Import modules in Python

1. Create a module with the name example.py with the following commands

def add(a, b):

"""This program adds two

numbers and return the result"""

result = a + b

return result

>>> import example

>>> example.add(3,4.5)

1. >>> import math

>>> print (“ the value of pi is “, math.pi)

1. Import with renaming

import math as m

print("The value of pi is", m.pi)

1. from math import \*

print("The value of pi is", pi)

1. Python module search path

* The current directory.
* PYTHONPATH (an environment variable with a list of directory).
* The installation-dependent default directory.

>>> import sys

>>> sys.path

['', 'C:\\Users\\srsenthi\\AppData\\Local\\Programs\\Python\\Python36\\python36.zip', 'C:\\Users\\srsenthi\\AppData\\Local\\Programs\\Python\\Python36\\DLLs', '

thi\\AppData\\Local\\Programs\\Python\\Python36\\lib', 'C:\\Users\\srsenthi\\AppData\\Local\\Programs\\Python\\Python36', 'C:\\Users\\srsenthi\\AppData\\Local\\

[\\Python36\\lib\\site-packages](file:///\\Python36\\lib\\site-packages)']

Reloading a module

Create a module with the name sample.py with a print statement

Print (“this is a new module”)

>>> import sample

This is a new module

Make a change to the sample.py module as

print(“this is a changed module”)

>>> import imp

>>>imp.reload(sample)

Built in function

dir() 🡪 gives the names defined in our namespace

from math import \*

print("The value of pi is", pi)