Chapter-10 Conditional Statements

Contents

10.1 Decision Making Statement ... 3
10.1.1 Python Comparison Operators ... 3
10.1.2 Python Logical Operators ... 3
10.1.3 Python Membership Operators ... 4
10.2 if Statement ... 4
10.2.1 Python if Statement Syntax and Flowchart ... 4
10.2.2 If ... Statement Example-1 ... 5
10.2.3 If ... Statement Example-2 ... 5
10.2.4 If ... Statement Example-3 ... 5
10.2.5 If ... Statement Example-4 ... 6
10.2.6 If ... Statement Example-5 ... 6
10.2.7 If ... Statement Example-6 ... 7
10.2.8 If ... Statement Example-7 ... 7
10.2.9 If ... Statement Example – 8 ... 8
10.2.10 If ... Statement Example – 9 ... 8
10.2.11 If ... Statement Example – 10 ... 8
10.3 if... else Statement ... 9
10.3.1 If ...else ... Statement Example-1 ... 10
10.3.2 If ...else Statement Example-2 ... 10
10.3.3 If ...else ... Statement Example-3 ... 10
10.3.4 If ...else ... Statement Example - 4 ... 10
10.3.5 If ...else ... Statement Example - 5 ... 11
10.3.6 If ...else ... Statement Example-6 ... 11
10.4 if... elif... else statement ... 12
10.4.1 If elif else Statement Example-1 ... 13
10.4.2 If elif else Statement Example - 2 ... 13

10.4.3 If elif else Statement Example-3 14
10.4.4 If elif else Statement Example-4 15
10.4.5 If elif else Statement Example-5 15
10.4.6 If elif else Statement Example - 6 15
10.4.7 If elif else Statement Example-7 16
10.5 Python Nested if statements 16
10.5.1 Nested if statements Example-1 16
10.5.2 Nested if statements Example-2 17

Chapter-9 Conditional Statements

10.1 Decision Making Statement

ပထမဦးစွာ Decision-making statement များကို နားလည် သဘောပေါက်သင့်သည်။ ဆုံးဖြတ်ရမဲ့ အခြေအနေ(condition)တစ်ခုသည် မှန်သည်(True) သို့မဟုတ် မှားသည်(False)ဟူ၍ ရလဒ် နှစ်မျိုး ဖြစ်နိုင်သည်။ မှန်တယ်(True) လက်ခံနိုင်တယ် ဆိုရင် fulfilled ဖြစ်တယ်လို့ဆိုပြီးတော့လည်း သုံးနှုန်းကြသည်။ မှားတယ်(False)ဆိုရင် လက်မခံနိုင်ဘူး၊ fulfilled မဖြစ်ဘူး ဆိုပြီးတော့လည်း သုံးနှုန်းကြသည်။ အမှန်၊ အမှားဆုံးဖြတ်ရမဲ့ အခြေအနေ(condition)ကို စစ်ဆေးဘို့ ရေးသားထားတဲ့ code တွေကို decision-making statement လို့ ခေါ်သည်။ Conditional statement လို့လည်းခေါ်သည်။ Conditional statement ဆိုသည့် အခေါ်အဝေါ်ကို အသုံးအများသည်။

Python တွင် အဓိကကြသည့် control statement နှစ်မျိုး ရှိသည်။

- if statement နှင့်
- while statement တို့ ဖြစ်သည်။ အတိုခေါက်အားဖြင့် If statement ဟု လည်းခေါ်သည်။

If statement ဆုံးဖြတ်ချက်တစ်ခုခုကို ချပေးရန်(decision making လုပ်ရန်)အတွက် If statement ကို အသုံးပြုသည်။ ထိုဆုံးဖြတ်ချက်မှ True သို့မဟုတ် False ကို ထုတ်ပေးသည်။ True သို့မဟုတ် False ရလဒ် အပေါ် မူတည်၍ နောက်ထပ် ဆောင်ရွက်ရမည့် အလုပ်များ ကွဲပြားသွားသည်။

10.1.1 Python Comparison Operators

Decision-making statement များတွင် comparison operator များကို အသုံးပြုသည်။ True သို့မဟုတ် False ရလဒ် ထုတ်ပေးသည်။

Operator	Name	Example
==	Equal to	**x == y**
!=	Not equal to	**x != y**
>	Greater than	**x > y**
<	Less than	**x < y**
>=	Greater than or equal to	**x >= y**
<=	Less than or equal to	**x <= y**

```
5 != 5    # 5 is not equal to 5.
```

```
False
```

10.1.2 Python Logical Operators

Conditional statement များတွင် logical operator များကို တွဲ၍ အသုံးပြုကြသည်။

Operator	Description	Example
and	Returns True if both statements are true	x < 5 **and** x < 10

or	Returns True if one of the statements is true	x < 5 **or** x < 4
not	Reverse the result, returns False if the result is true	**not**(x < 5 **and** x < 10)

10.1.3 Python Membership Operators

Object တစ်ခုသည် သိလိုသည့် sequence ထဲတွင် ပါဝင်နေသည် မပါဝင်သည်ကို သိရန်အတွက် membership operator ကို အသုံးပြုသည်။

Operator	Description	Example
in	Returns True if a sequence with the specified value is present in the object	x in y
not in	Returns True if a sequence with the specified value is not present in the object	x not in y

10.2 if Statement

if statement ၏ ရလဒ်သည် True ဖြစ်လျှင် ၎င်းအောက်ရှိ instruction များကို ဆောင်ရွက်ရမည် ဖြစ်ပြီး False ဖြစ်လျှင် နောက်အဆင့်တစ်ခုသို့ ကျော်သွားရမည်။ (The if statement performs an action if a condition is True or skips the action if the condition is False.)

If statement တွင်

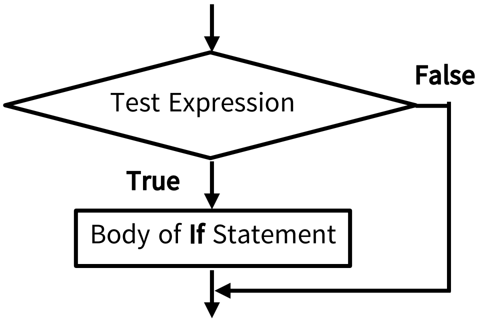
(၁) If တစ်ခုတည်းသာပါသည့် statement
(၂) If … else statement
(၃) If … elif … else statement
(၄) Nested If statement ဟူ၍ ကွဲပြားသည်။

If တစ်ခုတည်းသာပါသည့် statement နှင့် သင့်လျော်သည့် အခြေအနေများရှိသော်လည်း အသုံးနည်းသည်။

10.2.1 Python if Statement Syntax and Flowchart

Python if Statement Syntax

```
if test expression:
       statement(s)
```



if statement တစ်ခုတွင် test condition များစွာ ပါဝင်နေနိုင်သည်။

if Statement Syntax

if ကိုရေးသည့်အခါ if နှင့် "**:**" တို့အကြားတွင် မိမိ စစ်ဆေးလိုသည့်အချက်(test expression)ကို ထည့်ရေး ရသည်။ အောက်တွင် if statement ရေးသားပုံ syntax ကို ဖော်ပြထားသည်။ If statement သည်

True condition အတွက်သာ ဖြစ်သည်။ မိမိ စစ်ဆေးသည့်အချက်(test expression)သည် မှားနေလျှင်(False condition) ဘာမှ လုပ်ပေးရန် မလိုသည့်အခါတွင် သုံးသည်။

ဥပမာ-ထည့်ပေးသည့် character သည် 'a' သော်လည်းကောင်း၊ 'e' သော်လည်းကောင်း၊ 'i' သော်လည်းကောင်း၊ 'o' သော်လည်းကောင်း၊ 'u' သော်လည်းကောင်း ဖြစ်လျှင် သရ(vowel) ဟု သတ်မှတ်သည်။ Test condition (၅)ခုကို **or** operator ဖြင့် ဆက်ထားသည်။

```
if((character == 'a') or (character == 'e') or (character == 'i') or
   (character == 'o') or (character == 'u')):
```

and သို့မဟုတ် **or** စသည့် logical operator များ နှင့် တွဲ၍ အသုံးပြုနိုင်သည်။ if statement ကို စာကြောင်း တစ်ကြောင်းတည်းဖြင့်လည်း ဤကဲ့သို့ ရေးနိုင်သည်။ သို့သော် ဖတ်ရန် မလွယ်ကူပါ။

```
if a > b: print("a is greater than b")
```

10.2.2 If ... Statement Example-1

ယနေ့သည် user ၏ မွေးနေ့ဖြစ်ပါက "Happy Birthday" ဖြင့် ဖော်ပြပြီး နှုတ်ခွန်းဆက်ရန် ဥပမာ ဖြစ်သည်။ မွေးနေ့ မဟုတ်ပါက ဘာမှ လုပ်ပေးရန် မလိုသောကြောင့် if statement တစ်ခုသာ လိုသည်။

```
If today == user.birthday
    Print ("Happy Birthday")
```

10.2.3 If ... Statement Example-2

တနင်္ဂနွေနေ့ (day = 'Sunday') မိုးရွာလျှင်(weather = 'raining') အိမ်မှာပဲ အိပ်နေမည်။

```
weather = 'Raining'
day = 'Sunday'
if weather == 'raining' and day == 'Sunday':
    print('Sleep at home')
```

```
Sleep at home
```

10.2.4 If ... Statement Example-3

ဤဥပမာသည် If တစ်ခုတည်းသာပါသည့် statement အတွက် အကောင်းဆုံး ဥပမာဖြစ်သည်။ True ဖြစ်လျှင်သာ အလုပ်လုပ်ရန် လိုသည်။ False ဖြစ်လျှင်ဘာမှ လုပ်ရန်မလိုပါ။ အပေါင်းကိန်း(positive number) ဟုတ် မဟုတ် စစ်ဆေးသည့် if statement ဖြစ်သည်။ ထည့်ပေးသည့် number သည် သုညထက် ငယ် မငယ် (number<0) စစ်သည်။ True ဖြစ်လျှင် အပေါင်းကိန်း ဖြစ်အောင် ပြောင်းပေးသည်။ ထည့်ပေးသည့် number -34 သည် အနုတ်ကိန်းဖြစ်သောကြောင့် အပေါင်းကိန်း ဖြစ်အောင် ပြောင်းပေးသည်။

```
# input ကို လက်ခံ ပြီး integer အဖြစ်သို့ ပြောင်းသည်။ int()

number = int(input("Please enter a number: "))

if(number<0):                          # number သည် သုညထက်ငယ်လျှင် (အနုတ်ကိန်းဖြစ်လျှင်)
```

```
    number = -(number)                      # အပေါင်းကိန်းဖြစ်အောင် ပြောင်းသည်။
print("Absolute number is: {}".format(number))
```

```
Please enter a number: -34
Absolute number is: 34
```

10.2.5 If … Statement Example-4

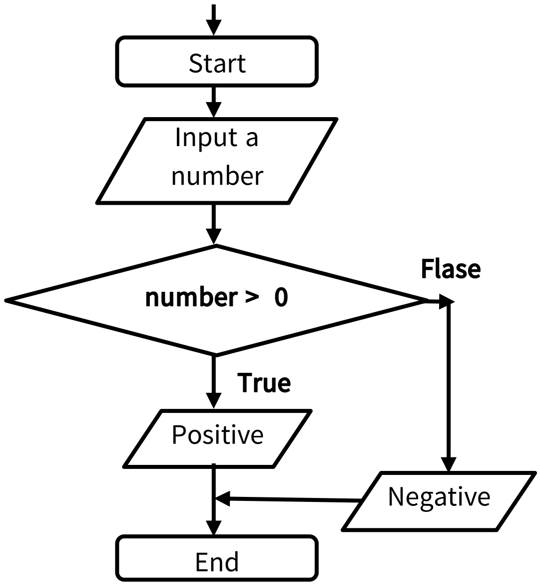
If တစ်ခုတည်းသာပါသည့် ဥပမာ ဖြစ်သည်။ အသက်(၂၀)ပြည့်ပြီး အမျိုးသားတစ်ယောက်သည် ကျန်းမာရေးကောင်းလျှင် (၂)နှစ် စစ်မှုထမ်းရမည်။ ထိုအချက် (၃)ချက်ထဲမှ တစ်ခုခုနှင့် မကိုက်ညီပါက စစ်မှုထမ်းရန် မလိုပါ။ မကိုက်ညီလျှင် ဘာမှ လုပ်ရန်မလိုသောကြောင့် if တစ်ခုတည်းသာပါသည့် statement ကို သုံးရန် သင့်လျော်သည်။ (၃)ချက်လုံးနှင့် ကိုက်ညီရမည် ဖြစ်သောကြောင့် **and** operator ကို သုံးသည်။

```
age_above_20 = True
male = True
health_fitness_ok = True

if age_above_20 and male and health_fitness_ok :
    print("Serve army for two years.")
```

```
Serve army for two years.
```

10.2.6 If … Statement Example-5



အပေါင်းကိန်း ဟုတ် မဟုတ် စစ်သည့် ဥပမာ ဖြစ်သည်။ `if num > 0:` သည် ပေးထားသည့် ကိန်းသည် သုညထက် ကြီးသည်၊ မကြီးသည်ကို စစ်ဆေးပြီး ဆုံးဖြတ်ပေးသည်။ မှန်ကန် ပါက(True ဖြစ်ပါက) ထိုကိန်းသည် ပေါင်းကိန်း ဖြစ်သည့်ဟု ဖော်ပြပေးသည်။ ဥပမာတွင် ထည့်ပေး သည့် ကိန်း(5) သည် သုညထက် ကြီးသောကြောင့် ပေါင်းကိန်း ဖြစ်သည့်ဟု ဖော်ပြ ပေးသည်။ False ဖြစ်လျှင် မည်သို့ ဆက်လုပ်ရမည်ကို ထည့်မရေး ထားပါ။ False ဖြစ်လျှင် လုပ်ရမည့်အလုပ်(instruction)ကို ရေးရန် အတွက် else… ထည့်ရေးရန် လိုသည်။ နောက်ပိုင်း တွင် If …else statement ဖြင့် ရေးပြထားသည်။

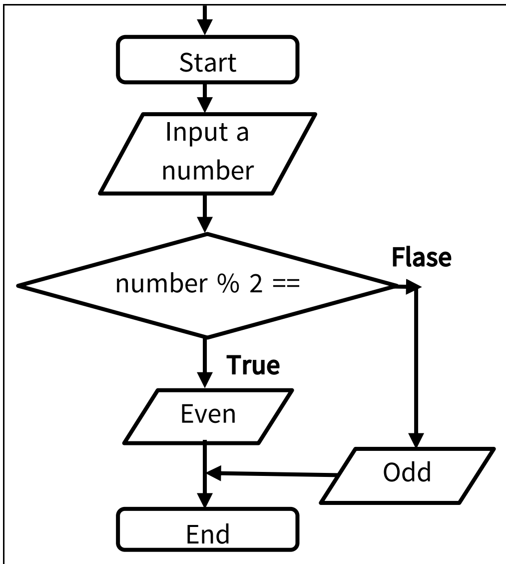
ဥပမာတွင် ထည့်ပေးသည့်ကိန်း(5)သည် သုညထက် ငယ်သောကြောင့် မည်သည့်အလုပ်မှ မလုပ် ပေးပါ။ အဘယ်ကြောင့်ဆိုသော် if statement မှန်ကန်လျှင်သာ လုပ်ရမည့်အလုပ်ကို ရေးထားပြီး မှားလျှင် လုပ်ပေးရမည့် အလုပ်ကို မရေးထားသောကြောင့် ဖြစ်သည်။

```
num = 5
print(num > 0)
if num > 0:              # သုညထက် ကြီးသည် မကြီးသည်ကို စစ်ဆေးသည်။
```

```
    print(num, "is a positive number.")
```

```
True
5 is a positive number.
```

10.2.7 If ... Statement Example-6



ဤဥပမာတွင် ထည့်ပေးသည့် ကိန်း(integer) တစ်ခုသည် စုံကိန်း(even number) ဖြစ်သည် သို့မဟုတ် မကိန်း(odd number)ဖြစ်သည်ကို ဆုံးဖြတ် ပေးသည်။ ပရိုဂရမ်ငယ် တစ်ခုကို ရှင်းပြ ထားသည်။ နံပါတ် တစ်ခုကို input အဖြစ် လက်ခံပါ။ ထိုကိန်းကို (၂)ဖြင့်စား၍ အကြွင်း မရှိလျှင် `(num %2 ==0)`, True ဖြစ်လျှင် စုံကိန်း(even number) ဖြစ်သည်။ ထိုသို့ မဟုတ်လျှင် မကိန်း ဖြစ်သည်။ တစ်နည်းအားဖြင့် False ဖြစ်လျှင် မကိန်း ဖြစ်သည်။

```
# input ကို လက်ခံ ပြီး integer အဖြစ်သို့ ပြောင်းသည်။
num  = int(input("Enter the number: "))

# num ကို (၂)ဖြင့်စားသည့် ရလဒ်သည် 0 နှင့် ညီ မညီ စစ်သည်။
if num %2 ==0:          # True ဖြစ်လျှင် "Number is even" ဟု ဖော်ပြပေးရန် ဖြစ်သည်။
       print(num, " is even number")
else:
       print(num, " is odd number")
```

```
Enter the number: 66
66 is even number
```

10.2.8 If ... Statement Example-7

စာမေးပွဲအောင် မအောင် စစ်ပေးသည် ဥပမာ ဖြစ်သည်။ အမှတ်(၆၀)နှင့် အထက်ရလျှင် စာမေးပွဲ အောင်သည်။ If တစ်ခုတည်းသာပါသည့် statement ကို သုံးထားသောကြောင့် True ဖြစ်လျှင်သာ အလုပ် လုပ်သည်။ False ဖြစ်လျှင် မည်သည့်အလုပ်မျှ လုပ်လိမ့်မဟုတ်။ ကျော်သွားလိမ့်မည်။ စာမေးပွဲကျလျှင် အောင်စာရင်းတွင် နာမည် ထည့်ပေးရန်မလိုသောကြောင့် else statement မလိုပါ။

```
marks = 85
if marks >= 60:              # အမှတ် (၆၀) ကျော် မကျော် စစ်သည်။
    print('Passed')          # အမှတ် (၆၀) နှင့် အထက်ရလျှင် စာမေးပွဲအောင်သည်။
```

```
Passed
```

10.2.9 If ... Statement Example – 8

If တစ်ခုတည်းသာပါသည့် statement ကို အသုံးပြုပြီး ထည့်ပေးသည့် ကိန်း(၃)လုံး အနက်မှ အငယ်ဆုံး ကိန်းကို ရှာသည့် ဥပမာ ဖြစ်သည်။

```
number1 = int(input('Enter first integer: '))
number2 = int(input('Enter second integer: '))
number3 = int(input('Enter third integer: '))

minimum = number1          # number1 ကို minimum အဖြစ် assign လုပ်သည်။
if number2 < minimum:      # number2 ကို minimum ထက် ငယ်လား စစ်သည်။
    minimum = number2      # ငယ်လျှင် number2 ကို minimum အဖြစ် assign လုပ်သည်။

if number3 < minimum:      # number3 ကို minimum ထက် ငယ်လား စစ်သည်။
    minimum = number3      # ငယ်လျှင် number3 ကို minimum အဖြစ် assign လုပ်သည်။

print('Minimum value is', minimum)
```

```
Enter first integer: 12
Enter second integer: 45
Enter third integer: 33
Minimum value is 12
```

10.2.10 If ... Statement Example – 9

အသက်(၆၅)နှစ် သို့မဟုတ် (၆၅)နှစ်ကျော်လျှင်(`age >= 65`) အမျိုးသမီးဖြစ်လျှင်(`gender == 'Female'`) အဖွား(senior female)ဟု သတ်မှတ်သည်။ အချက် နှစ်ချက်စလုံးနှင့် ပြည့်စုံရမည် ဖြစ်သော ကြောင့် **and** operator ကို သုံးသည်။

```
gender = 'Female'
age = 70
if gender == 'Female' and age >= 65:
    print('Senior female')
```

```
Senior female
```

10.2.11 If ... Statement Example – 10

if statement များစွာနှင့် comparison operator ကို သုံးထားသည့် ဥပမာ ဖြစ်သည်။ if statement တစ်ခုတည်းသာ ပါသောကြောင့် False ဖြစ်လျှင် ဘာမှ မလုပ်ပါ။

```
print('Enter two integers, and I will tell you','the relationships t
hey satisfy.')
# read first integer
```

```
number1 = int(input('Enter first integer: '))
# read second integer
number2 = int(input('Enter second integer: '))
if number1 == number2:
    print(number1, 'is equal to', number2)

if number1 < number2:
    print(number1, 'is less than', number2)

if number1 > number2:
    print(number1, 'is greater than', number2)

if number1 <= number2:
    print(number1, 'is less than or equal to', number2)

if number1 >= number2:
    print(number1, 'is greater than or equal to', number2)
```

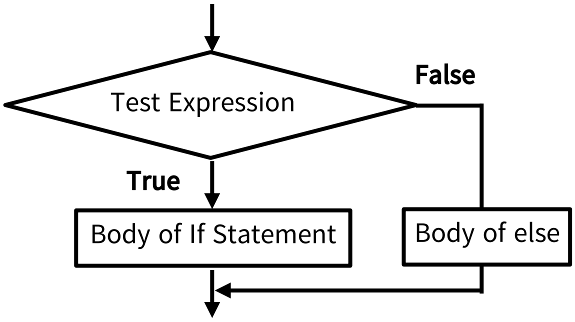
```
Enter two integers, and I will tell you the relationships they satis
fy.
Enter first integer: 77
Enter second integer: 12
77 is not equal to 12
77 is greater than 12
77 is greater than or equal to 12
```

10.3 if... else Statement

Python **if else** Statement Syntax and Statement Flowchart

if...else Statement Syntax

```
if test expression:
       statement(s)
else:
       statement(s)
```



အခြေအနေတစ်ခုသည် True ဖြစ်လျှင် အလုပ်ကိစ္စတချို့ကို ဆောင်ရွက် ရမည်ဖြစ်ပြီး False ဖြစ်လျှင် တခြားအလုပ်ကိစ္စတချို့ကို ဆောင်ရွက် ရမည်ဖြစ်သည်။(The if... else statement performs an action if a condition is True or performs a different action if the condition is False.)

Test expression မှထွက်လာသည့် ရလဒ် နှစ်ခုစလုံးတွက် သုံးလိုသည့်အခါ If..else statement ကိုသုံးသည်။ Test expression ၏ ရလဒ်သည် True ဖြစ်လျှင်(မှန်လျှင်) If အောက်ရှိ statement အတိုင်း လုပ်ပါ။ မှားလျှင်(False ဖြစ်လျှင်) else: အောက်က statement အတိုင်းလုပ်ပါ။

10.3.1 If ...else ... Statement Example-1

စုံကိန်းလား? မကိန်းလား? စစ်ပေးသည် ဥပမာ ဖြစ်သည်။

```
number = int(input("Please Enter a number:"))
if(number % 2 ==0):
    print("Even Number")
else:
    print("Odd Number")
```

```
Please Enter a number:12
Even Number
```

10.3.2 If ...else Statement Example-2

အပေါင်းကိန်းလား? အနုတ်ကိန်းလား? (positive or negative number) စစ်ပေးသည် ဥပမာ ဖြစ်သည်။

```
number = int(input("Please Enter a number: "))
if(number>0):
    print(number , "is positive number.")
else:
    print(number , "is negative number.")
```

```
Please Enter a number: 099
99 is positive number.
```

10.3.3 If ...else ... Statement Example-3

မဲပေးဘို့ အသက်ပြည့်ပြီလား?(eligible to vote)

```
age = int(input("Enter your age? "))
if age >=18:
    print("your are eligible to vote")
else:
    print("Sorry!! you have to wait!!!")
```

```
Enter your age? 17
Sorry!! you have to wait!!!
```

10.3.4 If ...else ... Statement Example - 4

စာမေးပွဲ အောင်လား? ကျလား?

```
grade = 85
if grade >= 60:
    print('Passed')
else:
    print('Failed')
```

```
Passed
```

10.3.5 If …else … Statement Example - 5

သရအက္ခရာလား? ဗျည်းအက္ခရာလား? (vowel or constant ?)စစ်ဆေးသည့် ပရိုဂရမ် ဖြစ်သည်။ Logical test သည် တစ်ခုထက် ပိုများသည့် အခြေအနေများစွာ ဖြစ်နိုင်သည်။ a, e, i, o, u စသည့် သရ အက္ခရာများကို တစ်လုံးချင်း "==" ဖြင့် စစ်သည်။ Test တစ်ခုသာ True ဖြစ်ရန်လိုသောကြောင့် or operator ဖြင့် ဆက်သည်။ Logical test များစွာ စစ်နိုင်ပုံကိုသာ ဆိုရင်းဖြစ်သည်။

```
input_char = str(input("Please Enter a Character: "))

if((input_char== 'a') or (input_char== 'A') or (input_char== 'e') or
   (input_char== 'E') or (input_char== 'i') or (input_char== 'I') or
   (input_char== 'o') or (input_char== 'O') or (input_char== 'u') or
   (input_char== 'U')):
    print("Vowel")
else:
    print("Consonant")
```

```
Please Enter a Character: E
Vowel
```

အောက်တွင် logical test သည် တစ်ခုချင်း စစ်နိုင်သလို သင့်လျော်သလို ပြင်ဆင်ပြီး စစ်စရာလိုသည့် logical test များကို အရေအတွက် နည်းသွားအောင် လျှော့ချနိုင်သည်။ a, e, i, o, u စသည့် သရအက္ခရာများကို တစ်လုံးချင်း "==" ဖြင့် စစ်မည့်အစား a, e, i, o, u တို့ကို list တစ်ခုတည်ဆောက်၍ "in" operator သုံးပြီး logical test တစ်ခုတည်းဖြင့် စစ်နိုင်ပုံကို ဖော်ပြထားသည်။

```
input_char = str(input("Please Enter a Character: "))
vowel = [ 'a', 'e', 'i','o','u', 'A', 'E', 'I', 'O', 'U']
if(input_char in vowel):
    print("Input character is Vowel.")
else:
    print("Input character is Consonant.")
```

```
Please Enter a Character: e
Input character is Vowel.
```

10.3.6 If …else … Statement Example-6

ထည့်ပေးသည့် ကိန်းနှစ်လုံးမှ ငယ်သည့်ကိန်းကို ရွေးပါ။

```
number1 = int(input("Please Enter First Number: "))
number2 = int(input("Please Enter Second Number: "))

if(number1 < number2):
    print("Min Number is: {}".format(number1))
else:
    print("Min Number is: {}".format(number2))
```

```
Please Enter First Number: 53
```

```
Please Enter Second Number: 23
Min Number is: 23
```

10.4 if... elif... else statement

စစ်ရန်အခြေအနေ(test conditions)နှင့် လုပ်ဆောင်ရန် ကိစ္စများစွာ ပါဝင်နေသည့်အခါ if... elif... else statement ကို အသုံးပြုသည်။ တချို့သော လက်တွေ့အခြေအနေများသည် အနည်းငယ် ရှုပ်ထွေးသောကြောင့် ဒါဖြစ်လျှင် ဒါလုပ်ရမည့် ဆိုသည့် ပုံသေပြောရန် မဖြစ်နိုင်သည့်အခြေအနေများရှိသည်။ ထို့ကြောင့် if... elif... else သည် လက်တွေ့တွင် အသုံးများသည်။

if..elif..else Statement အလုပ်လုပ်ပုံ

(၁) ပထမဦးစွာ if statement အတွက် condition တစ်ခုကို စစစ်သည်။ True ဖြစ်လျှင် if statement အောက်က ကိစ္စကို ဆောင်ရွက်(execute)သည်။ False ဖြစ်လျှင် elif statement ကို စစ်သည်။

(၂) elif statement မှ ရလဒ်သည် True ဖြစ်လျှင် elif statement အောက်က ကိစ္စကို ဆောင်ရွက်(execute)သည်။ False ဖြစ်လျှင် else statement မှ ဆီသို့ရောက်သည်။

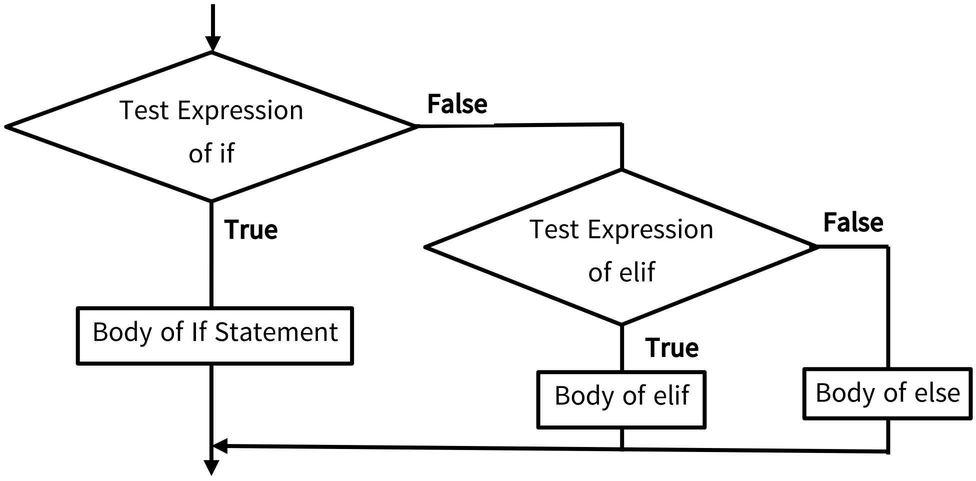
if..elif..else statement Syntax

ပထမ test expression မှန်လျှင် အောက်က အတိုင်းလုပ်ပါ။ မှားလျှင် နောက်ထပ် test expression တစ်ခုကို elif: ဖြင့် ထပ်စစ်ပြီး မှန်လျှင် elif: အောက်က statement အတိုင်းလုပ်ပါ။ မှားလျှင် else အောက်က statement အတိုင်းလုပ်ပါ။

```
if test expression:
       statement(s)

elif test expression:
       statement(s)
else test expression:
       statement(s)
```

Python if..elif..else Statement Flowchart

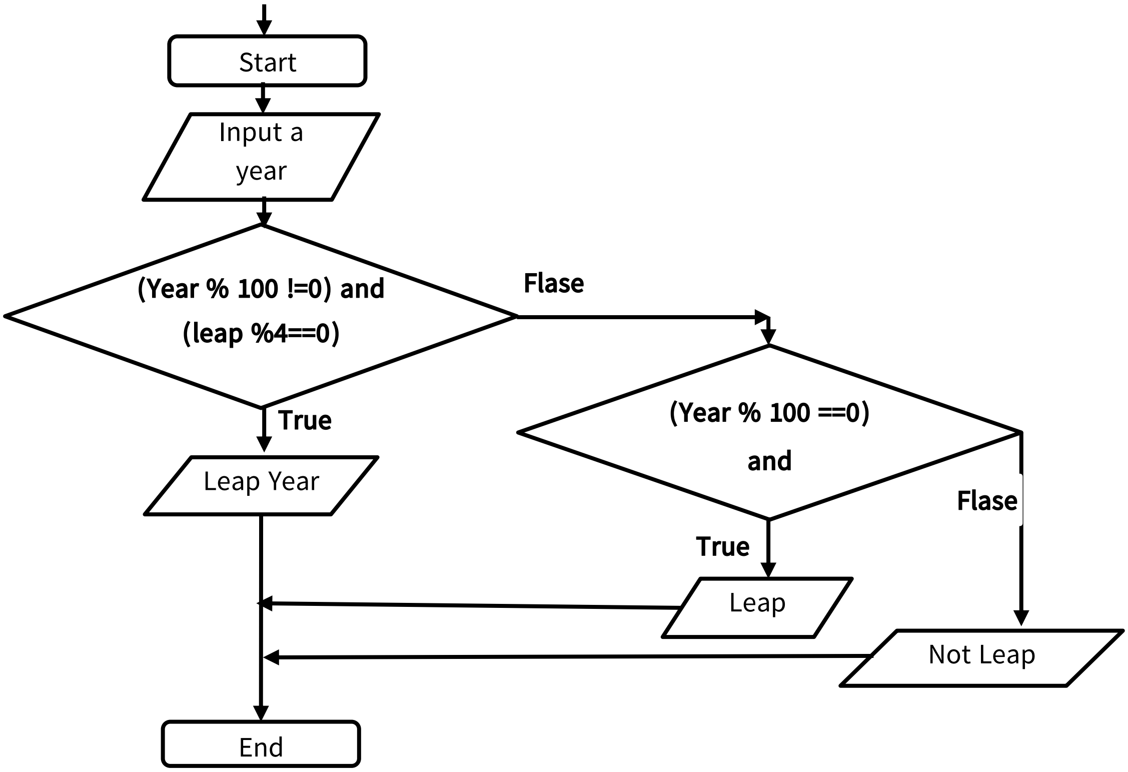


10.4.1 If elif else Statement Example-1

if..elif statement example 2 ကို ပိုကောင်းစေရန် ထပ်ဖြည့်ထားသည့် ဥပမာ ဖြစ်သည်။ ထည့်ပေးသည့်ကိန်း(integer) တစ်ခုသည် အပေါင်းကိန်း(positive number)ဖြစ်သည် သို့မဟုတ် အနုတ်ကိန်း (negative number) ဖြစ်သည်ကို ဆုံးဖြတ်ပေးသည့် အပေါ်ကပရိုဂရမ် ပိုပြည့်စုံအောင် test condition တစ်ခု ထည့်ဖြည့်ထားသည်။ သုည(Zero)သည် စုံကိန်း(even number)လည်းမဟုတ် မကိန်း(odd number) မဟုတ် သောကြောင့် အခြေအနေ တစ်ခုထပ်ဖြည့်ရန် လိုသည်။ ထို့ကြောင့် If elif else statement ကိုသုံး၍ ဖော်ပြထားသည်။

```
num = int(input('Enter a number: '))
if num > 0:
    print('The number is positive')
elif num < 0:
    print('The number is negative')
else:
    print('The number is zero')
```

```
Enter a number: 86
The number is positive
```



10.4.2 If elif else Statement Example - 2

ရက်ထပ်နှစ်လား? (Leap year?) ခုနှစ်ကို (၄)နှင့်စား၍ ပြတ်လျှင် ရက်ထပ်နှစ် ဖြစ်သည်။

```
year = int(input("Please enter Year: "))

if(year%100 != 0 ) and (year%4 ==0):
    print("leap Year")
elif(year % 100 == 0) and (year % 400 == 0):
    print("leap Year")
else:
    print("Not Leap Year")
```

```
Please enter Year: 2020
leap Year
```

10.4.3 If elif else Statement Example-3

သစ်သီးလား ဟင်းသီးဟင်းရွက်လား? (fruit or vegetable?) စစ်ဆေးသည့် ဥပမာ ဖြစ်သည်။

```
# Test expression တွင် test condition များစွာပါသည်။
```

```
a = str(input())
if (a == 'banana' or a == 'apple' or a == 'kiwi' or
    a == 'cherry' or a == 'lemon' or a == 'grapes'):
    print(a , " is fruit.")
elif (a == 'tomato' or a == 'cucumber' or
      a == 'pepper' or a == 'carrot'):
    print(a ," is vegetable.")
else:
    print("unknown")banana
banana  is fruit.
```

```
# Test expression တွင် test condition တစ်ခုသာ ပါသည်။
a = str(input())
fruit = [ 'banana', 'apple', 'kiwi', 'cherry',
          'lemon', 'grapes']
vegetable = [ 'tomato', 'cucumber', 'pepper', 'carrot']

if a in fruit:
    print(a , " is fruit.")
elif a in vegetable:
    print(a ," is vegetable.")
else:
    print("unknown")
```

```
tomato
tomato is vegetable.
```

10.4.4 If elif else Statement Example-4

Weekend နှင့် working day ခွဲခြားသည့် ဥပမာ-

```
a = str(input())
if a == 'Monday' or a == 'Tuesday' or a == 'Wednesday' or a == 'Thur
sday' or a == 'Friday':
    print("Working day")
elif a == 'Saturday' or a == 'Sunday':
    print("Weekend")
else:
    print("Error")
```

```
Sunday
Weekend
```

10.4.5 If elif else Statement Example-5

ကျောင်းသားတစ်ယောက်၏ ရမှတ်များကို မူတည်၍ grading ခွဲပေးသည့် ဥပမာ ဖြစ်သည်။

အမှတ်(၈၅)နှင့် (၁၀၀) အကြားရလျှင် grade A ရသည်။

အမှတ်(၆၀)နှင့် (၈၅) အကြားရလျှင် grade B+ ရသည်။

အမှတ်(၄၀)နှင့် (၆၀) အကြားရလျှင် grade B ရသည်။ စသည့်ဖြင့် grading များ ခွဲသည်။

```
marks = int(input("Enter the marks?"))
if marks >=85 and marks <=100:
    print("Congrats1 your scored grade A...")

elif marks >60 and marks <=85:
    print("you scored grade B+...")

elif marks >40 and marks <=60:
    print("you scored grade B...")

elif marks >30 and marks <=40:
    print("you scored grade C...")

else:
    print("Sorry you have failed")
```

```
Enter the marks?75
you scored grade B+...
```

10.4.6 If elif else Statement Example - 6

ရုပ်ရှင်ရုံဝင်ကြေးလက်မှတ်ခသည် နေ့ကိုလိုက်၍ ပြောင်းလဲသည်။ တနင်းလာနေ့၊ အင်္ဂါ နေ့ နှင့် သောကြာ နေ့များတွင် ဈေးသက်သာသည်။(2000ကျပ်)။ ဗုဒ္ဓဟူးနေ့နှင့် ကြာသပတေးနေ့တွင် ဈေးအသင့်အတင့် ဖြစ်သည်။ (4000ကျပ်)။ စနေ့နေ့နှင့် တနင်္ဂနွေနေ့များတွင် ဈေးကြီးသည်။ (6000ကျပ်)။ စသည့် နှုန်းများ

အတွက် တွက်ပေးသည့် if elif statement ကို အောက်တွင် ဖော်ပြထားသည်။ ရက်အမည်ကို string ဖြင့် ထည့်ပေးပါက ရုပ်ရှင်လက်မှတ်ဈေးနှုန်းကို တွက်ပေးသည့် ဥပမာ ဖြစ်သည်။

```
a = str(input())
if a == 'Monday' or a == 'Tuesday' or a == 'Friday':
    print(2000)
elif a == 'Wednesday' or a == 'Thursday':
    print(4000)
elif a == 'Saturday' or a == 'Sunday':
    print(6000)
```

```
Sunday
6000
```

10.4.7 If elif else Statement Example-7

Prime number ဟုတ် မဟုတ် စစ်သည့် ဥပမာ ဖြစ်သည်။ စစ်မည့်ကိန်းကို input အဖြစ် ထည့်ပေးသည်။

```
# checking for prime number
number = int(input("Enter any number: "))

# prime is always greater than 1
if number > 1:
    for i in range(2, number):
        if (number % i) == 0:
            print(number, " is not a prime number")
            break
    else:
        print(number, "is a prime number")
#if entered number is less than or equal to 1, then it is not prime
else:
    print(number, "is not a prime number")
```

```
Enter any number: 3
3 is a prime number
```

10.5 Python Nested if statements

if statement တစ်ခုအတွင်း၌ တခြားသော if statement များကို ထည့်သွင်းရေးသားနိုင်သည်။ ထိုသို့ if statement များ အထပ်ထပ်ရေးထားခြင်းကို Nested if statement ဟုခေါ်သည်။ အောက်တွင် Nested if statement ဥပမာ တစ်ခုကို ဖော်ပြထားသည်။

10.5.1 Nested if statements Example-1

အောက်တွင် အရောင်းဝန်ထမ်းတစ်ယောက် ရရှိမည့် ကော်မရှင် ပမာဏကို တွက်ပေးမည့် ဥပမာကို ရှင်းပြ ထားသည်။ ရောင်းသည့်မြို့ကို လိုက်၍ ကော်မရှင်ရာခိုင်နှုန်း ကွဲပြားသည်။ ရောင်းအားပေါ် မူတည်၍လည်း ကော်မရှင်ရာခိုင်နှုန်း ကွဲပြားသည်။

- ရန်ကုန်တိုင်း သို့မဟုတ် မန္တလေးတိုင်း အတွက် ရောင်းအား ၅သိန်းနှင့် အောက်ဖြစ်လျှင် ၅% ဖြစ်သည်။ ၅သိန်းနှင့် ၁၀သိန်းအကြားတွင် ၇% ဖြစ် စသည်ဖြင့် အသီးသီးကွဲပြားသည်။
- ကျန်ဒေသများအတွက် ရောင်းအား ၅သိန်းနှင့် အောက်ဖြစ်လျှင် ၅.၅% ဖြစ်သည်။ ၅သိန်းနှင့် ၁၀သိန်း အကြားတွင် ၈% ဖြစ် စသည်ဖြင့် အသီးသီးကွဲပြားသည်။

ပထမဦးစွာ if statement ဖြင့် Yangon, Mandalay နှင့် ကျန်မြို့များကို ခွဲသည်။ ထို့နောက် ရောင်းအားဖြင့် ကော်မရှင်ပမာဏ အသီးသီးကို တွက်သည်။

```
city = str(input())
sales = float(input())

if city == 'Yangon' or city == 'Mandalay':
    if 0 <= sales <= 500000:
        print(f'{(sales * 0.05):.2f}')
    elif 500 < sales <= 1000000:
        print(f'{(sales * 0.07):.2f}')
    elif 1000 < sales <= 10000000:
        print(f'{(sales * 0.08):.2f}')
    elif sales > 10000000:
        print(f'{(sales * 0.12):.2f}')
    else:
        print("error")

elif city == 'other':
    if 0 <= sales <= 500:
        print(f'{(sales * 0.055):.2f}')
    elif 500 < sales <= 1000:
        print(f'{(sales * 0.08):.2f}')
    elif 1000 < sales <= 10000:
        print(f'{(sales * 0.12):.2f}')
    elif sales > 10000:
        print(f'{(sales * 0.145):.2f}')
    else:
        print("error")
else:
    print("error")
```

```
Yangon
9300000
744000.00
```

10.5.2 Nested if statements Example-2

ကုန်တင်မောင်းသူများ၏ ဝင်ငွေ(ဒရိုင်ဘာကြေး)သည် မောင်းရသည့် ရာသီ(နွေ, မိုး, ဆောင်း)နှင့် အကွာအဝေး အပေါ် မူတည်သည်။

- 'Spring' သို့မဟုတ် 'Autumn' တွင် တစ်လလျှင် ကီလိုမီတာ 5000 အောက်မောင်းပါက ထိုလ ဝင်ငွေသည် (km_per_month = 0.75 *4 * 0.9) ဒေါ်လာ ဖြစ်သည်။
- 5000 နှင့် 10000 အကြားမောင်းခဲ့လျှင် ထိုလဝင်ငွေသည် (km_per_month = 0.95 * 4 * 0.9) ဒေါ်လာ ဖြစ်သည်။ 'Summer' နှင့် 'Winter' တို့တွင် တွက်သည့်နှုန်းများ မတူကြပေ။ စသည်ဖြင့် ရာသီနှင့် အကွာအဝေး အသီးသီးတွက်ယူပုံကို Nested if statement သုံးထားသည်။

```
season = str(input())
km_per_month = float(input())

if season == 'Spring' or season == 'Autumn':
    if km_per_month <= 5000:
        km_per_month *= 0.75 * 4 * 0.9
    elif 5000 < km_per_month <= 10000:
        km_per_month *= 0.95 * 4 * 0.9
    elif 10000 < km_per_month <= 20000:
        km_per_month *= 1.45 *4 * 0.9

elif season == 'Summer':
    if km_per_month <= 5000:
        km_per_month *= 0.9 * 4 * 0.9
    elif 5000 < km_per_month <= 10000:
        km_per_month *= 1.1 * 4 * 0.9
    elif 10000 < km_per_month <= 20000:
        km_per_month *= 1.45 * 4 * 0.9

elif season == 'Winter':
    if km_per_month <= 5000:
        km_per_month *= 1.05 * 4 * 0.9
    elif 5000 < km_per_month <= 10000:
        km_per_month *= 1.25 * 4 * 0.9
    elif 10000 < km_per_month <= 20000:
        km_per_month *= 1.45 * 4 * 0.9

print(f'{km_per_month:.2f}')
```

```
Winter
11000
57420.00
```

-End-