conditional 條件判斷式

2020年8月14日　　下午 09:35

if

[
if condition:
　　statements
]

[
if 一段條件:
　　在此條件下之執行作為
]

注意

：也就是這個colon 條件判斷式 必須

而:之下一行空格控制 statements是否在block內，空格長度(indention size)多少可自行決定，但同一個block中空格長度要一致系統才能認得是同個block ，可以統一使用TAB

EX:
[

```
a=10
if a<1:
        print("a<1")
        print("great")
```

]

和下方式子結果不同，因為 因為少了空格

[

```
a=10
if a<1:
        print("a<1")
print("great")
```

]

EX:

[

```
a=10
if a<1:
        print("a<1")
  print("great")
```

]

上式因空格長度不一而產生了IndentationError

EX:
[
#累進稅

```
income=float(input())

if income <= 10000:
        tax = 0.02 * income
if income>10000:
        tax = 0.08 * (income-10000)+200
print("Tax amount : $"+str(tax))
```

]

上式中條件必須互斥，如沒互斥會有潛在的不一致性(potential unconsistency)，可由if else寫較簡潔

if-else
(隔行)

[

```
if condition:
        statements 1
else:
        statements 2
```

]

***else正上方(前方)一定要對一個同等級的if，而if 的block裡面不一定要有else**

而else前面沒有空格，和正上方(前方)if在同一行，而因為在同一行(也可以說同一個地位)之if已有其條件故else 後面不用再放條件else 正後方沒有東西，因為一個else代表著除了上方對應if以外其它的情況，故

EX:

[

```
#累進稅

income=float(input())

if income <= 10000:
        tax = 0.02 * income
else:
        tax = 0.08 * (income-10000)+200
print("Tax amount : $"+str(tax))
```

]

EX:

[

```
#累進稅
```

```
income=float(input())

if income <= 10000:
        tax = 0.02 * income
        else:
        tax = 0.08 * (income-10000)+200
print("Tax amount : $"+str(tax))

]
```

因else正上方(前方)無對應之if 故error

Nested if-else

巢狀if-else

```
[
if  condition 1:
        if condition 2:
                statement A
        else
                statement B
else:
        statement C
]
```

可在 if 或 else的block中加入 if-else的條件，像是用 Logicial operator 的拉長版(兩個if

以上是兩層為例子，可以無限多層，

EX:

```
[
a = int(input())
b = int(input())
c = int(input())

if a <= b  :
        if a <= c  :
                print(a, " is the smallest")
        else :
                print(c, " is the smallest")
else:
        if b <= c  :
                print(b, "is the smallest")
        else:
                print(c, "is the smallest")
]
```

以上是先判斷a和b的大小，再判斷小的那個與c的大小(等號等於同為最小)

而上式is the smallest打太多次，容易有potential inconsistency的打錯(或複製不完全)發生

故有更好(簡潔)寫法

```
[

a = int(input())
b = int(input())
c = int(input())
```

```
min = 0

if a <= b :
	if a <= c :
		min = a
	else :
		min = c
else :
	if b<= c :
		min = b
	else :
		min = c
print(min, "is the smallest")

]
```

先給一個min值，決定好a b c誰值最小後再print min

而上式又可以更簡潔

```
[

a = int(input())
b = int(input())
c = int(input())

min = c
if a <= b :
	if a <= c :
		min = a
else :
	if b <= c :
		min = b
print(min, "is the smallest")

]
```

先認定min 就是 c ，如果a或b最小min就重新被指派，如c還是最小就維持原樣min = c

Ex:

邏輯錯誤例子

```
[
a = 10
b = 9

if a == 10 :
	if b == 10 :
		print("a and b are both ten.")
	else :
		print("a is not ten.\n")
]
```

此式有logical error 式子雖然可以跑但邏輯上有錯誤

```
[
a = 10
b = 9

if a == 10 :
	if b == 10 :
		print("a and b are both ten.")
else :
	print("a is not ten.\n")
```

]

tenary if operator
if -else
(同行)

[

operation A if condition else operation B

]

其實就if else 寫成直行的樣子

[

```
a = int(input())
b = int(input())
c = int(input())

min = 0

if a<= b :
        min = a if a<= c else c
else :
        min = b if b<= c else c

print(min, "is the smallest")
```

]

=(assignment) 後面 是要計算出一個值(可為計算式，不可包含另一個assignment)

而可多加利用 Parentheses 括弧標出先後處理順序在不影響結果之下(EX:加減乘除中括號有時有差)讓人看語法更好解讀

[

```
a = int(input())
b = int(input())
c = int(input())

min = 0

if a<= b :
        min =( a if (a<= c) else c)
else :
        min = (b if (b<= c) else c)

print(min, "is the smallest")
```

]

Nested else if (即elif)

將else 下方加上 if

或是用elif

用來處理:[如果是A條件，就執行甲，B條件，就執行乙，如果不是A也不是B就執行丙]

[

```
a = int(input())

if a<10 :
        print("a<10")
else :
        if a>10 :
                print("a>10.")
        else :
                print("a=10.")
```

]

上式同一件事情分不同階層會降低可讀性，故可用elif

[

```
a = int(input())

if a<10 :
        print("a<10")
elif a>10 :
        print("a>10.")
else :
        print("a=10.")
```

]

EX:

[

```
month = int(input())

if month == 1:
        print("31 days")
else :
        if month == 2 :
                print("28 days")
        else :
                if month ==3 :
                        print("31 days")
                else:
                .
                .
```

]

以上例子以同理可用elif簡潔成同一件事情

[

```
month = int(input())

if month == 31:
```

```
        print("31 days")
elif month ==2:
        print("28 days")
elif month ==3:
        print("31 days")
.
.
.

]
```

*上式最後不一定要else