**VALUES AND TYPES**

>>> print(4)

4

>>> type('Hello')

<class 'str'>

**Integer:**

>>> type(17)

<class 'int'>

**Float:**

>>> type(3.2)

<class 'float'>

**String or Character:**

>>> type('17')

<class 'str'>

**VARIABLES:**

A variable is a name that refers to a value. An assignment statement creates new variables and gives them values:

a=12

b=5.239

c=’Hello’

To display the value of a variable, you can use a print statement:

>>> print(a)

12

>>> print(b)

5.239

**VARIABLE NAMES:**

* Can use alphabets, underscore and numbers
* Can only start with alphabets or underscore
* Keywords cannot be used
* Cannot use special characters

a12=23, c2=’Python’

**KEYWORDS:**

Special words which have a pre-defined meaning in Python

Words are - and del from None True as elif global nonlocal try assert else if not while break except import or with class False in pass yield continue finally is raise async def for lambda return await

**STATEMENTS :**

A statement is a unit of code that the Python interpreter can execute.

print(5)

x = 3

print(x)

produces the output

5

3

**OPERATORS AND OPERANDS :**

Operators are special symbols that represent computations like addition and multiplication. The values the operator is applied to are called operands. The operators +, -, \*, /, and \*\* perform addition, subtraction, multiplication, division, and exponentiation,

c=12+70

>>> minute = 59

>>> minute/60

0.9833333333333333

**EXPRESSIONS:**

An expression is a combination of values, variables, and operators

x=5

c= x + 17

or

>>> 1 + 1

2

**ORDER OF OPERATIONS:**

1. Parenthesis
2. Exponents
3. Multiplication, Division, Remainder & Quotiont
4. Addition and Subtraction

* Operators with the same precedence are evaluated from left to right.

**MODULUS OPERATOR:**

The modulus operator works on integers and yields the remainder when the first

operand is divided by the second. In Python, the modulus operator is a percent

sign (%). The syntax is the same as for other operators:

>>> quotient = 7 // 3

>>> print(quotient)

2

>>> remainder = 7 % 3

>>> print(remainder)

1

**STRING OPERATIONS:**

The + operator works with strings, but it is not addition in the mathematical sense. Instead it performs concatenation, which means joining the strings by linking them end to end. For example:

>>> first = 10

>>> second = 15

>>> print(first+second)

25

and

>>> first = '100'

>>> second = '150'

>>> print(first + second)

100150

The \* operator also works with strings by multiplying the content of a string by an integer.

For example:

>>> first = 'Test '

>>> second = 3

>>> print(first \* second)

Test Test Test

**ASKING THE USER FOR INPUT:**

>>> inp = input()

Charles

>>> print(inp)

Charles

or

>> name = input('What is your name?\n')

What is your name?

Chuck >>>

print(name)

Chuck

**COMMENTS:**

It is a good idea to add notes to your programs to explain in natural language what the program is doing. These notes are called comments, and in Python they start with the # symbol.

Example: percentage = (minute \* 100) / 60 # percentage of an hour