

Chapter 7"if" statement

Example 1

Write a program that prints the maximum number of two numbers.

```
def Main():  
    print("Compute max of two numbers")  
    → x0, x1 = eval(input("Type two numbers x0, x1: "))  
    if x0 > x1:  
        print("Max number is ", x0)  
    else:  
        print("Max number is ", x1)
```

Main()

②

Example 2

Write a program to obtain the maximum of three numbers:

```
def Main():  
    print("Compute max of three number")  
    x0, x1, x2 = eval(input("Type three numbers x0, x1, x2: "))  
  
    if x0 > x1 and x0 > x2:  
        print("Max number is ", x0)  
    else:  
        if x1 > x0 and x1 > x2:  
            print("Max number is ", x1)  
        else:  
            print("Max number is ", x2)
```

Main()

As a short notation instead of ③

if ~~$x_0 > x_1$~~ and ~~$x_0 > x_2$~~
print(....

{ else:
if $x_1 > x_0$ and x_2

else: } → elif...
if }

we can have

elif $x_1 > x_0$ and x_2

so it becomes:

if $x_0 > x_1$ and $x_0 > x_2$:

print("Max number is", x_0)

→ { elif $x_1 > x_0$ and $x_1 > x_2$:
print("max number is", x_1)
else:
print("max number is", x_2)

(4)

Another approach to write this program.

Have a variable "max" that is initialized to x_0 and then change "max" if x_1 is larger than "max" or x_2 is larger than "max"

```
def Main():
```

```
     $x_0, x_1, x_2 = \text{eval}(\text{input}(\text{"type } x_0, x_1, x_2: \text{"}))$ 
```

```
    ②  $\text{max} = x_0$   
        IF  $\text{max} < x_1$ :
```

```
    ③  $\text{max} = x_1$   
        IF  $\text{max} < x_2$ :  
        ④  $\text{max} = x_2$ 
```

```
    Main()
```

Example run:

type x_0, x_1, x_2 : 3, 5, 2

① $x_0 = 3$ $x_1 = 5$ $x_2 = 2$

② $\text{max} = 3$

③ $\text{max} = 5$

④ $\text{max} = 5$

⑤
~~Let~~ We can generalize this
approach to any number of numbers.

Let write a function `myMax(list)`
that computes the maximum in a list
of numbers.

$i = 0 \quad 1 \quad 2 \quad 3$
→ `[3, 5, 2, 7]`

```
def myMax(list):  
    max = list[0] # Initialize max with first num  
    for i in range(len(list)):  
        if max < list[i]:  
            max = list[i]  
    # max is ready  
    return max
```

def Main():

②

$x_0, x_1, x_2, x_3 = \text{eval}(\text{input}(\text{"Type } x_0, x_1, x_2, x_3:"}))$

nums = $[x_0, x_1, x_2, x_3]$

$m = \text{myMax}(\text{nums})$

print("Max is ", ~~myMax~~_m(nums))

Exception Handling

⑦

It is used to handle errors.

It will jump to a section of code called "handler" if there is an error.

try:

<body>

except <error type>:

<handler>

Example:

$$ax^2 + bx + c = 0$$

⑧

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

```
# quadratic.py  
# computes real roots of a  
# quadratic expression
```

```
import math
```

```
def Main():
```

```
    print("real solution of a quadratic equation")
```

```
    try:
```

```
        a, b, c = eval(input("Enter a, b, c: "))
```

```
        discRoot = math.sqrt(b*b - 4*a*c)
```

```
        root1 = (-b + discRoot) / (2*a)
```

```
        root2 = (-b - discRoot) / (2*a)
```

```
        print("The solutions are: ", root1, root2)
```

```
    except ValueError:
```

```
        print("No real roots")
```

It will
jump if
sqrt of
a negative
number.