A+ Computer Science

PYTHON BASICS

A Simple Program

```
print("Comp Sci Rocks!")
```

Output

```
Comp Sci Rocks!
```

Python Output

To output a line use print

```
print("Comp Sci")
print("Rocks!")
```

Output

```
Comp Sci
Rocks!
```

Escape Sequences

To print certain characters or add lines you will need to add \

```
print("Comp Sci \\Rocks!")
print("Comp Sci\tRocks!")
```

Output

```
Comp Sci \Rocks!
Comp Sci    Rocks!
```

Escape Sequences

frequently used combinations

Name	Use
\t	tabs over five spaces
\n	moves to front of next line
\r	moves to front of current line
\\	displays one backslash \ when printed
\"	displays one double quote " when printed
\’	displays one single quote ’ when printed

output.py

Python Comments

```
# Single line comment

"""
This is a multi line comment
"""
    # this line prints stuff on the screen
    print("stuff")
```

Python Comments

```
# Single line comment

"""
This is a multi line comment
"""

    """
    This prints out stuff
    """
    print("stuff")
```

Variables

Variables allow us to store values.

```
grade = 86
student = "Bob"
theEnd = False
```

Data Types

Used for variables

Type	What it stores
boolean	True or False
integer	A whole number (ex. 50)
float	A decimal number (ex. 5.02)
string	A series of characters – this can be letters, words, or numbers (ex. “60 seconds)
list	A series of variables (ex. [23, "Hi", False])
tuple	An unchangeable list (ex. (23, "Hi", False))

Identifiers

This is the name you give your variables. Identifiers can contain letters and numbers, but should start with a letter.

```
grade = 86
student = "Bob"
theEnd = False
```

Identifiers

Use descriptive identifiers that mean something

Bad

```
supercalifragilisticexpialidocious  = 86
thatOneGuy = "Bob"
asdfghjkl = False
```

Good

```
grade = 86
student = "Bob"
theEnd = False
```

Spelling

Spelling matters

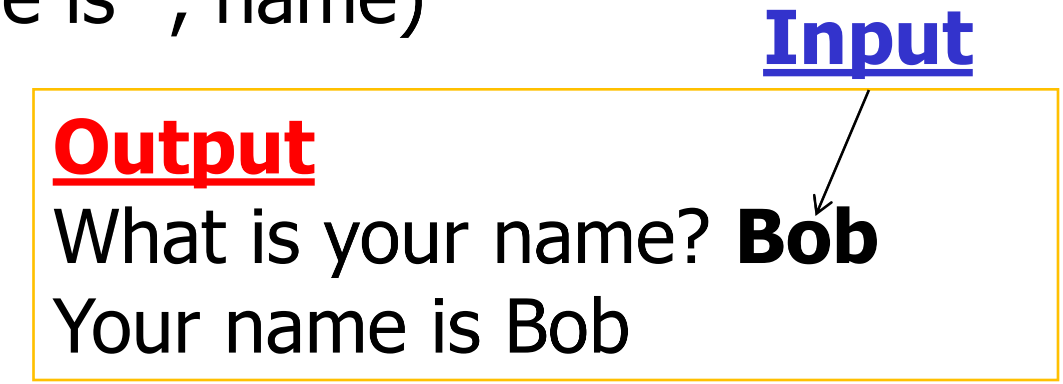
Name is not the same as **name**
Name is not the same as **mane**

© A+ Computer Science - www.apluscompsci.com

Input

Sometimes we want the user to give us information. We can store it in a variable.

```
name = input("What is your name? ")
print("Your name is ", name)
```



Input

Output

```
What is your name? Bob
Your name is Bob
```

Input with Numbers

Numbers in Strings need to cast to an int or float

```
numString = "56"
print(4 + numString)
```

Output

```
ERROR
```

Casting a String to a Number

```
numString = "56"
num = int(numString)
print(4 + num)
```

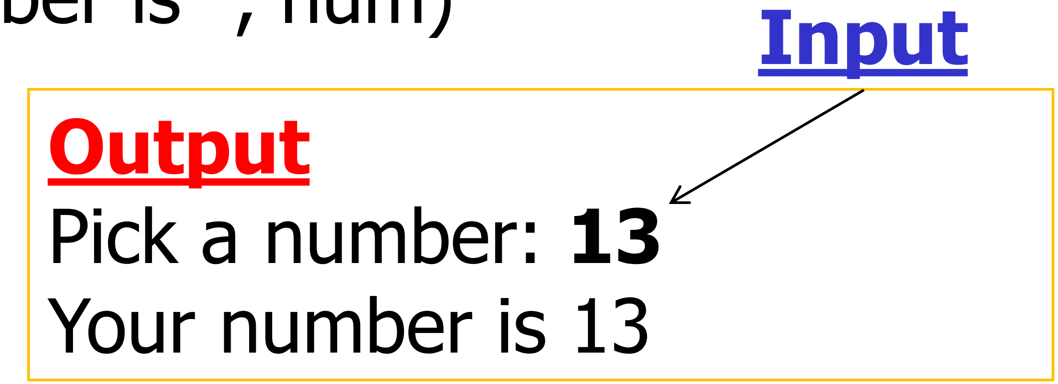
Output

```
60
```

Input with Numbers

Numbers from input need to cast to an int or float.

```
num = int(input("Pick a number: "))
print("Your number is ", num)
```



Input

Output

```
Pick a number: 13
Your number is 13
```

Output with Variables

Commas allow you to print multiple things on the same line

```
name = Bob
print("Name is ", name)

num = 54
print("Num is ", num)
```

Output

```
Name is Bob
Num is 54
```

variables.py

Math Operators

To do math expressions, you use math operators

```
total = one + two + three
product = four * five
```

Math Operators

+	Addition
-	Subtraction
*	Multiplication
/	Division
//	Floor Division
%	Modulus (getting the remainder)
**	Exponential (ex. 5**2 is 25)

Operator Precedence	
(expression)	Parenthesis – things inside parenthesis happen first
**	Exponential
-x	Negation (-5**2 is -25)
*, /, %	Multiplication, Division, Modulus
+, -	Addition, Subtraction

© A+ Computer Science - www.apluscompsci.com

Integer Math vs. Real Math

If there is a decimal number, the result is a decimal. If all numbers are integers, the result is an integer.

```
print("Total is", str(3 + 4))
print("Product is", str(1.5 * 12))
```

Output

```
Total is 7
Product is 18.0
```

Integer Math vs. Real Math

```
intDiv = 3 // 4
print  ("Integer division is", intDiv)
```

Output

```
Integer division is 0
```

Integer Math vs. Real Math

```
decDiv = 3 / 4.0
print  ( "Decimal division is", decDiv )
decDiv = 3 / 4
print  ( "Decimal division is", decDiv )
```

Output

```
Decimal division is 0.75
Decimal division is 0.75
```

Shortcut Operators

num = num + 1 can also be written num += 1. All of the math operators can be used like this.

```
num = 3
num = num * 2
print (num)
num *= 2
print (num )
```

Output

```
6
6
```

math.py

Work on Programs!

Crank

Some Code!

A+ Computer Science

PYTHON BASICS