PMDS508L - Python Programming
Control Structures

Dr. B.S.R.V. Prasad
Department of Mathematics
School of Advanced Sciences
Vellore Institute of Technology
Vellore

srvprasad.bh@gmail.com (Personal)
srvprasad.bh@vit.ac.in (Official)
+91-8220417476

Python Conditions

Python supports the usual logical conditions from mathematics:

- Equals: $a == b$
- Not Equals: $a \,!= b$
- Less than: $a < b$
- Less than or equal to: $a <= b$
- Greater than: $a > b$
- Greater than or equal to: $a >= b$

If Statements in Python

```
if <condition>:
    statements to be executed if condition is true
elif <conditon>:
    statements to be executed if else condition is true
else:
    statements to be executed when all the above
  conditions fails
```

Short Hand If and If...Else

- If you have only one statement to execute, one for if, and one for else, you can put it all on the same line:

```
a = 20
b = 330
print("a > b") if a > b else print("b > a")
```

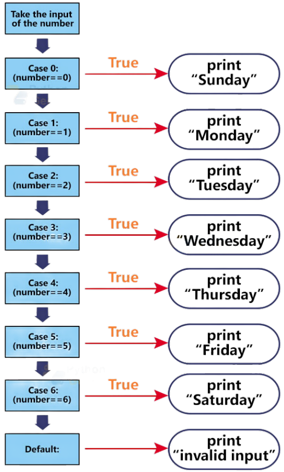
- You can also have multiple else statements on the same line:

```
a = 330
b = 330
print("a > b") if a > b else print("a = b") if a ==
   b else print("b > a")
```

Combining the conditions and Nested If

- To check if two conditions are satisfied or not then we need to use the keyword and
- To check if any one of the wto conditions are satisfied or not then we need to use the keyword or
- Nested Loops can also be used in the Python but we need to take care of the spacing.
- `pass` statement.
    - The `if` statements cannot be empty. If it empty then we need to use `pass` statement to avoid getting an error.

Multi-branching



Multi-branching

if...elif...else

```python
no = int(input('Enter the number'))
if no == 0:
    print('Sunday')
elif no == 1:
    print('Monday')
.
.
.
elif no == 6:
    print('Saturday')
else:
    print('Invalid Input')
```

Multi-branching
`match-case`

- Developers coming from languages like C/C++ or Java know that there is a conditional statement known as a Switch Case using which above code can be coded.
- In Python this is `match-case` and was introduced in Python 3.10.
- The match case statement in Python is initialized with the match keyword followed by the parameter to be matched.
- Then various cases are defined using the case keyword and the pattern to match the parameter.
- The "_" is the wildcard character that runs when all the cases fail to match the parameter value.

Multi-branching

match-case

```
match parameter:
    case pattern1:
        # code for pattern 1
    case pattern2:
        # code for pattern 2
    .
    .
    .
    case patterN:
        # code for pattern N
    case _:
        # default code block
```

Multi-branching

match-case

```
no = int(input('Enter the number'))
match no:
    case 0:
        print('Sunday')
    case 1:
        print('Monday')
    .
    .
    .
    case 6:
        print('Saturday')
    case _:
        print('Invalid Input')
```

Python Loops

Python has two primitive loop commands:

- `while` loops
- `for` loops

While Loops

```
i=1
while i<10:
    print(i)
    i += 1
```

break Statement can be used to come out of the while loop even if the condition is true

```
i=1
while i<10:
    print(i)
    if i==3:
        break
    i += 1
```

While Loops

continue Statement can be used to stop the current iteration and go to next iteration.

```
i=1
while i<10:
    print(i)
    if i==3:
        continue
    i += 1
```

While Loops

With the else statement we can run a block of code once when the condition no longer is true:

```
i=1
while i<10:
    print(i)
    i += 1
else:
    print("i is no longer greater than 10")
```

Python For Loops

- A `for` loop is used for iterating over a sequence (that is either a list, a tuple, a dictionary, a set, or a string)
- This is much like the `for` keyword in other programming languages, and works more like an iterator method as found in other object-orientated programming languages.
- With the `for` loop we can execute a set of statements, once for each item in a list, tuple, set etc.
- The `for` loop does not require an indexing variable to set beforehand.

Python For Loops
The range() Function

The `range()` function returns a sequence of numbers, starting from 0 by default, and increments by 1 (by default), and ends at a specified number (excluding the specified number).

- `range(5)` - Returns the values starting from 0 to 5 increment by 1.
- `range(a,b)` - Returns the values starting from a to $b - 1$ increment by 1.
- `range(a,b,n)` - Returns the values starting from a to $b - 1$ increment by n.

Python For Loops

```python
for x in range(1,5):
  print(x)
```

```python
for x in "Banana":
  print(x)
```

Python For Loops
break and continue statements

```python
for x in range(1,10,2):
  if x == 5:
    break
  print(x)
```

```python
for x in range(1,10,2):
  if x == 5:
    continue
  print(x)
```

Python For Loops
Else in For Loop

The `else` keyword in a `for` loop specifies a block of code to be executed when the loop is finished:

```
for x in range(6):
    print(x)
else:
    print('For loop finished!')
```

Nested For Loops

```python
for x in range(1,10):
  for y in (x,x+2):
    print(x, y)
```

Python For Loops
The pass Statement

for loops cannot be empty, but if you for some reason have a for loop with no content, put in the pass statement to avoid getting an error.

```
for x in range(1,5):
  pass
```

Python Loops
Sample Programs

Program

Print all the numbers which are divisible by 2 between any two given numbers

Python Loops
Sample Programs

```python
num1 = int(input("Enter the first number: "))
num2 = int(input("Enter the second number: "))

for x in range(num1,num2+1):
    if x%2 == 0:
        print(x," is divisible by 2")
```

Python Loops
Sample Programs

Program

Check whether a given number is prime or not?

Python Loops
Sample Programs

```
num = int(input("Enter the number: "))

if num <= 1:
    print("Entered number should be greater than 1")
else:
    for x in range(2,num//2+1):
        if num%x == 0:
            print(num," is not prime")
            break
    else:
        print(num," is prime")
```

Python Loops
Exercise Programs

Program

Write a Python program which accepts two numbers from the user and prints all the prime numbers between them.

Program

Write a Python program which accepts a number from the user and checks whether it is palindrome or not?

Program

A number N is said to Armstrong number if N equal to sum of the cubes of each digit in that number. For example 153 is Armstrong number as $153 = 1^3 + 5^3 + 3^3$.

Write a Python program which checks whether a given number is Armstrong number or not?

Python Loops
Exercise Programs

Program

A positive integer N is said to be perfect if N is equal to the sum of its proper divisors.

Write a Python program to check whether a given positive integer is prefect or not?

Program

Write a Python program to find the factorial of a given number.

Program

Write a Python program to which accepts number of lines and then prints one $*$ in the first line, two $*$'s in the second line, etc...