

Python

Using Variables and Math

Types of Data

a) String

- can contain any text – letters of the alphabet, digits, punctuation marks, special characters
- usually found between “ “ (quotation marks)
- **For example:** name = input(“Enter in your name: “)

b) Integer

- Can contain only integer values (positive or negative whole numbers)
- To convert string data to integer data use the **int()** function

c) Real or Float

- Can contain only real values (positive or negative numbers that can contain decimal places)
- To convert string data to real data use the **float()** function

The **input()** function, by default, converts the input data to a string.

Use the **int()** function with the input() function to enter integer data into the program.

Example: width = int(input(“Enter the width: “))

Use the **float()** function with the input() function to enter real or float data into the program.

Example: width = float(input(“Enter the width: “))

Math Operators and Functions

Python follows the rules of BEDMAS when evaluating a mathematical expression.

Since input using the **input()** function results in the input being converted to a string, any numeric input should be converted to an integer or to a real value as required. To convert a value to its integer equivalent use the **int()** function. To convert a value to its real number equivalent use the **float()** function.

Common Math Operators

Operation	Result	Operation	Result
$x + y$	sum of x and y	$+x$	x unchanged
$x - y$	difference of x and y	$abs(x)$	absolute value or magnitude of x
$x * y$	product of x and y	$int(x)$	x converted to integer
x / y	quotient of x and y	$round(x, y)$	x will be rounded to y decimal places

Operation	Result	Operation	Result
$x // y$	floored (integer) quotient of x and y	<code>float(x)</code>	x converted to a float or real
$x \% y$	remainder of x / y	<code>pow(x, y)</code>	x to the power y
$-x$	x negated	$x ** y$	x to the power y

Understanding the difference between `/` and `//` and also the `%` operator.

To better understand these operators lets use the example of $11/4$. $11/4$ is 2.75 or a quotient of 2 with a remainder of 3.

`print(11 / 4)` will display 2.75

`print(11 // 4)` will display 2 as 4 goes into 11 twice completely aka a floored integer division

`print(11 % 4)` will display 3 as 3 is the remainder when 11 is divided by 4

Example:

```
#introduction
print("Number of Planets")
print("-----")
print("") #an empty print statement is a good way to add a blank line

#asking for user input
rockyPlanets = float(input("How many inner rocky planets are there? ")) #4
gasGiants = float(input("How many outer gas giants are there? ")) #4

#calculations
total = rockyPlanets + gasGiants

#displaying the output for the user
print("")
print("In total there are", total, "planets in our solar system")
print("There are", rockyPlanets, "inner planets that are smaller and made
from rock. These are known as terrestrial planets")
print("There are", gasGiants, "outer planets that are large balls of gas.")
```

Proper organization in your code, much like it is displayed above with spacing and comments, is crucial for debugging (fixing errors) and collaborating with others. Also note the input function will be evaluated first and then the float function as python works from the inside out on those lines asking users for input.

The Math Library – One of the key advantages of python as a coding language is the large number of libraries containing pre-made functions we can use so we don't need to re-invent the wheel. One such library is the math library. The math library contains many important math functions that can be used in a python program. The math library is made available to the python program by using the **import** statement. **import math**

It is not necessary for the simple math operations outlined in the table above

Some of the commonly used math functions are:

math.pow(x, y) - Return x raised to the power y.

math.sqrt(x) - Return the square root of x.

math.hypot(x, y) - Return the Euclidean norm, $\sqrt{x^2 + y^2}$. This is the length of the vector from the origin to point (x, y).

math.pi - The mathematical constant $\pi = 3.141592\dots$, to available precision.

Example:

```
import math

#Introduction
print("Characteristics of Earth")
print("-----")
print("")

#asking for user input
radius = float(input("Enter the radius of Earth in km: ")) #answer: 6371

#performing calculations
surfaceArea = 4 * math.pi * radius ** 2
mass = 5 * 10**24

#displaying the characteristics nicely for the user
surfaceArea = round(surfaceArea,2)
print("")
print("Earth has a surface area of:",surfaceArea,"km squared and a mass
of",mass,"kg")
```

If **math.pi** is the only thing we want to use from the math library we can also write: **from math import pi**. If we do this, we can simply write **pi** instead of **math.pi** when we use it in a calculation.

****Remember we must use **int()** around our input to convert the input into an integer since input on it's own will make your input a string****

Ex. 1: Write a program that asks the user how many moons Jupiter (79) and Saturn (82) have and calculates the difference between the two. Final output. **Saturn has 82 moons which is 3 more than Jupiter's 79 moons.**

Ex 2.: Prompt the user for the length in days of 1 Mars year (687) and 1 Venus year (225) in Earth days. Calculate the ratio between them. Final output: **Mars's year is 3.05 times longer than Venus's.**

Ex. 3: Prompt the user for the radius of the Earth (6371) and the Sun (695700) in km and calculate both volumes. After doing this find out how many entire Earths can fit into the sun and display the final answer.

Note: volume of a sphere = $\frac{4}{3} * \pi * r ** 3$

Final Output: **You can fit 1,302,096 Earths into the Sun. That's a lot of Earths.**