```
# -*- coding: utf-8 -*-
"""
Created on Tue Sep 29 17:51:17 2015
@author: hina
Reference: https://docs.python.org/3/tutorial/index.html
"""

print ()

print ("Power operator:   **")

print ("2**4 = ", 2**4)

print ("-3**2 = ", -3**2)

print ("3**-2 = ", 3**-2)

print ()

print ("Unary and bitwise operators:   -   ~ ")

print ("-2 = ", -2)

print ("~2 = ", ~2, "(bitwise inversion)")

print ()

print ("Binary operators:   +   -   *   /   %   //")

print ("The Floor operator // yields the quotient: 13.5//2 = ", 13.5//2)

print ("The modulo operator % yeilds the remainder: 13.5%2 = ", 13.5%2)

print ()

print ("Shifting operators:   >>   <<")

print ("100 >> 3 = ", 100 >> 3)

print ("    same as: 100 // (2**3) = ", 100 // (2**3))

print ("100 << 3 = ", 100 << 3)

print ("    same as: 100 * (2**3) = ", 100 * (2**3))

print ()

print ("Binary bitwise operators:   |   &   ^")

print ("3 | 0 = ", 3 | 0)

print ("3 ^ 3 = ", 3 ^ 3)

print ("3 & 0 = ", 3 & 0)

print ()
```

```
print ("Comparisons:   >   >=   <   <=   ==   in   not in")

print ("5 > 2?", 5 > 2)

print ("5 == 3?", 5 == 3)

print ("5 <= 4?", 5 <= 4)

print ("2 in (1, 2, 3)?", 2 in (1, 2, 3))

print ("5 not in (1, 2, 3)?", 5 not in (1, 2, 3))

print ()

print ("Boolean operations: AND   OR   NOT")

print ("5 > 2 AND 5 == 2?", (5 > 2) and (5 == 2))

print ("5 > 2 OR 5 == 2?", (5 > 2) or (5 == 2))

print ("5 NOT == 2?", not (5 == 2))

print ()

a, b = 0, 1
print ("You can do multiple assignment: ", a, b)

a, b = b, a+b
print ("RHS Expressions are evaluated before any of the assignments take place.\
 The RHS expressions are evaluated from left to right: ", a, b)

print ()

# the Python math library provides many standard math functions
# https://docs.python.org/3/library/math.html
# below are ones that are most frequently used

# import the math library
import math

var1 = 2
var2 = 3

# round(x): round to nearest integer
print (round(var1/var2))

# floor(x): the largest integer less than or equal to x
print (math.floor(var1/var2))

# ceil(x): the smallest integer greater than or equal to x
print (math.ceil(var1/var2))

# pow (x, y): x raised to the power y
print (pow(var1, var2))
```

```
# fabs (x): absolute value of x
print (abs(var1 - var2))

# sqrt (x): square root of x
print (math.sqrt(var1))

print ()

# test

var1 = 5
var2 = 2
var3 = 2

var4 = math.sqrt(pow(var1,var2))

var5 = pow(var1-var2,var3)

print (var4, var5)

print ()
```