Informatics Practices

Python Fundamentals

By

Rajesh Verma

INTRODUCTION

Python is a high-level, interpreted and general purpose dynamic programming language that focuses on code readability. The syntax in Python helps the programmers to do coding in fewer steps as compared to Java or C++.

Let us learn the basic elements of python programming

PYTHON CHARACTERSET

Character set is a bunch of identifying elements in the programming language.

- *Letters:- A-Z, a-z*
- *Digits:- 0 to 9*
- *Special Symbols:- space + - / ( ) [ ] = ! = < > , ‘ “ $ # ; : ? &*
- *White Spaces:- Blank Space , Horizontal Tab, Vertical tab, Carriage Return.*
- *Other Characters:- Python can process all 256 ASCII and Unicode Characters*

Tokens Or Lexical Unit

What is Token?

Individual elements that are identified by programming language are called tokens or lexical unit.

TOKENS / LEXICAL UNITS

Keywords

Identifiers

Literals

Operators

Punctuators

1. Keyword/Reserved Word

What is Keyword?

Keywords are also called as reserved words these are having special meaning in python language. The words are defined in the python interpreter hence these cant be used as programming identifiers.

Keywords of Python Language

and	exec	not
as	finally	or
assert	for	pass
break	from	print
class	global	raise
continue	if	return
def	import	try
del	in	while
elif	is	with
else	lambda	yield
except		

2. Identifiers

What is an identifier?

A Python Identifier is a name given to a function, class, variable, module, or other objects that you'll be using in your Python program.

In short, its a name appeared in the program.

For example: a, b, c a b and c are the identifiers and a b & c and , are the tokens

Python Naming Conventions

- An identifier starts with a letter A to Z or a to z or an underscore (_) followed by zero or more letters, underscores and digits (0 to 9).
- Python does not allow special characters
- Identifier must not be a keyword of Python.
- Python is a case sensitive programming language.
- Thus, Rollnumber and rollnumber are two different identifiers in Python

VALID IDENTIFIERS:

Myfile1	DATE9_7_8
y3m9d3	_xs
MYFILE	_FXd

INVALID IDENTIFIERS:

MY-REC	28dre	break
elif	false	del

3. Literals / Constant Values

What is literals?

Literals are also called as constants or constant values these are the values which never change during the execution of program.

Or

Literals in Python can be defined as number, text, or other data that represent values to be stored in variables.

Types Of Literals / Constant Values

1) String Literals or Constants.

2) Numeric Literals or Constants.

3) Boolean Literals or Constants.

4) Special Literal None.

1. String Literals Or Constants

What is string?

Sequence of letters enclosed in quotes is called string or string literal or constant.

Python supports both form of quotes i.e.

'Hello'

"Hello"

Types Of Strings

Python supports two ways of representation of strings:

1) Single Line Strings.

2) Multi Line Strings.

Single Line

Strings created using single quote or double quote must end in one line are called single line strings

For Example:

Item="Computer"

Or

Item= 'Computer'

Multi Line Strings

Strings created using single quote or double quote and spread across multiple lines are called Multi Line Strings.

By adding backslash \ one can continue to type on next line.

For instance: Item = 'Key

board'

STRINGS WITH TRIPLE QUOTES

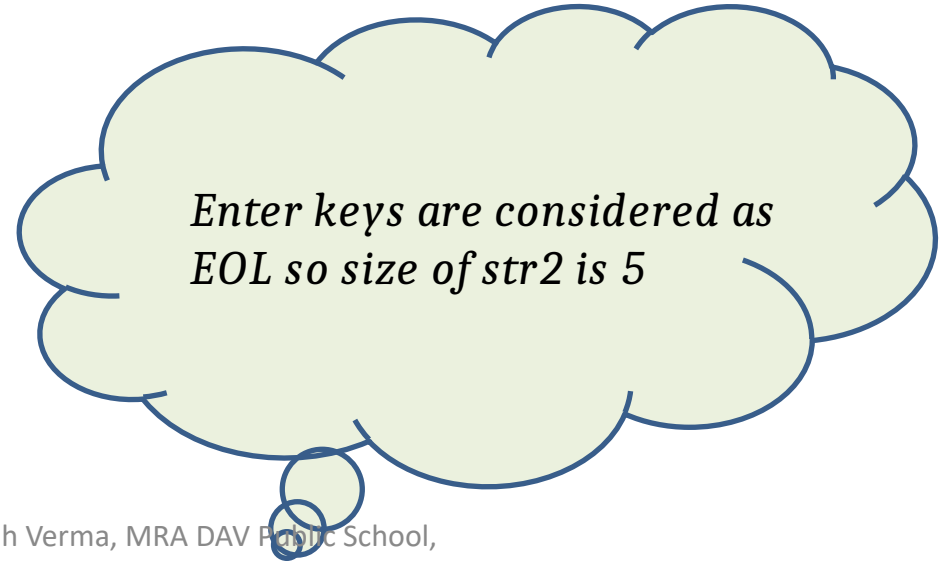
For multi line strings created by triple quotes, while calculating size, the EOL(End of Line) character at the end of line is also counted.

For instance:

Str2="'x ↵

y ↵

Z"'



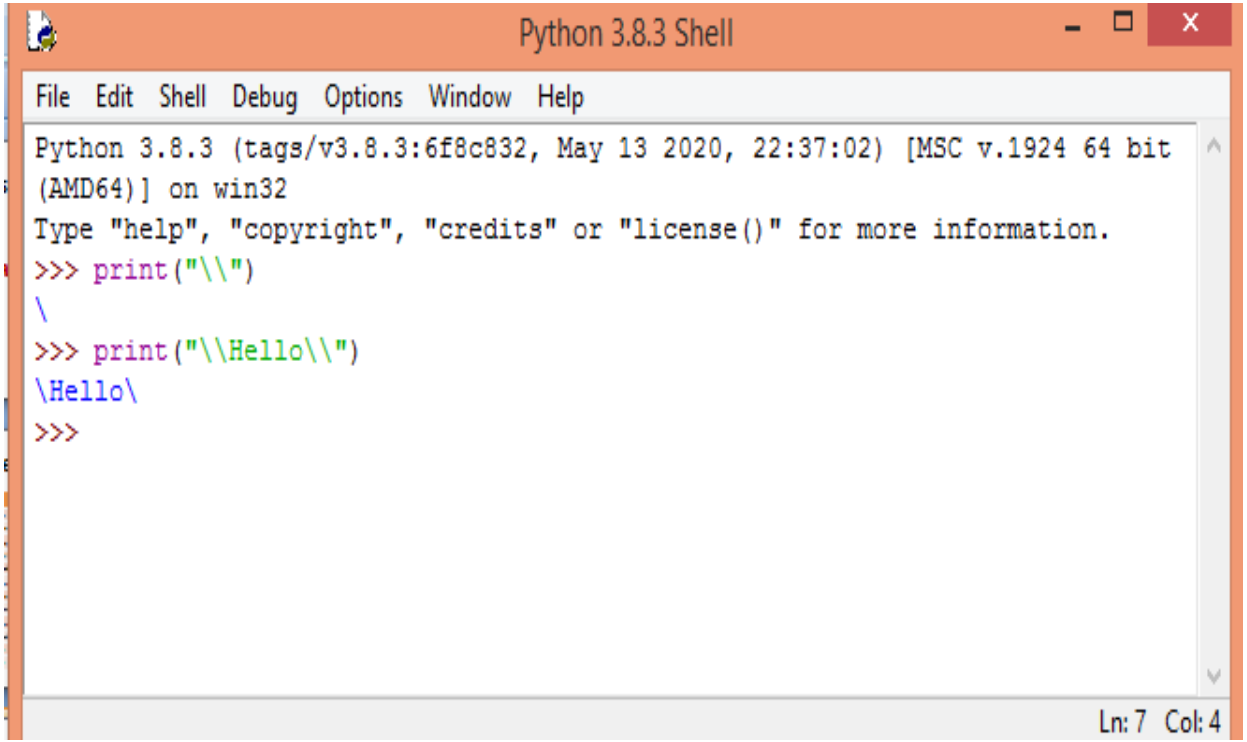
Escape Sequences

- Python allows you to have certain non-graphic character in string values.
- Non-graphic character are those characters that cannot be typed directly from the keyboard e.g backsapce, tabs, carriage return.
- These non-graphic characters can be represented by escape sequence. An escape sequence is represented by a backslash(\) followed by on or more characters.

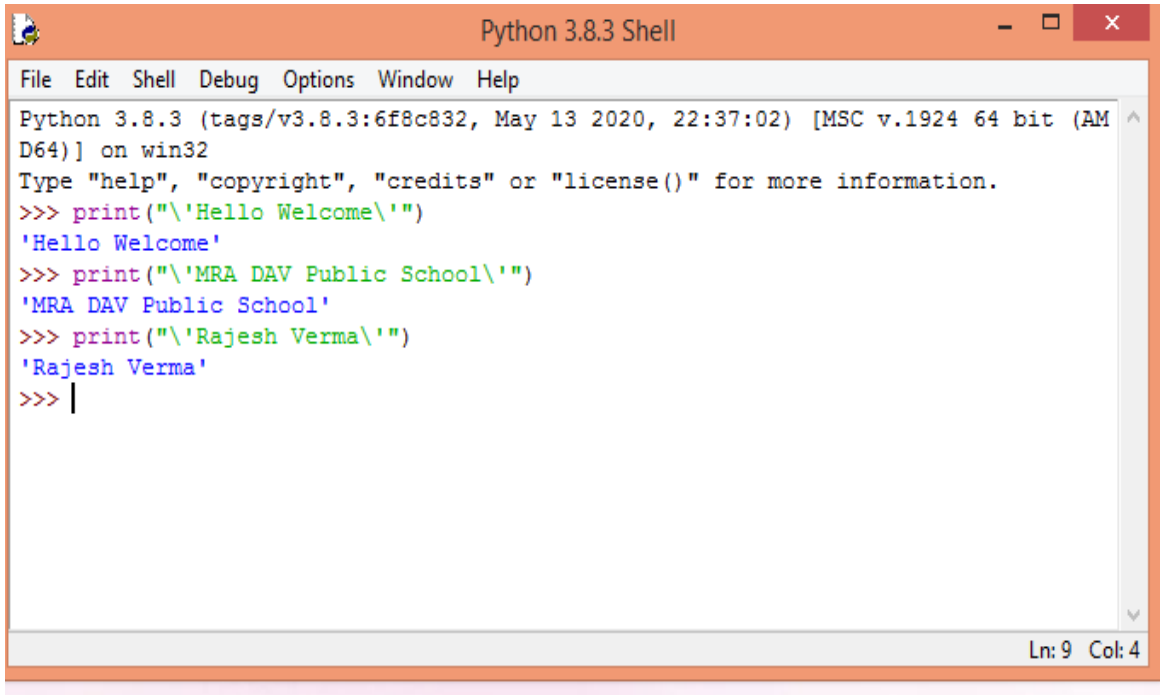
Escape Sequence

Escape Sequence	Description
\\	Backslash (\)
\'	Single quote (')
\"	Double quote (")
\a	ASCII Bell (BEL)
\b	ASCII Backspace (BS)
\f	ASCII Formfeed (FF)
\n	ASCII Linefeed (LF)
\r	ASCII Carriage Return (CR)
\t	ASCII Horizontal Tab (TAB)
\v	ASCII Vertical Tab (VT)
\ooo	Character with octal value ooo
\xhh	Character with hex value hh

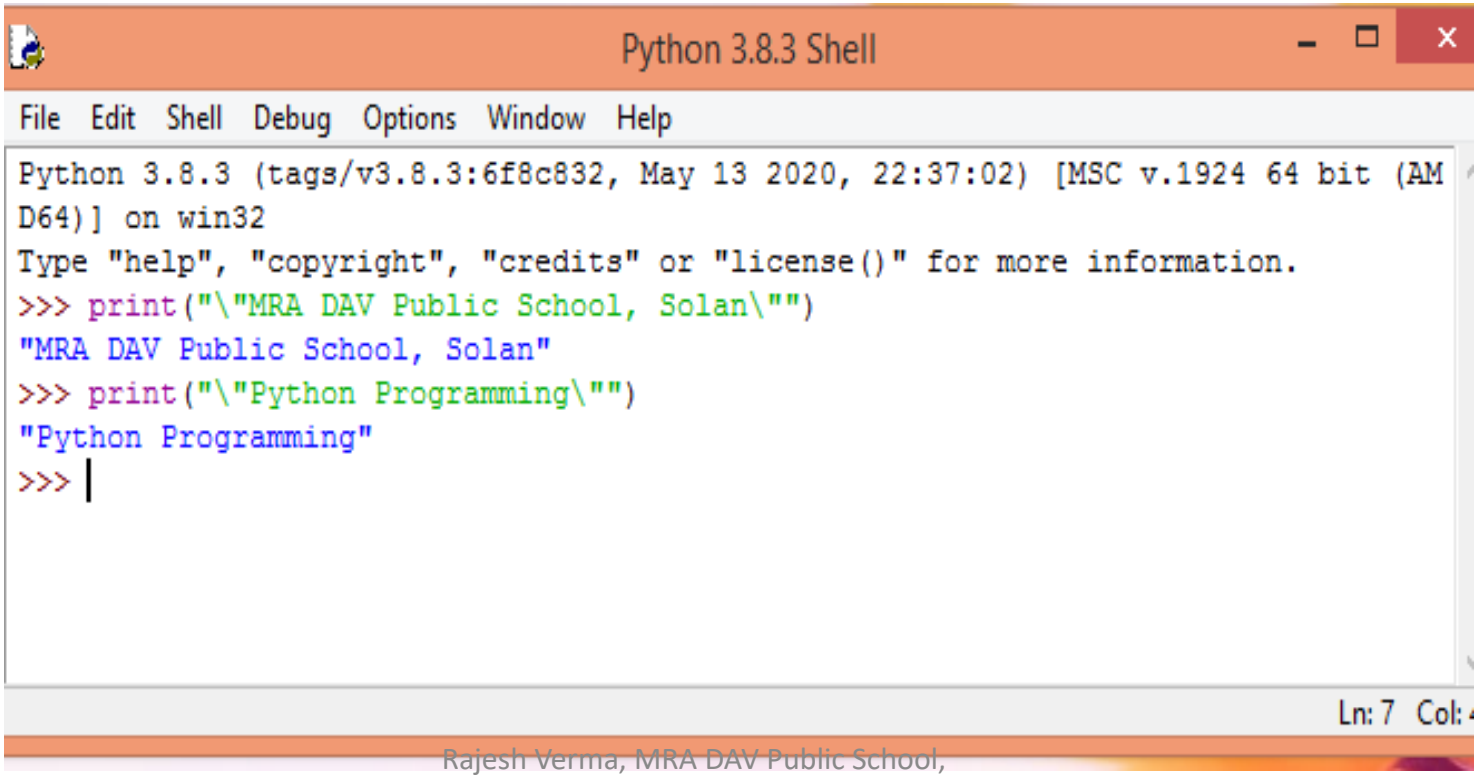
\\ Back Slash


Python 3.8.3 Shell
File Edit Shell Debug Options Window Help
Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22:37:02) [MSC v.1924 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> print("\\")
\
>>> print("\\Hello\\")
\Hello\
>>>
Ln: 7 Col: 4

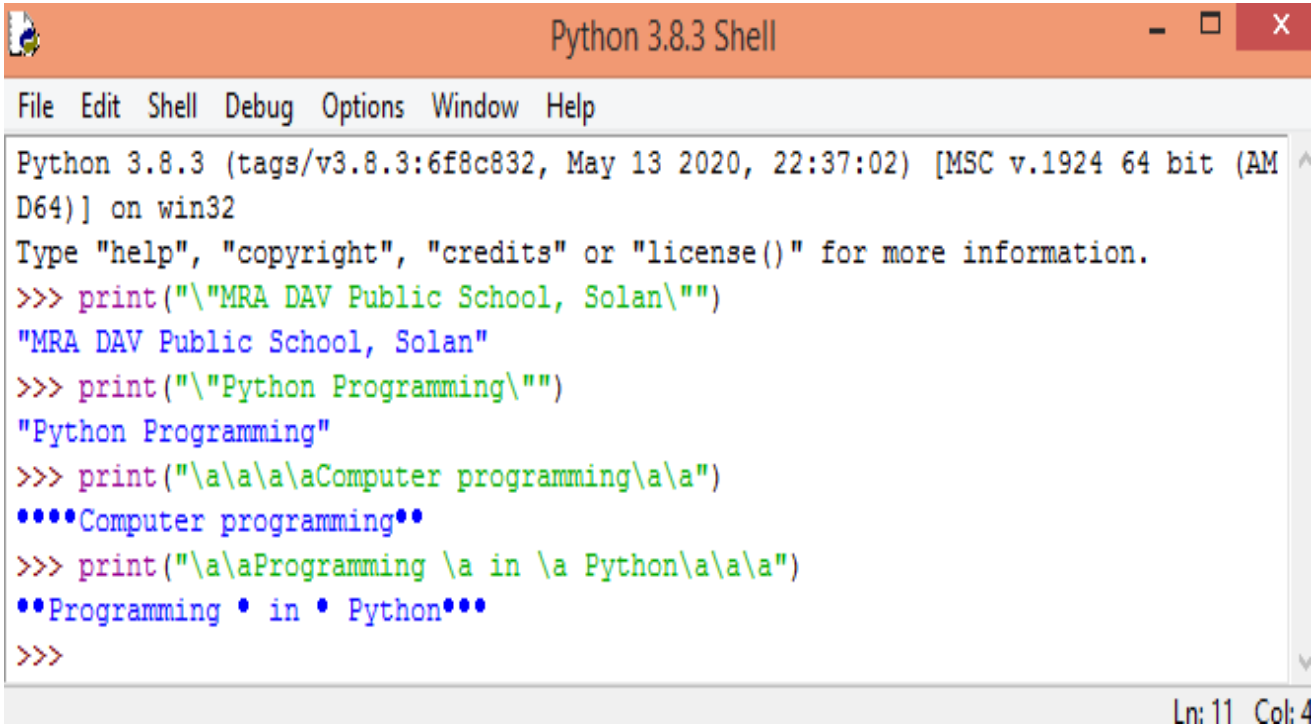
\' Single Quote



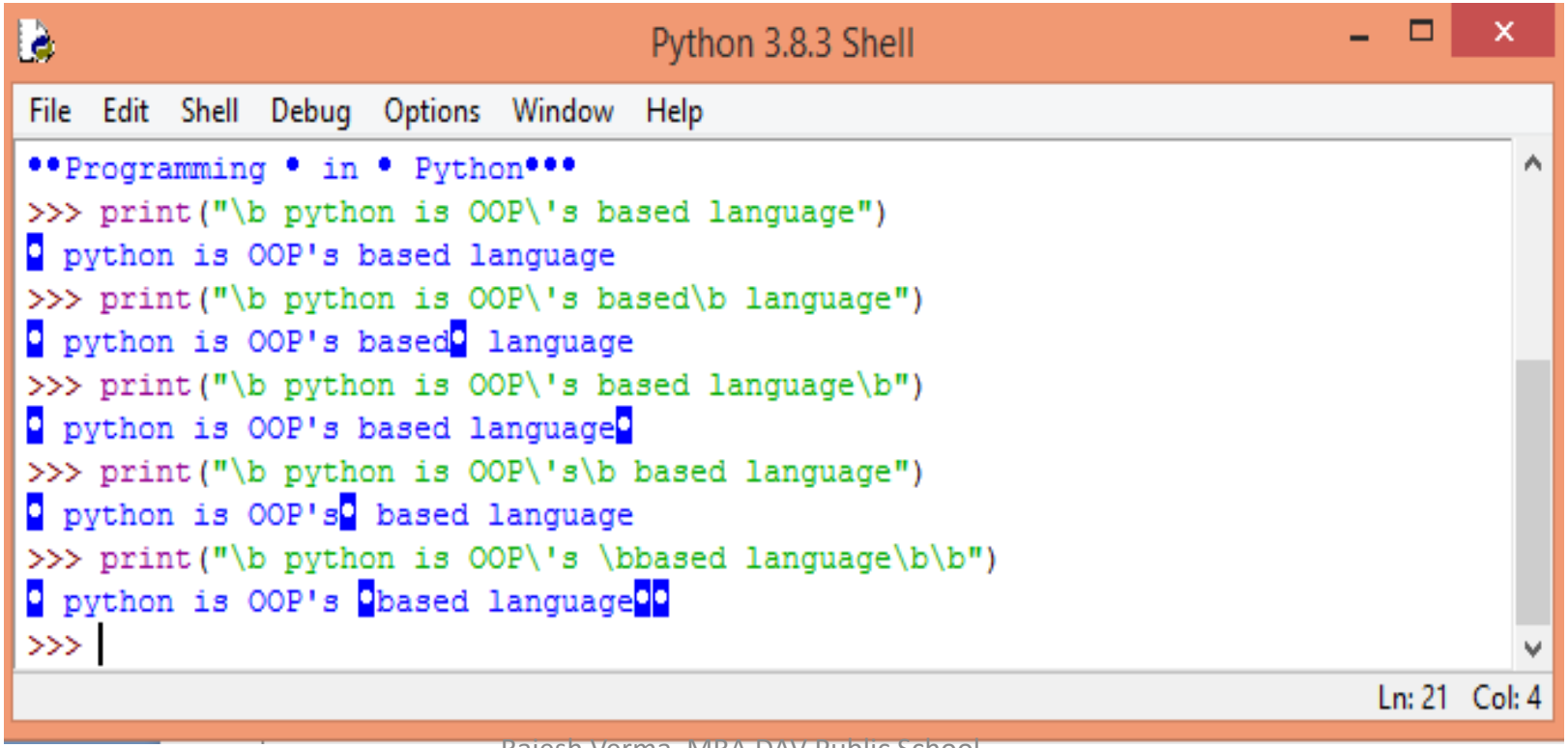
\” Double Quote



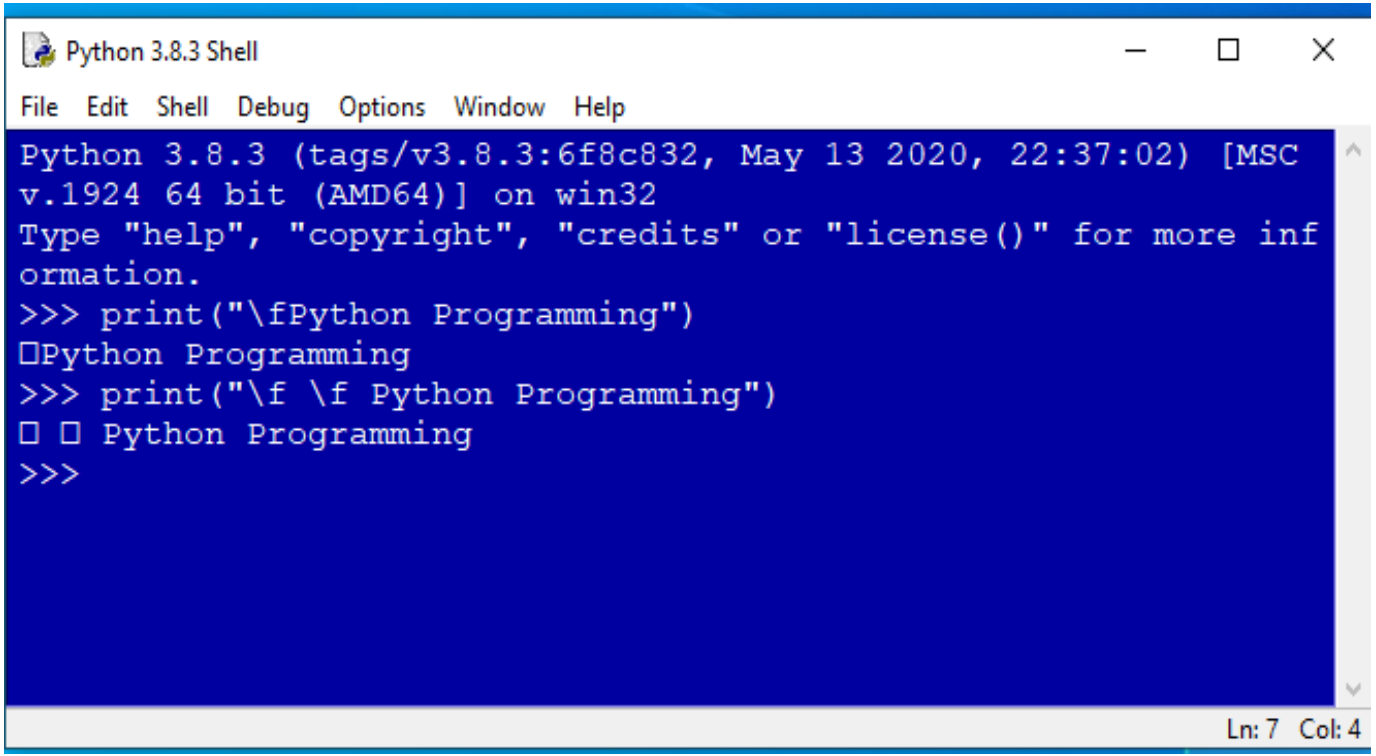
\a ASCII Bell


Python 3.8.3 Shell
File Edit Shell Debug Options Window Help
Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22:37:02) [MSC v.1924 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> print("\"MRA DAV Public School, Solan\"")
"MRA DAV Public School, Solan"
>>> print("\"Python Programming\"")
"Python Programming"
>>> print("\a\a\a\aComputer programming\a\a")
••••Computer programming••
>>> print("\a\aProgramming \a in \a Python\a\a\a")
••Programming • in • Python•••
>>>
Ln: 11 Col: 4

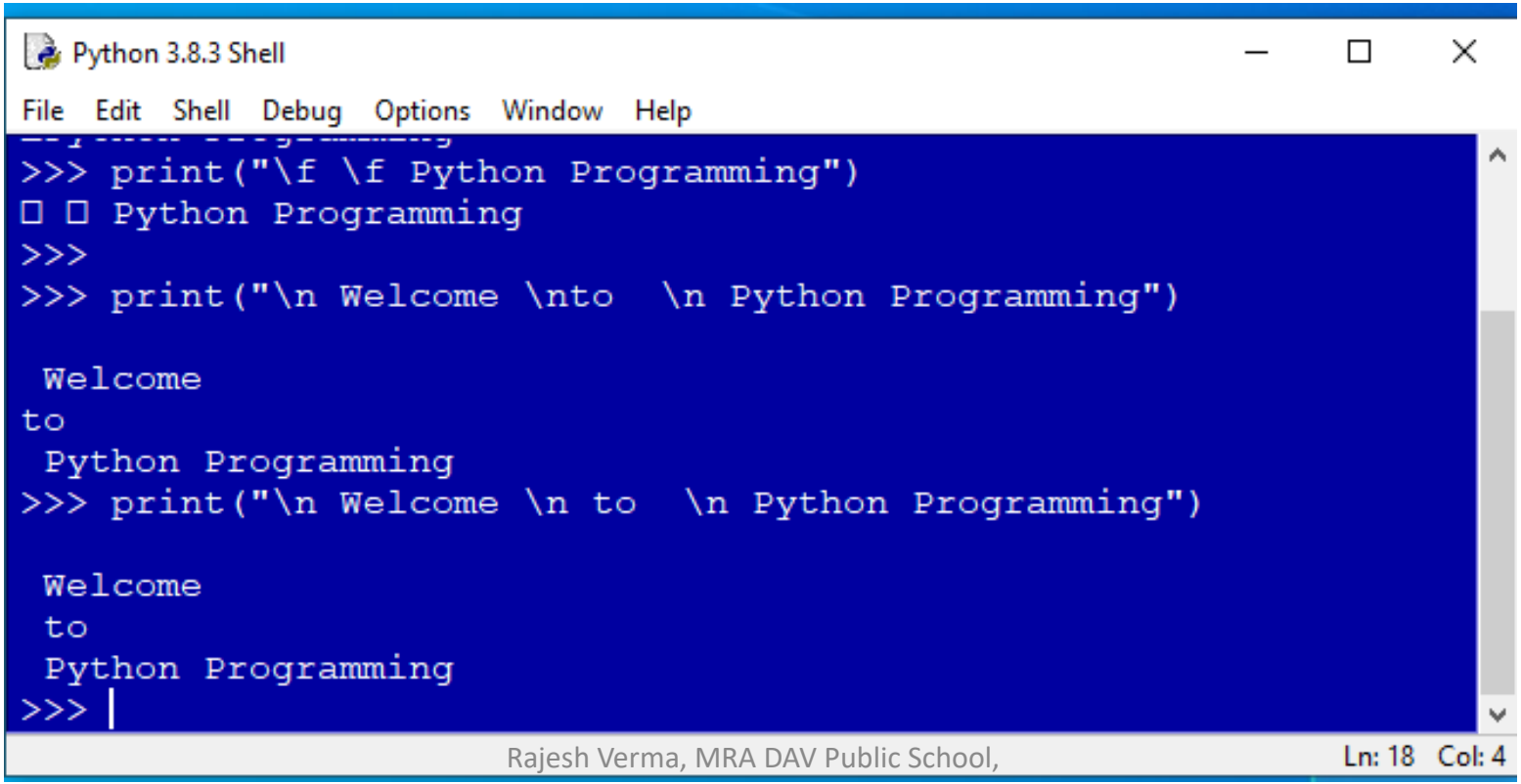
\b ASCII Backspace



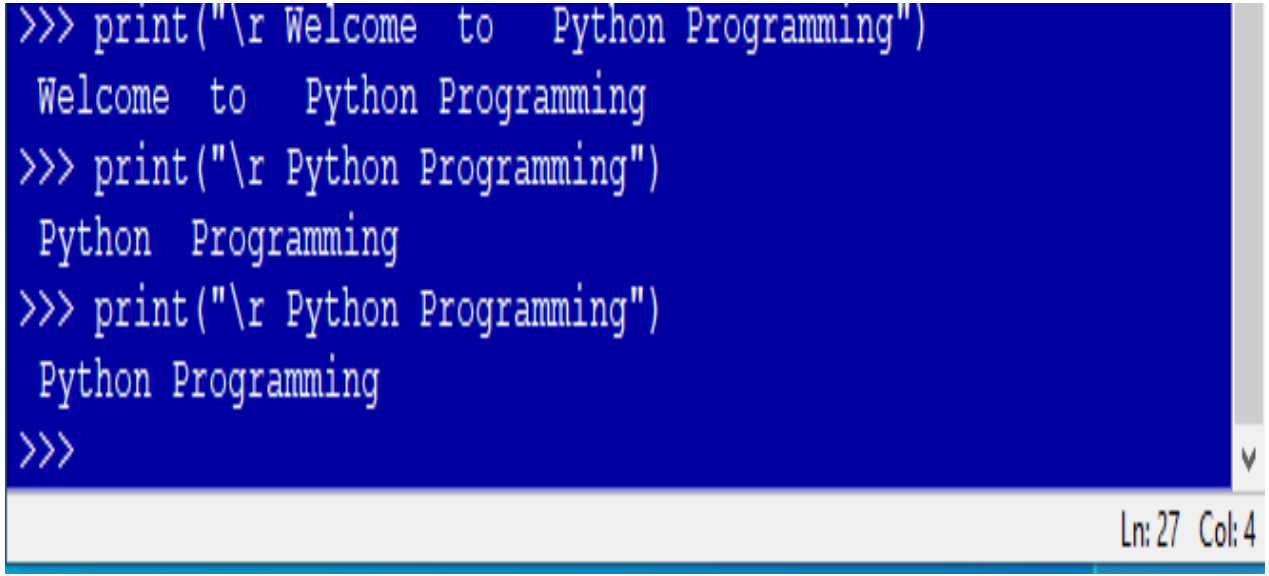
\f ASCII Form Feed



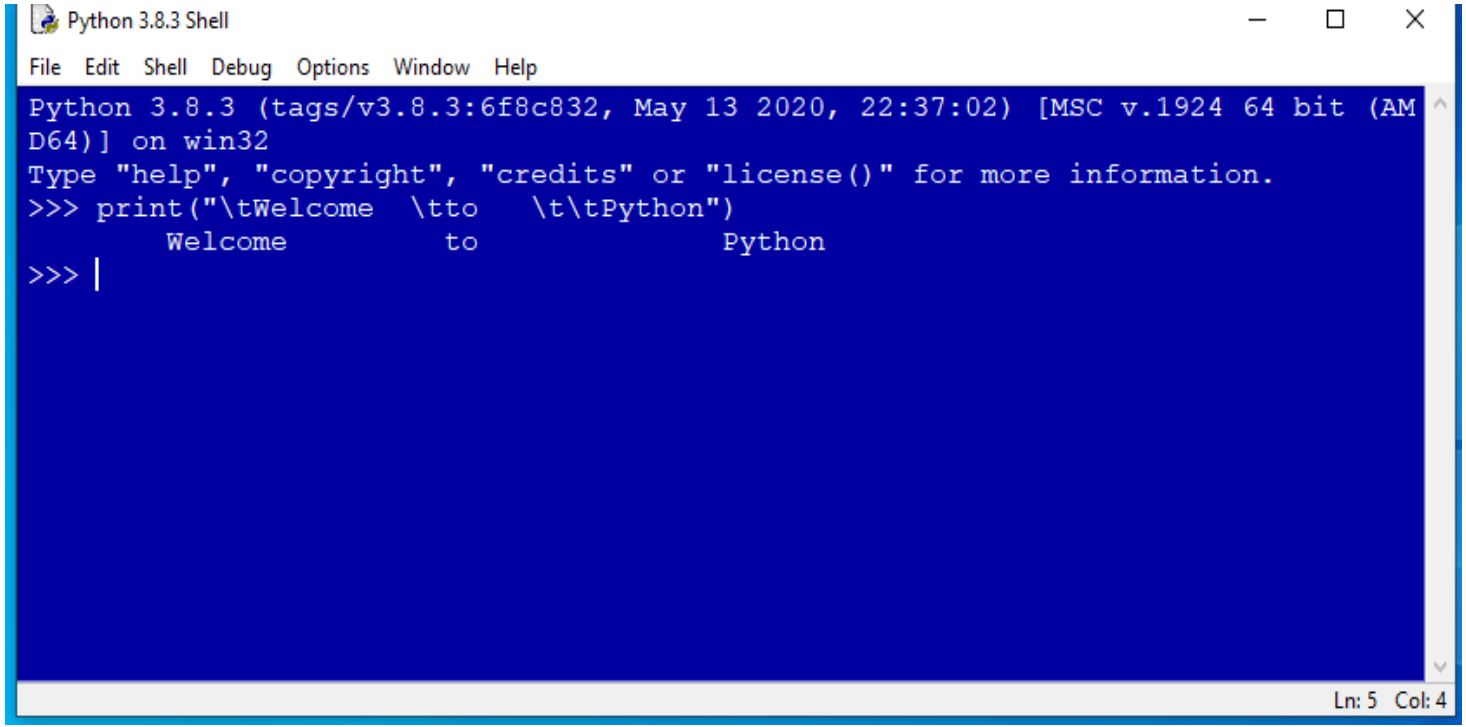
\n New Line


Python 3.8.3 Shell
File Edit Shell Debug Options Window Help
>>> print("\f \f Python Programming")
□ □ Python Programming
>>>
>>> print("\n Welcome \nto \n Python Programming")
Welcome
to
Python Programming
>>> print("\n Welcome \n to \n Python Programming")
Welcome
to
Python Programming
>>>
Ln: 18 Col: 4

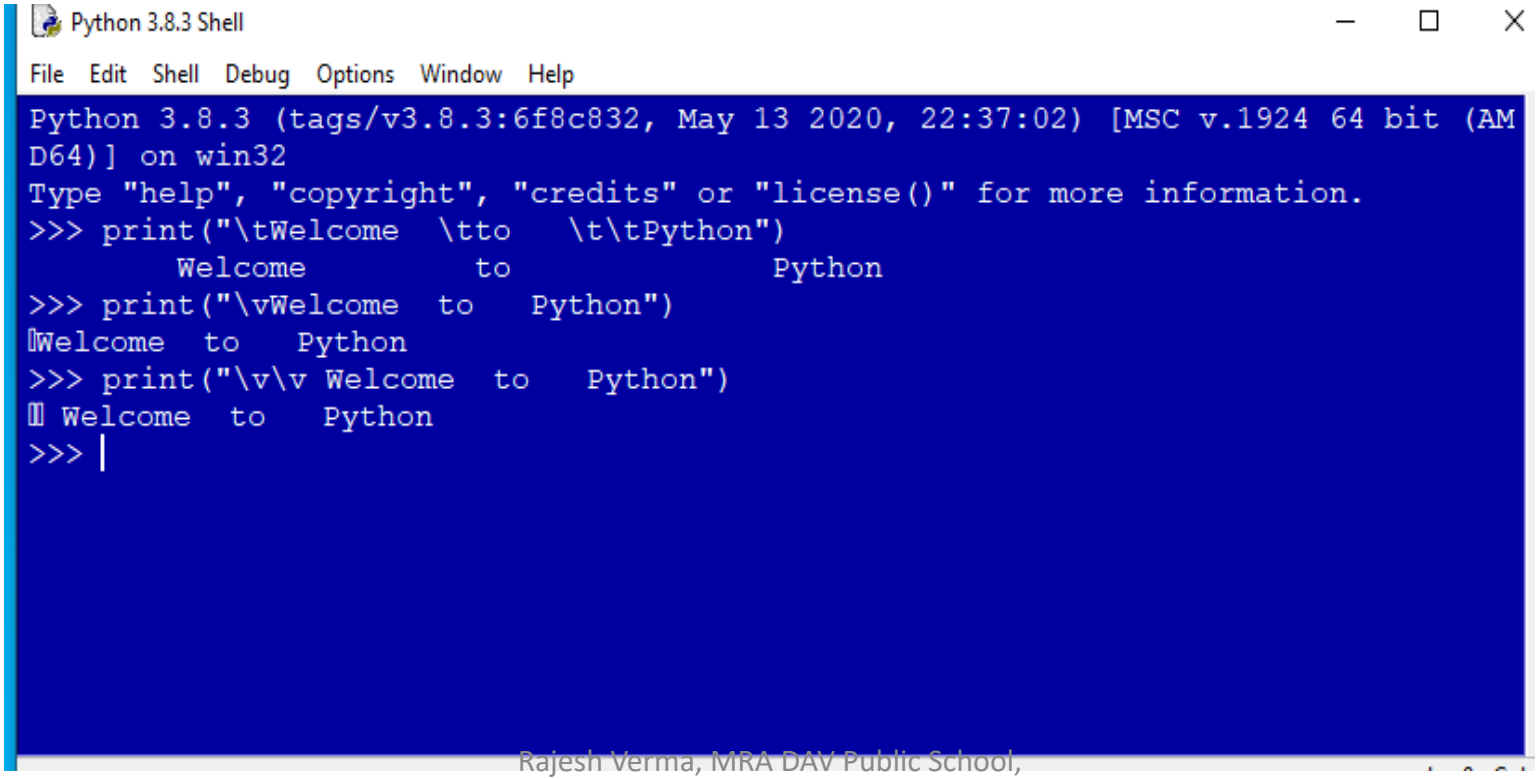
\r Carriage Return



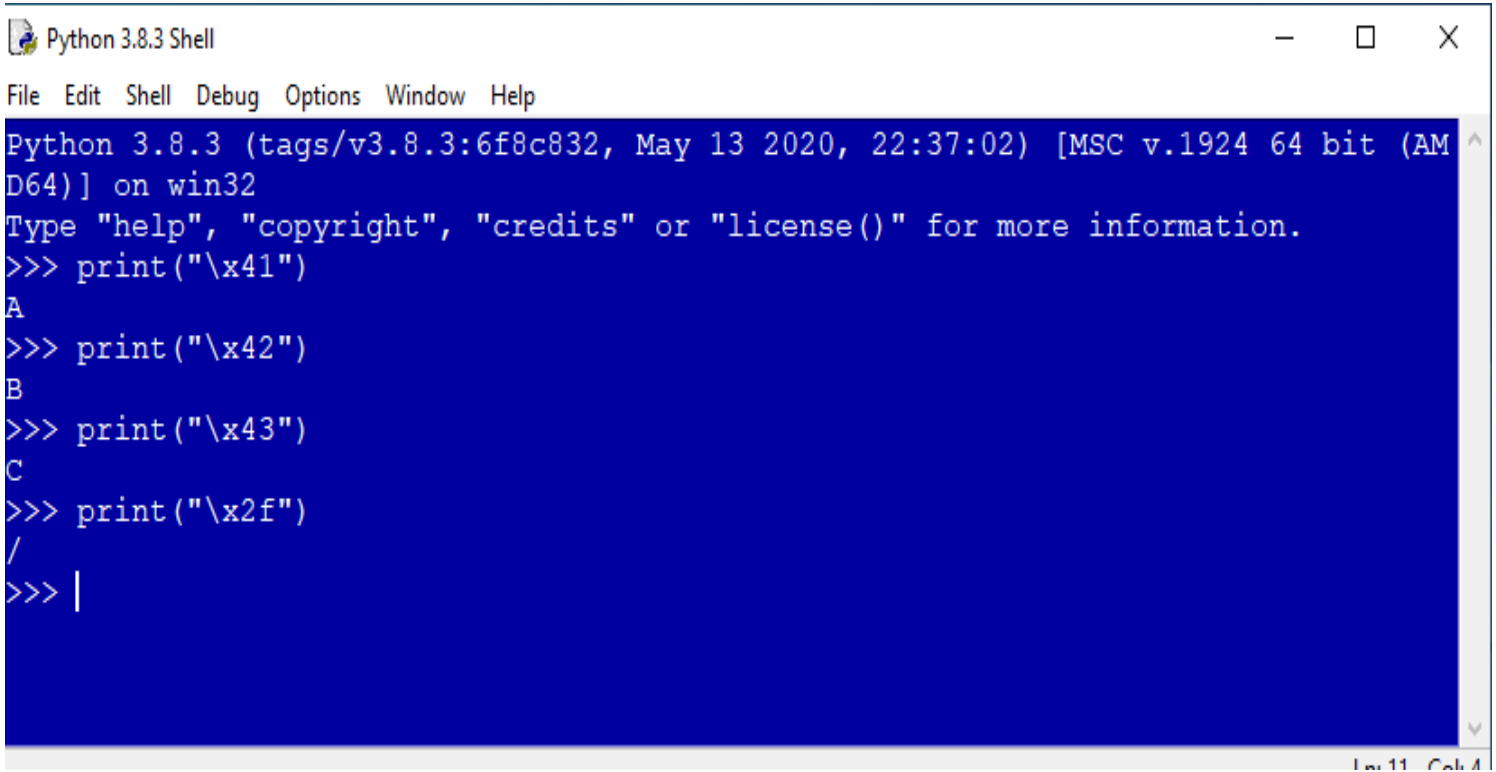
\t Horizontal Tab



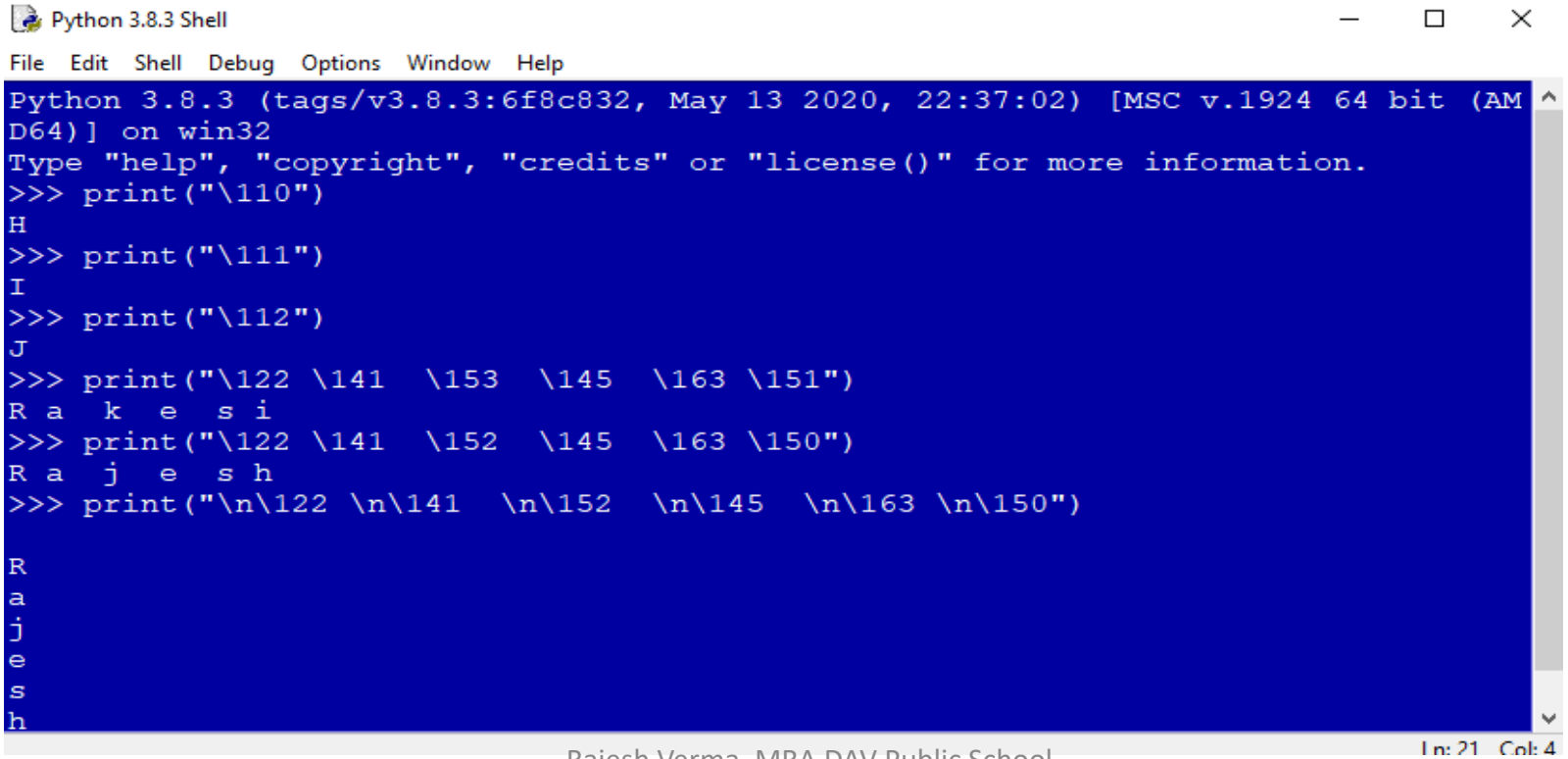
\v Vertical Tab



\x 16 bit Hex Val



\ooo Octal Value



2. NUMERICAL LITERALS

Numerical Literals have the following types:

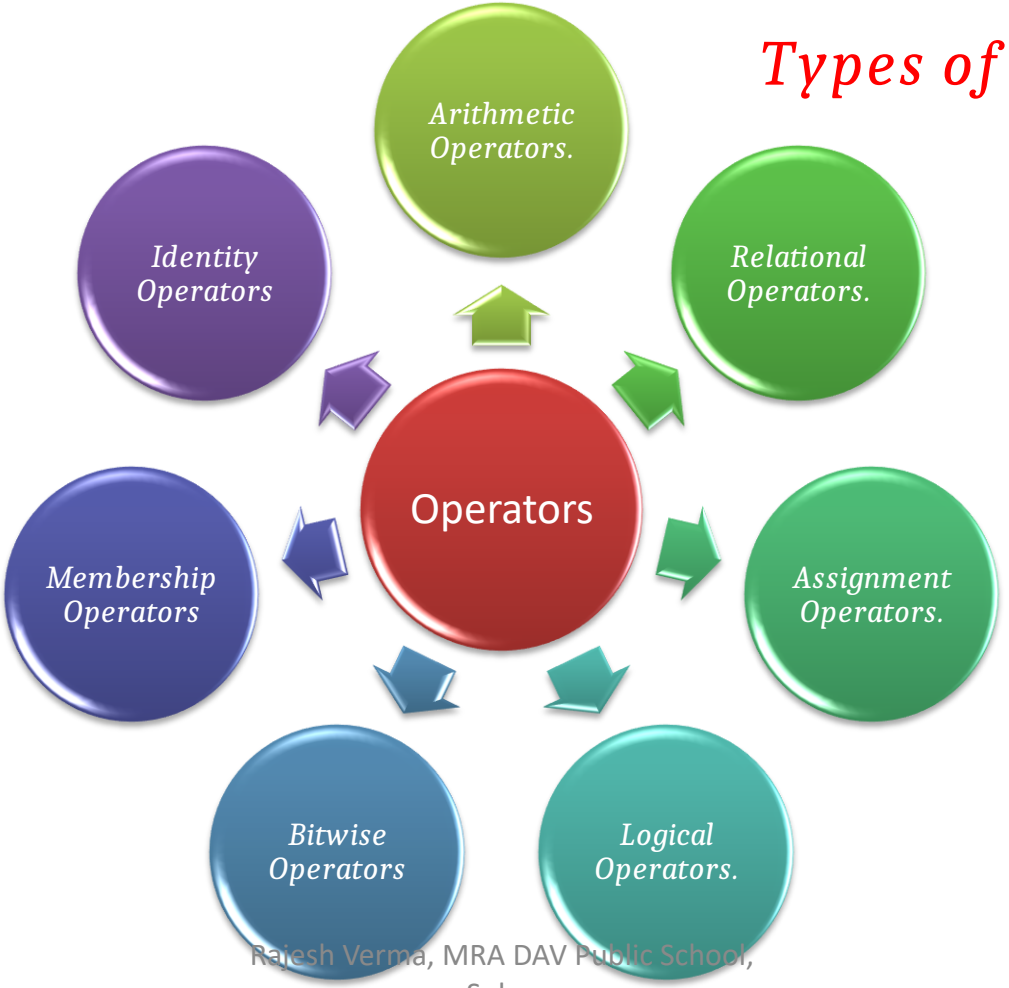
1. *int or integers - Whole numbers*
2. *float - real values*
3. *Complex - Complex numbers*

OPERATORS

What is an operator?

Operators are tokens that trigger some computation when applied to a variable.

Types of Operators



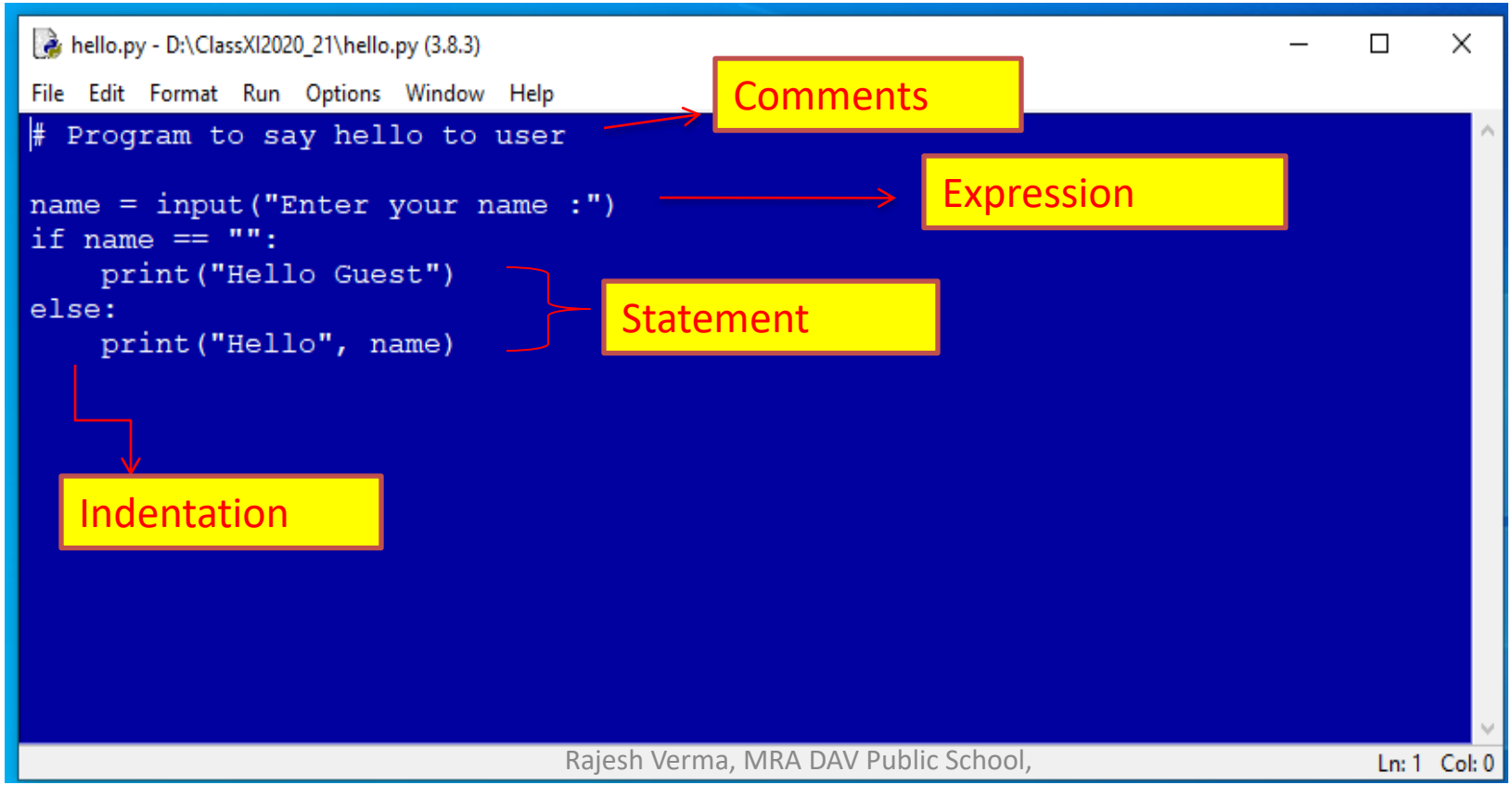
PUNCTUATORS

Punctuators are also called as separators

The followings are used as punctuators:

- *Brackets* [ ]
- *Parentheses* ( )
- *Braces* { }
- *Comma* ,
- *Semicolon* ;
- *Colon* :
- *Asterisk* *
- *Ellipsis* ...
- *Equal Sign* =
- *Hash Sign* #

Barebone of a python program



A python program contain the following components

a. **Expression :** - which is evaluated and produce result. E.g. (20 + 4) / 4

b. **Statement :-** instruction that does something.

- e.g a = 20
- print("Calling in proper sequence")

c. **Comments :** which is readable for programmer but ignored by python interpreter

i. Single line comment: Which begins with # sign.

ii. Multi line comment (docstring): either write multiple line beginning with # sign or use triple quoted multiple line. E.g.

- '''this is my
  first
  python multiline comment '''

d. **Function**

- a code that has some name and it can be reused. e.g. keyArgFunc in above program

e. **Block & indentation :** group of statements is block indentation at same

Variables

Variable is a name given to a memory location. A variable can consider as a container which holds value. Python is a type infer language that means you don't need to specify the data type of variable. Python automatically get variable data type depending upon the value assigned to the variable.

Assigning Values To Variable

```
name = 'python' # String Data Type

sum = None   # a variable without value
```

VARIABLES AND ASSIGNMENTS

Named labels are called variables.

For example: marks =86

78	79	80	81	82	83	84	85	86	87
2000	2016	2018	2026	2032	2044	2048	2050	2054	2068

VARIABLES AND ASSIGNMENTS

Multiple Assignments

Python is very versatile with assignment statements.

1. Assigning same value to multiple variables:

```
a=b=c=d=e=10
```

2. *Assigning Multiple values to multiple variables:*

```
p, q, r =5,10,15
print(q, r) will print 10 15
p, q=q, p
print (p, q) will print 10 5
```

VARIABLES AND ASSIGNMENTS

type() function:

To know the data type of a value which is pointing use type ( )

```
>>>a=10
>>>type(a)
<class 'int'>
>>>a=20.4
>>>type(a)
<class 'float'>
```

Type returned as integer

Type returned as float

Dynamic typing

Data type of a variable depend/change upon the value assigned to a variable on each next statement.

```
X = 25              # integer type
X = "python"        # x variable data type change to string on just next line
```

Now programmer should be aware that not to write like this:

```
Y = X / 5  # error !! String cannot be divided
```

Input and Output

print() *Function In Python is used to print output on the screen.*

Syntax of print Function

print(expression/variable)

e.g. print(122) Output

:122

print('hello India')

Output :-hello India

print('Computer', 'Science')

print('Computer', 'Science',sep='&')

print('Computer', 'Science', sep=' & ', end='.')

Output :-Computer Science

Computer & Science

Computer & Science.

Input ()

variable = input(< Prompt to display>)

e.g. name= input('What is your name:')

The input () function always returns a value of string type

If you enter integer value it will be treated as string .

age= int(input('What is your age:'))

type(age) =>> int

int ( ) and float ( ) Functions:

Python offers two functions to be used with input( ) to convert the received values:

Example 1: >>age = int(input("Enter age"))

Example 2: >>sal=float(input("Enter salary))

Any Questions Please

Rajesh Verma, MRA DAV Public School,
Solan