Python Conditionals, Loops and Functions

Kunal Raghav

2018-08-31

Contents

- conditionals (if - elif - else)
- loops (for,while)
- functions

Conditionals

Definition

These are used to change the flow of a program if a particular condition is satisfied.

Syntax

```
if <condition>:
    <statement1>
    <statement2>
    .
    .
elif <condition2>: #elif and else blocks are optional
    <statement1>
    <statement2>
    .
    .
.
.
.
.
elif <conditionN>:
    <statement1>
    <statement2>
    .
    .
else:
    <statement1>
    <statement2>
    .
    .
```

Example

```python
# checking if a person is eligible for a driver's licence

age=int(input("Enter age: "))

if age<=10:
    print("pehle cycle chalana seekh.")

elif age < 18 :
    print("You can't get a driver's licence.")

elif age>=18:
    print("Congratulations, you are eligible for a driver's licence.")

else:
    print("What age did you enter to get this statement as output?")
```

Loops

What if you need to repeat a few lines of code multiple times. This is where loops are used. Loops are of 2 types:

- while loops
- for loops

While Loops

Definition

While loops are used to repeat a few lines of code as long as a particular condition is satisfied.

Syntax

```
while <condition>:
    |
    | <statements>
    |
    | #modify <condition> at the end
```

Example

```python
# appending n items in a list.

l=list()
n = int(input("Enter number of elements: "))
while n>0:
    item=input("enter item: ");      #getting item from user
    l.append(item)                   #appending item to list
    n=n-1                            #reducing n by 1 after
                                     #appending
```

For Loops

Definition

To repeat a few lines of code for pre defined number of times. Keywords:

- **iterator**: a temporary variable used to store the current iteration/ no. of times the loop has been executed.
- **range**: an inbuilt function used to create an immutable list.

```python
a=range[start value, stop value, step value]
a=range(10) # is analogous to [0,1,2,3,4,5,6,7,8,9]
a=range(1,10) # is analogous to [1,2,3,4,5,6,7,8,9]
a=range(1,10,3) # is analogous to [1,4,7]
```

Syntax

```
for <iterator variable> in range(n):
    |
    |   <statements>
    |
    |
```

Example

```python
# Printing a statement n times.
n=int(input("Enter n: "))
for i in range(n):
    print("statement printed " + str(i) + "time(s).")
```

Functions

Definition

Functions are predefined lines of code that can be called n-times without re-writing code to avoid **code duplication**.

- A function can be defined using `def` keyword.
- A function can be be to operate on pre-existing data using arguemnts passed inside the curly braces `()` of a function.

Syntax

```
    #Defining a function
    def <functionName>(<argument1>,<argument2>,...):
        |
        |<statements>
        |
```

Example

```
#defining a function to add two numbers.

def add(num1,num2):
    """ Returns the sum of num1 and num2 """ #DocString
    sum = num1+num2
    return sum          # return is used to pass a value
                        # back to the normal flow of
                        # program.

a= float(input("enter num1: "))
b= float(input("enter num2: "))
c=add(a,b)          # add function returns sum to c.
print("sum : " + str(c))
```