Things To Remember

Precedence Of Operators

In Decreasing Order (Arithmetic>Relational>Logical)

( )
*** ***
+x –x ~x
/ // * %
+ -
<< >>
&
^
|
== < > <= >= **!=**
not
and
or

Built-in Functions

1. input()
   enables user to accept an input. Default class is string

2. eval()
   eval('10+15') >>25
   evaluates the value of a string
3. composition
   the value returned by a function may be used as an argument for another function in a nested manner
4. print()
   prints or produce the output
   ***/n for new line /t for a tab(escape sequence characters)**
   ***use another backslash before an escape sequence character to avoid it**
   ***or use R or r before the string to do the same**
5. type()

   To print the type of a variable
6. round()
   rounds a number up to a specific number of decimal places
   round(89.625,2) >>89.62
   round(89.635) >>90
   round(89.625,0) >>90.0
7. type conversion

8. min() and max()
   to find the minimum and maximum value out of several values
9. pow()
   pow(a,b) a^b

10. random number generation
    import random
    random.random() randomint(1,n)

11. Functions from **math** module

import math

ceil(x) → returns the smallest integer greater than or equal to x
floor(x) → returns the smallest integer less than or equal to x
fabs(x) → returns the absolute value of x
exp(x) → returns the value of expression e**x
log(x,b) → returns the log(x) to the base b(or else to the base e)
log10(x) → returns the log(x) to the base 10
pow(x,y)
sqrt(x) → returns the square root of x
cos(x)→returns the cosine of x radians
sin(x) →returns the sine of x radians
tan(x) →returns the tangent of x radians
acos(x) →returns the inverse cosine of x radians
asin(x) →returns the inverse sine of x radians
atan(x) →returns the inverse tangent of x radians
degrees(x)→returns the value in degree equivalent of input value of x
radians(x)→ returns the value in radian equivalent of input value of x

If we want to see the complete list of Built-in functions we can use the built-in function dir(__builtins__)

Functions

def function_name(commaSeparatedListOfParameters):
statements

→calling a function

if __main__==‘__main__’:
main()

→Every Python module has a built-in variable called **__name__** containing the name of the module. When the module itself is being run as script, this variable **__name__** is assigned the sting **‘__main__’** designating it to be a **__main__** module.

➔ If there is no return statement in a function, the function returns the value **None** to the caller function on the execution of the last statement of the function. The value being returned may be assigned to a variable.

➔ The variables and expressions whose values are passed to the called function are called **arguments.**

➔ Fruitful and Void Functions

A function that returns a value is often called **fruitful function.**

A function that does not return any value is often called **void function.**

➔ Function **help**

can also be used to provide description of the function defined by user.

➔ The function parameters may be assigned initial values also called **default parameter values**.

➔ Keyword Arguments

parameterName=value

➔ Importing User Defined Modules

import nameOfTheModule

We can access all the functions defined in it by using the following notation: **module_name.function_name**

import area

import sys print(area.areaRectangle(length,breadth))

➔ **assert** statement

Used for error checking. **assert *condition***

For example the user has to enter marks obtained and the maximum marks, the range should be between 0-100. Now to make sure the input is not negative or the marks obtained is greater than maximum marks, **assert** function is used:

assert maxMarks >= 0 and maxMarks <=100

assert marks >= 0 and marks <= maxMarks