

Assignment-1.

Date : / / 20

Title :- Write python code for word count & find occurrence of word from file.

Problem statement :- To python code for word count & find occurrence of word from file.

Theory :- What is python? Python is a powerful modern programming language. It bears some similarities to Fortran, one of the earliest programming languages, but it is much more powerful than Fortran.

By learning Python you will also be learning a major tool used by many web developers.

Python Commands -

Comments in a Python command, anything after a # symbol is a comment.
for ex -

print "Hello world" # this is silly.
Comments are not part of command, but rather intended as documentation for anyone reading the code.

" """ Multiline comments are enclosed by triple double-quote symbols.

Numbers & other data types-

python recognize several different data types. for instance, 23 & -75 are integers while 5.0 & -23.09 are floats or floating point numbers.

The number 12345678901 is a long integer.

The type function

To see the type of some data, use python builtin type function:

```

type(-75)
<type 'int'>
type(5.0)
<type 'float'>
type(12345678)
<type 'long'>
>>> 2j
2j
>>> 2j-1
(-1 + 2j)
>>> complex(2,3)
9
(2+3j)
>>> type(-1+2j)
<type 'complex'>
```

2 Strings -

other useful data types are strings

```
>>> "This is a string"
```

This is a string

```
>>> type("This is a string")
```

<type 'str'>

Strings are an example of sequence type

3 Lists & tuples -

```
>>> [1, 3, 4, 1, 6]
```

[1, 3, 4, 1, 6]

```
>>> type([1, 3, 4, 1, 6])
```

<type 'list'>

```
>>> (1, 2, 3)
```

```
>>> type((1, 2, 3))
```

<type 'tuple'>

4) The range function -

```
>>> range(1, 10)
```

[1, 2, 3, 4, 5, 6, 7, 8, 9]

```
>>> range(-6, 0)
```

[-6, -5, -4, -3, -2, -1]

```
>>> range(1, 10, 2)
```

[1, 3, 5, 7, 9]

```
>>> range(10, 0, -2)
```

[10, 8, 6, 4, 2]

Note the use of negative increment in last example.

5) Boolean Values

```
>>> True
```

True

```
>>> type(True)
```

<type 'bool'>

```
>>> type(False)
```

<type 'bool'>

Boolean types are used in making decisions.

• Expressions -

An expression is anything which produces a value. Ex - $2+2$, 2^{100} , $f((x-1)(x+1))$

Expressions are formed from variables, constants, function evaluations & operators,

Operators -

The common binary operators for arithmetic are + for addition, - for subtraction

* for multiplication & // for division.

** for exponentiation.

```
>>> 25/3
```

8

```
>>> 5/2
```

2

```
>>> 25.0/3
```

8.33333

```
>>> 5/2.0
```

2.5

Variables & assignment

```
>>> x = 2 + 2
```

```
>>> print x
```

4

In the above examples the assignment statement sets x to 4, producing no output. If we want to see the values of x, we must print it. If we execute another assignment to x then the previous value is lost.

```
>>> x = 380.5
```

```
>>> print x 380.5
```

```
>>> y = 2 * x
```

```
>>> print y 761.0.
```

A single = is used for assignment, double == is used to test for equality.

```
>>> x = 10
```

```
>>> x = x + 1
```

```
>>> print x
```

11

```
>>> x = x + 1
```

```
>>> print x
```

12

Decisions -

~~or~~ if-else

if condition:

action-1

else:

action-2.

elif:

if $x \geq 0$ and $x < 10$:

digits = 1.

elif $x \geq 10$ & $x < 100$

digits = 2.

elif $x \geq 100$ & $x < 1000$

digits = 3.

else:

digits = 0 # more than 4.

- Loops

- 1) for loop

- 2) While loop.

for item in list:

action.

for i in [2, 4, 6, 0]

print i.

This produces output.

2 4 6 0.

while loop:

while condition.

action

Lists

In python lists are not required to be homogeneous i.e the items could be of different types.

ex -

```
a = [2, "Jack", 45, "23 Ave"]
```

```
>>> a.
```

```
[2, "Jack", 45, '23 Ave']
```

```
>>> a[0]
```

```
2
```

```
>>> a[2]
```

```
45.
```

Length of list ; empty list

```
>>> x = [9, 4, 900, 45]
```

```
>>> len(x)
```

```
4.
```

```
>>> x = []
```

```
>>> len(x)
```

```
0
```

Sublists (slicing)

sublist obtained by slicing

start < i < end .

ex -

```
x = range(0, 20, 2)
```

```
>>> x
```

```
[0, 2, 4, 6, 8, 10, 12, 14, 16, 18]
```

```
>>> x[2:5]
[4, 6, 8]
>>> x[:5]
[0, 2, 4, 6, 8]
>>> x[2:]
[4, 6, 8, 10, 12, 14, 16, 18]
```

```
>>> list[18::-1]
[17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0]
```

In general the slice `x[len(x):-1]`
reverses any existing list `x`.

joining two lists -
using + operator-

```
>>> [2,3] + [4,0]
[2, 3, 4, 0].
```

Lists methods.

`x.append(item)`

ex-

```
>>> x = [3, 6, 8, 9]
>>> x.append(999)
>>> x
[3, 6, 8, 9, 999]
```

A similar method is called `insert`

```
>>> x = ['a', 'c', '3', 'd', 'f']
>>> x.insert(0, 100)
>>> x
```

[100, 'a', 'c', '3', 'd', 7]

delete.

>>> x.remove('a')

>>> x

[100, 'c', '3', 'd', 7]

To delete item at index position i use

x.pop(i)

>>> x.pop(0)

100

>>> x

['c', '3', 'd', 7]

x.pop()

'7'

>>> x

['c', '3', 'd']

Strings -

A string in python is a sequence of characters.

>>> x = 'gobblety gook'

>>> x[2]

'b'

>>> x[5]

'e'

>>> x[5] = 's'

Type error

because unlike lists strings are immutable
If you need to change an existing string

you must make a new, changed one.

Conclusion :- Thus the python 2 programming tool is installed & exercised basic syntax, data types, variables, operators vectors, lists, matrices, Data frames, various types of graphs taking the suitable ex from & count no of word & vowels from file.