

# DECISION STRUCTURES

# Need for Decision Structures

- An algorithm is a complete sequence of actions to solve a problem if followed exactly. Algorithms are built using three basic building blocks: Sequence, Selection, and repetition.
- Sequence: A control flow where the statements are executed in a progressive order, without decisions or repetition.
- Selection/Decision or Branching: A decision is made like condition checking to decide the execution of one or more statements.
- Iteration/Repetition or Loop: A loop is one or more instructions the computer repeatedly performs.

# Forming Conditions

- In Python, the conditions are test expressions. The outcome of the test expression may be two or more outcomes. The decision depends upon the variables' values or test expressions.
- Test Expressions can be, A Boolean Value, Expression involving relational operators> Expressions that the logical operators combine.
- There are three types of statements, if, if-else, and if-elif.

# Conditions using a Boolean Variable

- Illustration of Boolean Variable.

```
>>> leapYear = True
>>> leapYear
True
>>>
```

Illustration of bool function in Interactive shell.

```
>>> bool(0)
False
>>> bool('This is string')
True
>>> x = 5
>>> bool(x)
True
>>>
```

## Illustration of Relational Operators

```
>>> x = 5
>>> bool(x)
True
>>> x >= 9
False
>>> hour = 9
>>> meeting = hour > 12
>>> meeting
False
>>>
```

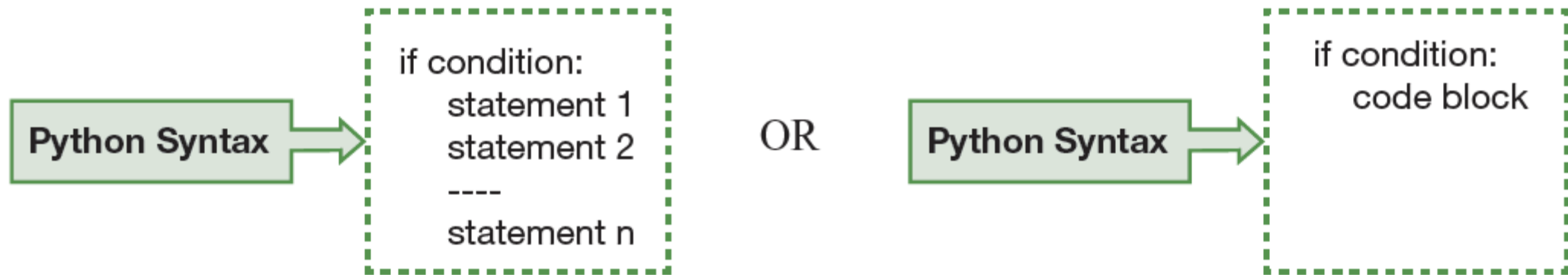
## Illustration of Chained Condition

```
>>> not False
True
>>> True and False
False
>>> True or False
True
>>> True or 0
True
>>>
```

# if-Statement

if-statement is one of the simplest forms of the decision control statement. It is a selection control statement that tests a test expression or a condition and executes a statement or many statements (Suite) based on the condition

- The syntax of the one-way decision statement can be given as,



## Notes

In other programming languages, the statements usually start with begin and end or brackets { and } would be used. In Python, this is done by indentation. At least four spaces should be given to mark the presence of a statement or suite of if—statements. If indentation is not proper, Python Interpreter would report an error.

# Illustration of if-statement in Interactive mode

Let us assume that a school conducts an entrance examination for 500 marks. This year the cutoff marks for admission is 400. If a student scores more than the cutoff marks, he would be admitted. The task is to print admission is granted if this condition is satisfied.

The pseudocode for this program would be like this

Step 1:	Start.
Step 2:	cutoff mark = 400
Step 3:	Get the school entrance examination mark say, scoreMark, the user.
Step 4:	if the entrance examination mark is greater than the cutoff mark, print admission is granted.
Step 5:	End.

The code would be as follows in interactive mode.

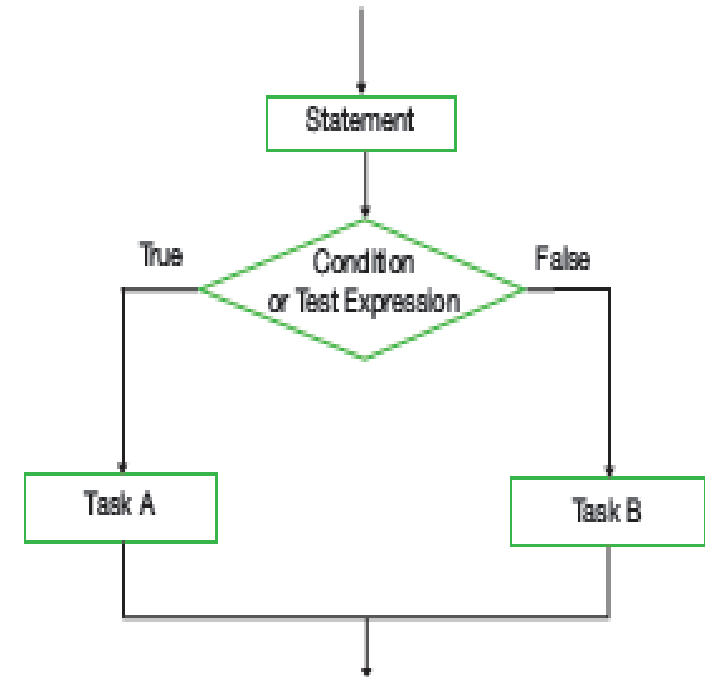
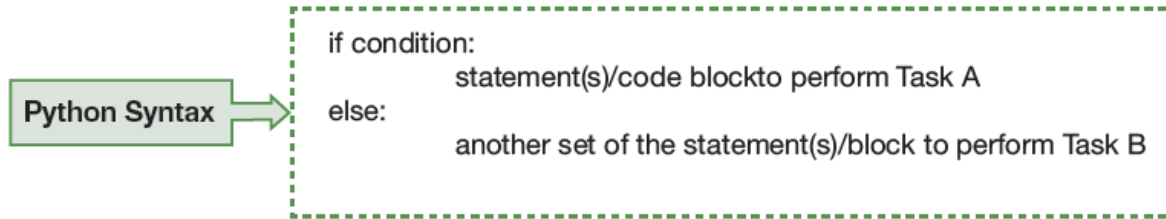
```
>>> cutoff = 400
>>> scoreMark = 425
>>> if scoreMark > cutoff:
... print('Admission is Granted')
...
Admission is Granted
>>>
```

if the condition fails and does not print anything.

# The if-else Statement

This is done by a two-way decision statement known as if-else. In this statement, If the condition is not satisfied, an alternative statement or suite specified by else is executed

- The syntax of if-else is as follows,





# Illustration of if-else statement

Consider example 5. Admission is granted if the entrance examination mark is greater than the cutoff mark. What if the entrance examination mark is lesser than the cutoff mark? In that case, a message should be printed as admission is not granted. For these two-way decisions, the if-else statement is useful. The pseudocode for this is given as follows:

Step 1:	Start.
Step 2:	Cutoff mark = 400
Step 3:	Get the entrance examination mark from the user.
Step 4:	If the entrance examination mark is greater than the cutoff mark, print admission is granted.
Step 5:	Else, print admission is not granted.
Step 6;	End.

The corresponding python script is shown below:

```
scoredMark=int(input("Enter entrance exam mark:"))
cutoff = 400
if scoredMark > cutoff:
    print("admission is granted")
else:
    print("admission is not granted")
```

```
C:\Users\Usr>python Listing2.py
Enter entrance exam mark:425
admission is granted
```

# Inline if-statement

- An inline if-statement is a convenient form of representing an if-else statement. This is also known as the ternary operator. This statement allows one to execute conditional if statements in a single line.
- The General inline statement is given as follows,

Python Syntax

expression1 if condition else expression2

If the condition is True, then expression1 is executed, otherwise, expression2 is executed. This ternary operator can be nested also.

Python Syntax

expression1 if condition1 else expression2 if condition2 else expression3

Here, the condition1 is evaluated. If it returns True, then expression1 is executed. If condition1 is False, then condition2 is evaluated. If condition2 is True, then expression2 is carried out, Otherwise, if condition False, expression3 is returned.

# If-elif statement

These are used when more than one comparison needs to be made. It primarily checks the condition. The statements are executed if the condition is true; otherwise, the control is transferred to another block. This statement is also known as the nested-if statement.

Python Syntax



```
if condition 1:  
    code block1  
elif condition 2:  
    code block2  
...  
elif condition n:  
    code blockn  
else  
    else-code block
```

Suppose a condition is evaluated first. If the value is true, statement block 1 is executed; otherwise, the following conditions in elif are evaluated, and the respective statement blocks are executed. If none of the conditions is satisfied, the other block is executed. This chain of conditions is also known as a ladder.

# Illustration of if-elif statement

A university department awards grades based on the marks, as shown in the following Table 4.1.

**Table 4.1: Grade Table**

Marks	Equivalent Grade
>90	S
80-89	A
70-79	B
60-69	C
55-59	D
50-54	E
< 50	U

The python program for the grade allocation can be done by the python program as shown below:

```
marks = int(input('Enter the marks'))
if(marks>=90):
    print("O")
elif (marks>79 and marks <=89):
    print('A')
elif(marks> 69 and marks <=79):
    print('B')
elif(marks > 60 and marks <= 60):
    print('C')
elif (marks > 55 and marks <=59):
    print('D')
elif (marks >50 and marks <= 54):
    print('E')
else:
    print('F')
```

```
C:\Users\Usr>python Listing7.py
Enter the marks34
F
```

# Nested-if conditions

Instead of using multiple if statements, it is possible to place one if-statement inside another if –statement. This is known as nested-if statements.

- Finding a maximum of three numbers Let us consider one more example. Let us try to implement the following algorithm for finding the largest of the three numbers. The python script is self-explanatory and is given below:

```
num1 = int(input('Enter the first number'))
num2 = int(input('Enter the Second number'))
num3 = int(input('Enter the Third number'))
if (num1 > num2) and (num1 > num3):
    print('The largest number is', num1)
else:
    if (num2 > num3):
        print('num2 is the largest', num2)
    else:
        print('num3 is the largest', num3)
```

```
C:\Users\Usr>python Listing9.py
Enter the first number23
Enter the Second number56
Enter the Third number78
num3 is the largest 78
```