

Day2

==今日必须掌握：==

- ☐ if语句，以及if-elif(可多)-else分支
- ☐ 理解布尔逻辑，如False and ...不会执行and后的语句
- ☐ 三元运算符：a if a>b else b

c语言的.c文件如果要执行

1.生成json文件阶段：终端-配置任务-生成活动文件 2.编译运行阶段：终端-运行生成任务-生成活动文件 3.在终端./test.exe即可运行

vscode的python代码补全插件——python snippets(cstrap)

一.if条件判断

- 每条if语句(if,elif,else)的核心都是一个值为True或False的表达式，这种表达式称作**条件测试**。python根据条件测试的值是True还是False来决定是否执行if语句中的代码。如果条件测试的值为True，则执行紧跟在if语句后的代码；如果为False，**python**就忽略if语句后面的代码。

if语句和布尔逻辑的结合

grade_calculator.py 等级计算

```
score = int(input('请输入成绩: '))
if score >= 90:
    grade = 'A'
elif score >= 80:
    grade = 'B'
elif score >= 70:
    grade = 'C'
elif score >= 60:
    grade = 'D'
else:
    grade = 'F'

if grade != 'F':
    last_digit = score % 10
    if last_digit >= 7:
        grade += "+"
    elif last_digit <= 3:
        grade += "-"

print(f"成绩等级: {grade}")
```

这段代码获取用户的输入转换为int型并赋给score，接下来通过if—elif—else语句给各个分数划分不同的等级 然后通过含有比较运算符!=的条件判断语句if筛选出大于60分的分数，接着让分数除以10后取余数，如果余数大于等于7，就在grade前添加'+',如果小于等于3，则在其前面增加'-'。最后通过f-string的格式化输出输出等级。

二.布尔逻辑

1.含义

python中的布尔值包括

- True(真or1), False(假or0)
- and(与), or(或), not(非)

2.代码

```
#闰年判断
year = int(input("请输入年份: "))
is_leap = (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0)
print(f'{year}是闰年' if is_leap else f'{year}不是闰年')
```

1.这段代码的第二行运用了布尔值and和or，当and连接的两条语句都为真时这一整条语句为真，当or连接的两条语句起码有一个为真时，这一整条语句为真 2.第三行运用了==布尔逻辑==，==三元运算符==，==f-string==格式化输出这三个知识点。

三元运算符实例

```
a = int(1)
b = int(2)
max = a if a>b else b
print(max)
```

其实现的功能类似以上的代码，如果a>b为True，则返回if前面的语句，如果为False，则返回else后面的语句，这就是==三元运算符==，其中涉及到了==布尔逻辑==，而如果把==f-string==运用其中，就可以把这个代码改写成：

```
a = int(input('请输入第一个数字: '))
b = int(input('请输入第二个数字: '))
print(f'{a}更大' if a>b else f'{b}更大')
```

这是python中一个非常强大而又简洁的表达方式