SNS COLLEGE OF ENGINEERING
Kurumbapalayam (Po), Coimbatore – 641 107

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

19IT103 – COMPUTATIONAL THINKING AND PYTHON PROGRAMMING

- **A readable, dynamic, pleasant, flexible, fast and powerful language**

UNIT III - CONTROL FLOW, FUNCTIONS, STRINGS

Conditionals:Boolean values and operators, conditional (if), alternative (if-else),chained conditional (if-elif-else);Iteration: state, while, for, break, continue, pass; Fruitful functions: return values,parameters, local and global scope, function composition, recursion; Strings: string slices,immutability, string functions and methods, string module; Lists as arrays. Illustrative programs: square root, gcd, exponentiation, sum an array of numbers, linear search, binary search.

3.1 Conditionals

3.1.1 Boolean Values and Operators

- A Boolean expression is an expression that is either true or false.
- Converting Boolean to integer, the value is always 0 for false and 1 for true
- Converting integer to Boolean, the value is True for all integers except zero
- Python type is bool.

3.1 Conditionals

3.1.1 Boolean Values and Operators

Boolean Expression	Meaning
x != y	x is not equal to y
x > y	x greater than y
x<y	x less than y
x<=y	x less than or equal to
x>=y	x greater than or equal to

3.1 Conditionals

3.1.1 Boolean Values and Operators

and Operator:

Op1	Op2	Op1 and Op2
TRUE	TRUE	TRUE
TRUE	FALSE	FALSE
FALSE	TRUE	FALSE
FALSE	FALSE	FALSE

3.1 Conditionals

3.1.1 Boolean Values and Operators

or Operator:

Op1	Op2	Op1 or Op2
TRUE	TRUE	TRUE
TRUE	FALSE	TRUE
FALSE	TRUE	TRUE
FALSE	FALSE	FALSE

3.1 Conditionals

3.1.1 Boolean Values and Operators

Example 1: “and” operator

```
print("Enter Month and Day")
month=input()
day=int(input())
if(month =="january" and day ==26):
    print ("republic day")
elif(month =="august" and day ==15):
    print ("independence day")
elif(month =="october" and day ==2):
    print ("Gandhi Jayanthi")
else:
    print ("invalid input")
```

Output:

```
Enter Month and Day
august
15
independence day
>>>
```

3.1 Conditionals

3.1.1 Boolean Values and Operators

Example 1: “or” operator

```
import sys
month=input("Enter the month to display number of days: ")
if(month=="may" or month=="july" or month=="august"):
    print ("31")
elif(month=="jun"):
    print ("30")
else:
    print ("invalid")
```

Output:

```
Enter the month to display number of days: may
31
```

3.1 Conditionals

3.1.2 Conditional Statements

- In programming language conditional statements are used to perform different computations or actions depending on whether a condition evaluates to true or false
- The conditions use comparisons and arithmetic expressions with variables
- The expressions are evaluated to Boolean values True or False

3.1 Conditionals

3.1.2 Conditional Statements

Python conditional statements are

1. if statement
2. If..else statement
3. If..elif..else statement
4. Nested if statement

3.1 Conditionals

3.1.2 Conditional Statements

Rules for conditional statements:

- The colon(:) is required at the end of the condition
- The body of the if statement is indicated by the indentation(four spaces are used for indentation)
- Python interprets non-zero values as true and 0 as false

3.1 Conditionals

3.1.2 Conditional Statements

1. 'if' Statement :

- Sometimes we want to execute a code or a block of code only if a certain condition is satisfied.
- The program evaluates the condition and will execute statement(s) only if the condition is True.
- If the condition is False, the statement(s) is not executed.
- In Python, the body of the if statement is indented.
- Python interprets non-zero values as True. None and 0 are interpreted as False.

3.1 Conditionals

3.1.2 Conditional Statements

Syntax:

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

Example:

```
print("Enter Your age")
n=int(input())
if(n>=18):
    print("Eligible for voting")
```

```
Enter Your age
20
Eligible for voting
>>>
```

3.1 Conditionals

3.1.2 Conditional Statements

2. ‘if...else’ Statement

- The if..else statement evaluates test expression and will execute body of if only when test condition is True.
- If the condition is False, body of else is executed. Indentation is used to separate the blocks.

3.1 Conditionals

3.1.2 Conditional Statements

2. 'if...else' Statement

Syntax:

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

3.1 Conditionals

3.1.2 Conditional Statements

2. 'if...else' Statement

Example 1:

```
num = 3
if num >= 0:
        print("Positive or Zero")
else:
        print("Negative number")
```

Output:

```
Positive or Zero
>>>
```

3.1 Conditionals

3.1.2 Conditional Statements

2. 'if...else' Statement

Example 2:

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

Output:

```
Enter a number
23
Odd Number
>>>
```

3.1 Conditionals

3.1.2 Conditional Statements

3. Chained conditional - if...elif...else

- The elif is short for else if. It allows us to check for multiple expressions.
- If the condition for if is False, it checks the condition of the next elif block and so on.
- If all the conditions are False, body of else is executed.
- Only one block among the several if...elif...else blocks is executed according to the condition.
- The if block can have only one else block. But it can have multiple elif blocks.

3.1 Conditionals

3.1.2 Conditional Statements

3. Chained conditional - if...elif...else

Syntax

```
if (test expression/condition):
        Body of if
elif (test expression/condition):
        Body of elif
else:
        Body of else
```

3.1 Conditionals

3.1.2 Conditional Statements

3. Chained conditional - if...elif...else

Example 1:

```
num = 3.4
if num > 0:
    print("Positive number")
elif num == 0:
    print("Zero")
else:
    print("Negative number")
```

Output:

```
Positive number
>>>
```

3.1 Conditionals

3.1.2 Conditional Statements

3. Chained conditional - if...elif...else

Example 2:

```
n=int(input("Enter a number between seven and ten: "))
if(n==7):
    print("heptagon")
elif(n==8):
    print("octogon")
elif(n==9):
    print("nanogon")
elif(n==10):
    print("decagon")
else:
    print("input should be from 7 to 10")
```

Output:

```
Enter a number between seven and ten: 8
octogon
```

3.1 Conditionals

3.1.2 Conditional Statements

4. Nested Conditional

- A conditional statement defined inside another conditional statement is called nested conditional statement.
- Any number of these statements can be nested inside one another.
- Indentation is the only way to figure out the level of nesting.
- Similarly, alternative and chained conditionals can also be nested

3.1 Conditionals

3.1.2 Conditional Statements

4. Nested Conditional

Example:

```
num = float(input("Enter a number: "))
if num >= 0:
    if num == 0:
        print("Zero")
    else:
        print("Positive number")
else:
    print("Negative number")
```

Output:

```
Enter a number: 12
Positive number
>>>
```

Summary

- A Boolean expression is an expression that is evaluated as either true or false.
- Two boolean operators are ‘and’ and ‘or’.
- If statement executes its body only when it is true.
- To execute alternative statements when a condition fails, if-else is useful
- If-elif-else is used to check multiple conditions
- Conditionals inside conditional is said to be nested conditional


THANK YOU