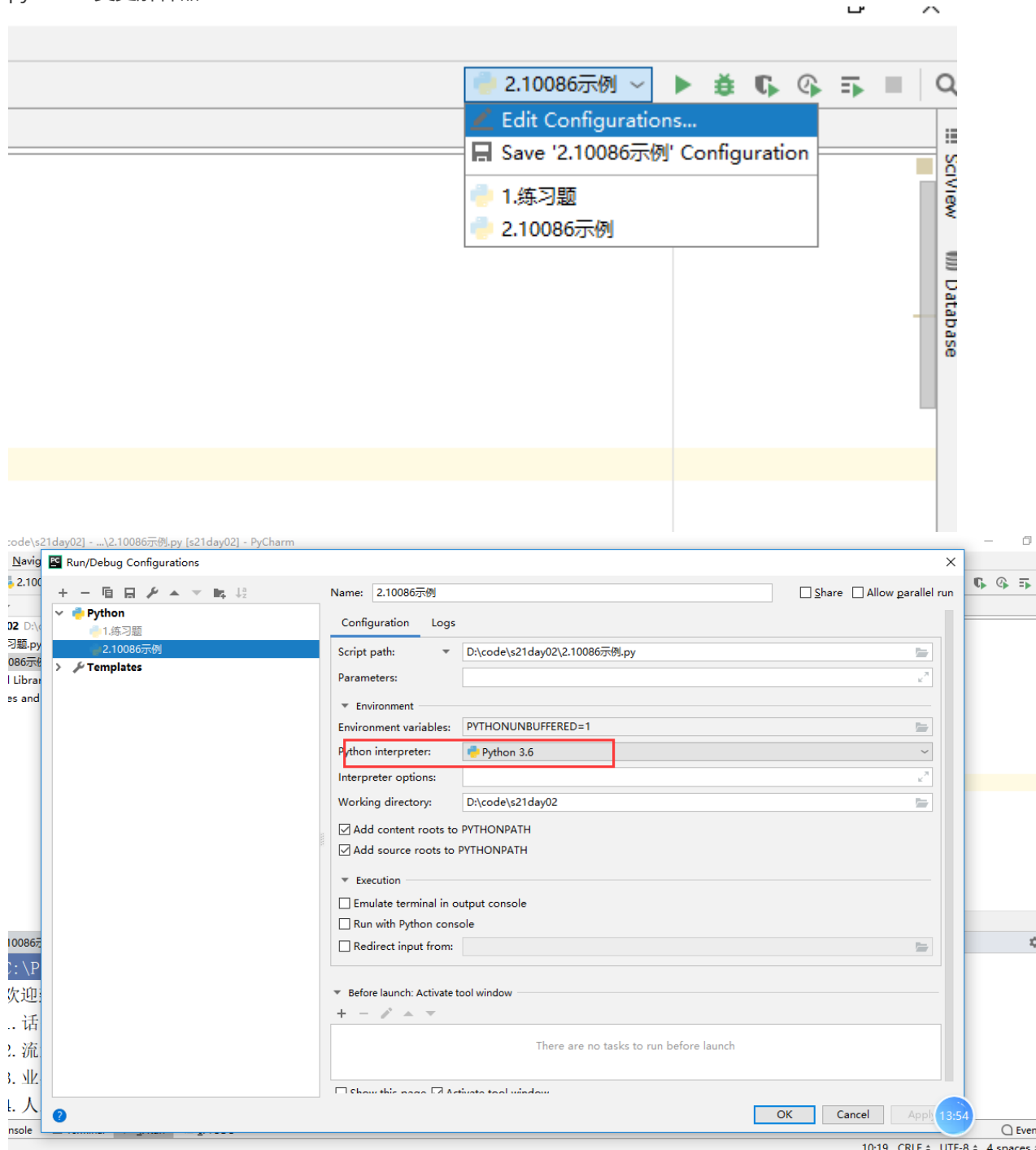
补充

1. if条件的嵌套

```
message = """欢迎致电10086
1.话费查询;
2.流量服务;
3.业务办理;
4.人工服务"""
print(message)

index = input('请输入你要选择的服务:')
index = int(index)
if index==1:
    print('话费查询')
elif index == 2:
    print('流量服务')
elif index == 3:
    content = """业务办理
1. 修改密码;
2. 更改套餐;
3. 停机; """
    print(content)
    value = input('请输入要办理的业务: ')
    value = int(value)
    if value == 1:
        print('修改密码')
    elif value == 2:
        print('更改套餐')
    elif value == 3:
        print('停机')
    else:
        print('错误')
elif index == 4:
    print('人工服务')
else:
    print('输入错误')
```

2. pycharm变更解释器



今日内容

1. 循环语句

1. 循环打印“人生苦短，我用Python。”

```
while True:
    print('人生苦短，我用Python。')
```

```

13
14     count = 1
15     value = count + 1
16     print(value)
17     """
18
19     # count = 1
20     # count = count + 1
21     # print(count)
22
23     # 4. 请通过循环, 让count每次循环都 + 1 .
24     count = 1
25     while True:
26         print(count)
27         count = count + 1

```

2. while后加入条件

```

while 1>0 and 2>1:
    print('人生苦短, 我用Python。')

```

3. 数字相加

```

"""
count = 1
value = count + 1
print(value)
"""
"""
count = 1
count = count + 1
print(count)
"""

```

4. 请通过循环, 让count每次循环都 + 1 .

```

"""
count = 1
while True:
    print(count)
    count = count + 1
"""

# 练习
"""
while True:
    count = 1
    print(count)
    count = count + 1
"""

```

5. 请通过循环, 1 2 3 .. 10.

```

count = 1
while count <= 10:
    print(count)
    count = count + 1
print('结束')

```

6. 请通过循环, 1 2 3 4 5 6 8 9 10.

```

"""
# 错误示例
count = 1
while count <= 10 and count != 7 :
    print(count)
    count = count + 1
"""

"""
# 正确
count = 1
while count <= 6:
    print(count)
    count = count + 1
count = 8
while count <= 10:
    print(count)
    count = count + 1
"""

"""
# 正确
count = 1
while count <= 10:
    if count != 7:
        print(count)
    count = count + 1
"""

```

```

"""
# 正确
count = 1
while count <= 10:
    if count == 7:
        pass
    else:
        print(count)
    count = count + 1
"""

```

7. break

```

while True:
    print(666)
    break # 终止当前循环
print('结束')

# 练习:
"""
# 通过break实现 1 ~ 10
count = 1
while True:
    print(count)
    if count == 10:
        break
    count = count + 1
print('结束')
"""

# break是终止当前循环
while True:
    print('你好')
    while True:
        print(666)
        break
    break
"""

```

8. continue

```

"""
count = 1
while count <=10:
    print(count)
    continue # 本次循环如果遇到continue, 则不在继续往下走, 而是回到while条件位置。
    count = count + 1
"""

# 示例: 1234568910

```

```

count = 1
while count <=10:
    if count == 7:
        count = count + 1
        continue
    print(count)
    count = count + 1

```

9. while else

```

"""
count = 1
while count < 10:
    print(count)
    count = count + 1
else: # 不再满足while后的条件时, 触发。 或 条件=False
    print('ELSE代码块')

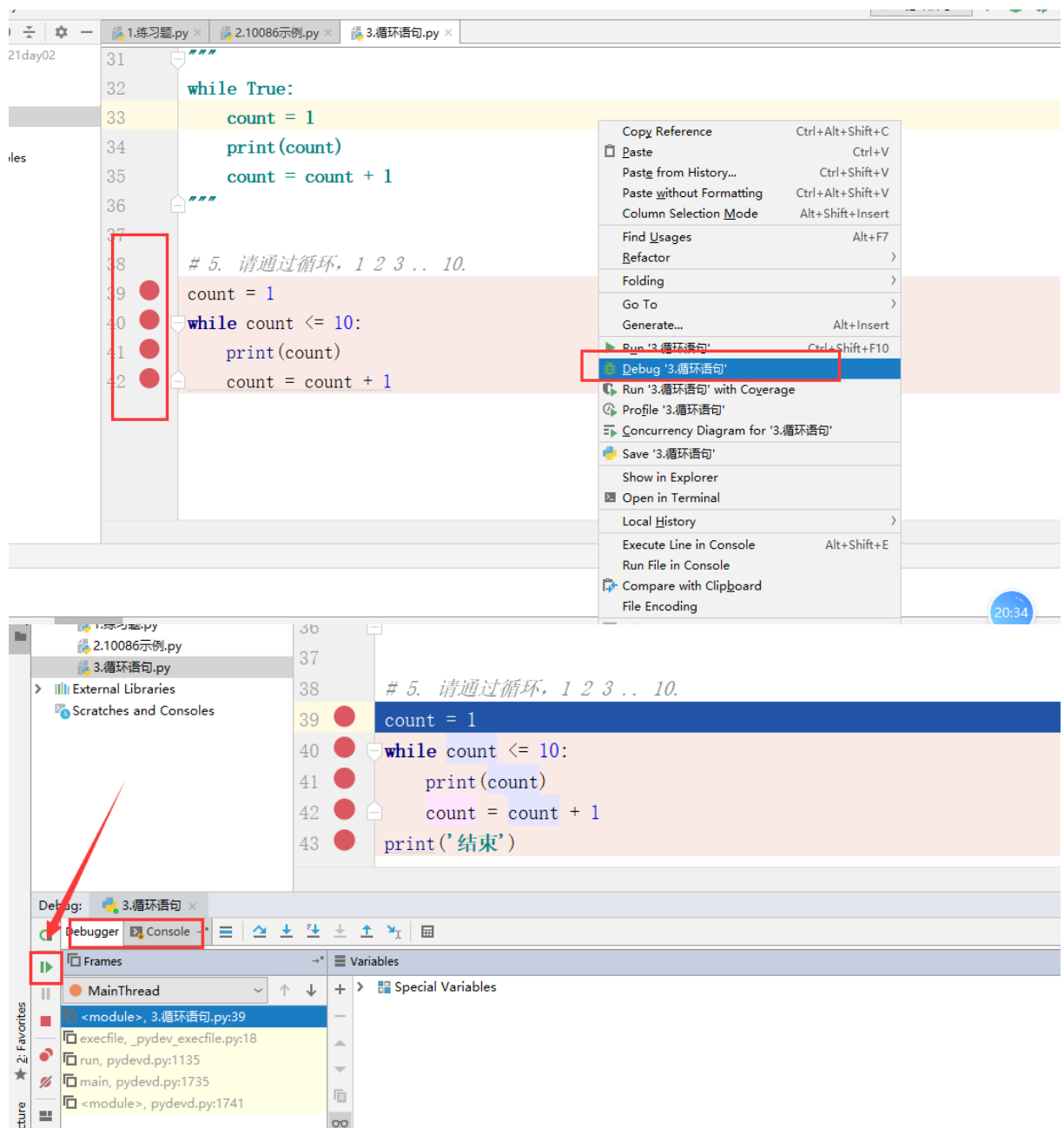
print('结束')
"""

"""
count = 1
while True:
    print(count)
    if count == 10:
        break
    count = count + 1
else: # 不再满足while后的条件时, 触发。 或 条件=False
    print('ELSE代码块')
print('结束')
"""

```

10. 其他

- 快速注释 ctrl+?
- pycharm断点



11. 总结

- o while基本结构
- o break
- o continue
- o while else

2. 字符串格式化

1. %s


```
# 字符串格式化存在的意义
name = input('姓名: ')
do = input('在干什么: ')
template = "%s在教室, %s。" %(name,do,)
print(template)

# 直接做占位符
# template = "我是%s, 年龄%s, 职业%s。" %("alex",73,'讲鸡汤',)
# print(template)
```

2. %d

```
# template = "我是%s, 年龄%d, 职业%s。" %("alex",73,'讲鸡汤',)
# print(template)
```

3. %%

```
# name = 'alex'
# template = "%s现在手机的电量是100%%" %(name,)
# print(template)
```

4. 练习

```
name = input('请输入姓名:')
age = input('请输入年龄:')
job = input('请输入职业:')
hobby = input('请输入爱好:')
msg = '''
----- info of Alex Li -----
Name   : %s
Age    : %s
job    : %s
Hobbie: %s
----- end -----'''

data = msg %(name,age,job,hobby,)
print(data)
```

3. 运算符

1. 算数运算

```
# 练习题: 1 ~ 100 之间所有的数相加。
# total = 0
# count = 1
# while count <=100:
#     total = total + count
#     count = count + 1
# print(total)
```

练习题: 打印 1 ~ 100 之间的奇数。

```
count = 1
while count <= 100:
    val = count % 2
    if val == 1:
        print(count)
    count = count + 1
```

2. 赋值运算

```
# count = 1
# while count <=100:
#     print(count)
#     count +=1 # count = count + 1
```

3. 逻辑运算

- 一般情况, 用于做判断。

```
if 1 > 0 and 1 > 2:
    print('666')
```

- 二般情况, 用于做取值。

■ or

```
"""
对于 or, 如果有遇到 value= 1 or 9
第一个值如果是转换成布尔值如果是真, 则value=第一值。
第一个值如果是转换成布尔值如果是假, 则value=第二值。
如果有多个or条件, 则从左到右依次进行上述流程。
示例:
    v1 = 0 or 1
    v2 = 8 or 10
    v3 = 0 or 9 or 8
"""
```

■ and

```
"""
对于and, 如果遇到 value= 1 and 9 这种情况
如果第一个值转换成布尔值是True, 则value=第二个值。
如果第一个值转换成布尔值是False, 则value=第一个值。
如果有多个and条件, 则从左到右依次进行上述流程。
示例:
    v1 = 1 and 9
    v2 = 1 and 0
    v3 = 0 and 7
    v4 = 0 and ""
    v5 = 1 and 0 and 9
"""
```

- 结合

```
# 先看and再看or
# v1 = 1 and 9 or 0 and 6
# print(v1)
```

- 其他

- 优先级 在没有()的情况下not 优先级高于 and，and优先级高于or，即优先级关系为(>)>not>and>or，同一优先级从左往右计算。
- 数据类型转换

```
# 数字转字符串
# v1 = 666
# v2 = str(v1)
# 字符串转数字
# v1 = "666"
# v2 = int(v1)

# 数字转布尔值
# v1 = 0
# v2 = bool(v1)
# print(v2)

# 字符串转布尔值
# v1 = ""
# v2 = bool(v1)
# print(v2)

# 布尔值转换其他
# v1 = True
# v2 = str(v1)
# print(v2)

# 需要掌握的转换知识点
"""
    - 字符串转数字
    - 数字转字符串
    - "" / 0 转换布尔之后是False
"""
```

4. 编码

- 编码扩展

- ascii
- unicode
 - ecs2
 - ecs4
- utf-8, 中文用3字节。
- utf-16

- gbk, 中文用2字节。
- gb2312, 中文用2字节。
- 单位

```
8bit = 1byte  
1024byte = 1KB  
1024KB = 1MB  
1024MB = 1GB  
1024GB = 1TB  
1024TB = 1PB  
1024TB = 1EB  
1024EB = 1ZB  
1024ZB = 1YB  
1024YB = 1NB  
1024NB = 1DB  
常用到TB就够了
```

5. 博客

博客园开自己的博客。

- 注册
- 申请开通博客
- 写博客
 - 随笔, 随便看。
 - 文章, 有URL就能看。
 - 日记, 只能自己看。

◦ 皮肤

随笔	文章	日记	评论	链接	相册	文件	设置	选项
----	----	----	----	----	----	----	----	----

修改设置

标题

子标题

Blog地址名: wupeiqi (更改Blog地址名会让以前的文章链接失效, 请不要轻易修改, 如果确实要修改, 请发邮件至contact@cnblogs.com联系管理员修改)

登录用户名: wptawy [\[修改登录用户名\]](#)

密 码: ***** [\[修改密码\]](#)

显示名称(作者名): 武沛齐 [\[修改显示名称\]](#)

Email: 424662508@qq.com [\[修改Email\]](#)

回复通知Email: [修改页面移到这里](#)

时区

Language/Locale

博客皮肤(124) [最新皮肤](#)
 [根据预览图片选择皮肤\(双击图片选择相应的皮肤\)](#)

6. git

- 安装git软件 (下一步直到完成)
- 码云注册 (保存代码)

创建代码托管仓库

https://gitee.com/wupeiqi/events

码云 开源软件 企业版 特惠 高校版 博客 我的码云

搜索项目、代码片段...

新建仓库

发布代码片段

创建组织

开通企业版

从 GitHub 导入仓库

武沛齐 wupeiqi

2 Followers 0 Stars 0 Following 31 Watches

武沛齐 很懒，啥也没写

加入于 5年前

仓库 组织 企业

Public Forks Private

Search...

s21day08

manu

s22day25

day04

luffy

查看所有 >>

所有 我的

昨天 2019-03-27

武沛齐

创建了 老男孩Python全栈21期/教学计划

删除了 老男孩Python全栈21期/教学计划

删除了 老男孩Python全栈21期/01张弛

2019-03-18

武沛齐

推送到了 武沛齐/pythonav 的 master 分支

ad0ff0a 分页数据和总页码

2019-03-15

武沛齐

推送到了 武沛齐/pythonav 的 master 分支

d427188 1.展示其他用户面试题答案; 2.每天可以刷20个面试题

推送到了 武沛齐/pythonav 的 master 分支

a439909 视频管理页面不再显示已删除的讲师信息

创建仓库

码云 开源软件 企业版 特惠 高校版 博客 我的码云

搜索项目、代码片段...

仓库名称 ✓

s21day02

归属

武沛齐

路径

https://gitee.com/wupeiqi/ s21day02

仓库介绍 非必填

这是我第二天的作业

是否开源

☐ 私有 ☒ 公开

选择语言

请选择语言

添加 .gitignore

请选择 .gitignore 模板

添加开源许可证

请选择开源许可证

☐ 使用Readme文件初始化这个仓库

☐ 使用Issue模板文件初始化这个仓库

☐ 使用Pull Request模板文件初始化这个仓库

导入已有仓库

企业版介绍

社区版与企业版功能对比

他们都在用

已有超过 50000 企业研发团队选择码云企业版，60% 客户来自口碑推荐。

招商银行 招商证券 中国电信 中国联通 比亚迪 比亚迪 比亚迪 SuperMap

了解更多

企业版介绍

社区版与企业版功能对比

- 创建“文件夹”



- 写作业并提交到 码云。
 - 在某个文件夹下写作业。
 - 写完之后，在此文件夹下 鼠标右键，并选择【git bash here】
 - 在“黑框”里输入命令
 - git init，用于做初始化，其实就是让git把当前所在的文件夹管理起来。
 - git add .，将当前所在的文件夹中的所有文件收集起来。
 - git commit -m "第二天的作业"，做个记录。

第一次执行此命令时候，需要先执行：

```
git config --global user.email "wupeiqi@live.com"
git config --global user.name "wupeiqi"
```

```
git remote add origin https://gitee.com/wupeiqi/s21day02.git
```

- git push origin master
- 提示：输入用户名和密码（码云）

今日作业

- 猜数字，设定一个理想数字比如：66，让用户输入数字，如果比66大，则显示猜测的结果大了；如果比66小，则显示猜测的结果小了；只有等于66，显示猜测结果正确，然后退出循环。
- 在上一题的基础，设置：给用户三次猜测机会，如果三次之内猜测对了，则显示猜测正确，退出循环，如果三次之内没有猜测正确，则自动退出循环，并显示‘大笨蛋’。
- 使用两种方法实现输出 1 2 3 4 5 6 8 9 10。

4. 求1-100的所有数的和
5. 输出 1-100 内的所有奇数
6. 输出 1-100 内的所有偶数
7. 求1-2+3-4+5 ... 99的所有数的和
8. 用户登陆（三次输错机会）且每次输错误时显示剩余错误次数（提示：使用字符串格式化）
9. 简述ASCII、Unicode、utf-8编码
10. 简述位和字节的关系？
11. 猜年龄游戏 要求：允许用户最多尝试3次，3次都没猜对的话，就直接退出，如果猜对了，打印恭喜信息并退出
12. 猜年龄游戏升级版 要求：允许用户最多尝试3次，每尝试3次后，如果还没猜对，就问用户是否还想继续玩，如果回答Y，就继续让其猜3次，以此往复，如果回答N，就退出程序，如果猜对了，就直接退出。
13. 判断下列逻辑语句的True,False
 - o $1 > 1 \text{ or } 3 < 4 \text{ or } 4 > 5 \text{ and } 2 > 1 \text{ and } 9 > 8 \text{ or } 7 < 6$
 - o $\text{not } 2 > 1 \text{ and } 3 < 4 \text{ or } 4 > 5 \text{ and } 2 > 1 \text{ and } 9 > 8 \text{ or } 7 < 6$
14. 求出下列逻辑语句的值。
 - o $8 \text{ or } 3 \text{ and } 4 \text{ or } 2 \text{ and } 0 \text{ or } 9 \text{ and } 7$
 - o $0 \text{ or } 2 \text{ and } 3 \text{ and } 4 \text{ or } 6 \text{ and } 0 \text{ or } 3$
15. 下列结果是什么？
 - o $6 \text{ or } 2 > 1$
 - o $3 \text{ or } 2 > 1$
 - o $0 \text{ or } 5 < 4$
 - o $5 < 4 \text{ or } 3$
 - o $2 > 1 \text{ or } 6$
 - o $3 \text{ and } 2 > 1$
 - o $0 \text{ and } 3 > 1$
 - o $2 > 1 \text{ and } 3$
 - o $3 > 1 \text{ and } 0$
 - o $3 > 1 \text{ and } 2 \text{ or } 2 < 3 \text{ and } 3 \text{ and } 4 \text{ or } 3 > 2$