

THEORY QUESTION ASSIGNMENT

1. Python theory questions

1. Python is a free/open source programming language

The main features of Python are :

- Being an Object Oriented Language which means a programming which works around the concept of an object
- Being an open source language
- Being an High level Language : a programming language designed to be more user friendly programming
- Having extensible feature : we can code some python code into C or C++
- Being a portable language : python can be used on licensed platforms like windows, mac but as well open source platforms like Linux
- Being an interpreted language : Python code has a line by line execution, You can even add some steps to begin to run the code exactly where you want
- Being dynamically typed language : you don't need to specify the type of variables in advance

2. Python is a open source language which evolves through different versions

Python 2 and Python 3 are versions of Python, Python 2 is more a legacy version of Python even if it is still widely used while Python 3 is the recent more used version especially for beginners .

Differences between Python 2 and 3 :

- Syntax ; there is more parentheses used with Python 3 which changes for example print 'hello ' statement for Python 2 to a print('hello') function in Python 3
- Default text strings : is ASCII in Python 2 but UTF-8 in Python 3
- Libraries : libraries are not forwards compatible using Python 2 but they are using Python 3

References

W3SCHOOLS: <https://www.w3schools.com/python/>

GeeksforGeeks: <https://www.geeksforgeeks.org/>

Programiz : <https://www.programiz.com/python-programming>

Baeldung : <https://www.baeldung.com/cs/>

Dataflair : <https://data-flair.training/blogs/>

Analytics Vidhya : <https://www.analyticsvidhya.com/blog/>

- Number operations accuracy : Python 3 give you the exact number result for a division hwile Python 2 would round down to the nearest whole number,
- 3. PEP stands for Python Enhancement proposal, PEP is a document providing the best practices and guidelines to code in Python. PEP 8 is focus on the improvements of the readability and consistency of Python code, PEP 8 describes new features , processes and/or environement of Python,
- 4. A program in computer science is a set of instructions that the computer will interprete and perform,
- 5. A process is a program in execution
- 6. Cache is hardware or software which is used to store data in computing environment
- 7. A thread is a unit/sequence of execution within a process, Multithreading is a way process can contain several simultaneous or parallell threads,
- 8. Parralelism is a process which splits into threads/tasks into subtasks/subthreads which can be executed in parallel while concurrency is threads/tasks which can be executed at the same time
- 9. GIL stands for Global Interepreter Lock which is a type of process lock, It prevents to have multi threads execution. Indeed, Only one thread wil be executed at a time in Python
- 10.DRY principle stands for don't repeat yourself;the dry principle is a principle to reduce the repetition and duplication in your code. KISS principle stands for Keep It Simple Stupid ;the KISS principle is a principle to make sure to keep the code as simple as possible. BDUF principle stands for Big Design Up Front ; the BDUF idea is to have the program to be completed and perfected before the program implementation,
- 11.Garbage collector in Python is realeasing memory when the objects is not used anymore
- 12. At the run time, computer memory is split in different memory parts:code, stack and heap, Code is part of the memory which stores the program :the set of instructions for the computer to interprete and execute. During the program execution , the computer will interprete functions and local variables

References

W3SCHOOLS:<https://www.w3schools.com/python/>

GeeksforGeeks:<https://www.geeksforgeeks.org/>

Programiz :<https://www.programiz.com/python-programming>

Baeldung :<https://www.baeldung.com/cs/>

Dataflair :<https://data-flair.training/blogs/>

Analytics Vidhya :<https://www.analyticsvidhya.com/blog/>

temporarily and statically in Stack memory. The objects are stored in Heap memory as Heap memory is a dynamic memory allocation. Indeed, we have to be able to change the size of memory and the objects, This is why we can't store objects in Stack memory. Fortunately, Python interpreter is allocating and deallocating actively and automatically on the heap memory,

13. Modules are code that someone else has written that you can reuse in your program,

14. Docstrings are strings within a function, method, class. They are represented with closing and opening quotes,

15. Pickling is the process by which Python objects are converted to byte stream. Unpickling is the reverse operation. Pickling and unpickling are used to help to transfer Python objects from one machine to another.

16. For static analysis, you can use Pychecker or/and Pylint tools, For debugging, if using a 'clever' IDE e.g. PyCharm, we can use in-built debugging functionality to run a program in debugging mode.

17. Arguments in Python are passed by reference. If you change parameter refers in the function, it will reflect back in the function

```
price_fruits={'orange':2,'banana':1,'grapes':4,'strawberry':5}
```

```
def test(student):
```

```
    new={'clementine':3,'apple':4}
```

```
    price_fruits.update(new)
```

```
    print("Inside the function",price_fruits)
```

```
    return
```

```
test(price_fruits)
```

```
print("outside the function:",price_fruits)
```

18. Dictionaries store a collection of labelled items where each item has a key and a value. The following is a dictionary

```
ample_dict = {
```

```
    "name": "Lisa",
```

```
    "age":25,
```

```
    "salary": 50000,
```

References

W3SCHOOLS:<https://www.w3schools.com/python/>

GeeksforGeeks:<https://www.geeksforgeeks.org/>

Programiz :<https://www.programiz.com/python-programming>

Baeldung :<https://www.baeldung.com/cs/>

Dataflair :<https://data-flair.training/blogs/>

Analytics Vidhya :<https://www.analyticsvidhya.com/blog/>

```
"city": "London"  
}
```

List comprehensions provide a concise way to create lists

the following example calculates the power of 3 of each item in my_list

```
my_list=[1,2,3,4,5]  
power=list(map(lambda x:pow(x,3),my_list))  
print(power)
```

19. A namespace is a collection of names and information of the objects referenced by the name
20. The pass is a statement which doesn't do anything ; It avoids getting error when empty code is not allowed lie in loops definition
21. A unit test ensures that each part of the code delivers the desired output. In unit testing, developers only look at the interface and the specification for a component.
22. Slicing creates a new substring from the source string and the original string remains unchanged.
23. Index are used in array to pick item in the array, it start from 0 at the beginning of the array (from the left to the right) while negative indexes start from where the arrays ends,
24. Ternary operators called as well conditional operators are operators based on a condition being true or false

Example below :

```
new_list = []
```

```
for word in ['cat', 'dog', 'rat', 'cow']:
```

```
    if word != 'rat':
```

```
        new_list.append(word)
```

```
print(new_list)
```

References

W3SCHOOLS: <https://www.w3schools.com/python/>

GeeksforGeeks: <https://www.geeksforgeeks.org/>

Programiz : <https://www.programiz.com/python-programming>

Baeldung : <https://www.baeldung.com/cs/>

Dataflair : <https://data-flair.training/blogs/>

Analytics Vidhya : <https://www.analyticsvidhya.com/blog/>

25. *args is a non keyword arguments and **kwargs is a keyword arguments We use *args and **kwargs as an argument when we are unsure about the arguments numbers to pass in the function
26. range returns a Python list object while xrange returns an xrange objects
27. Flask is a micro web framework that provides useful tools and features to create web applications easily in Python
28. **Cluster INDEX** :A cluster index is the table itself, which enforces the order of the rows in the table.
A non-clustered index doesn't sort the physical data inside the table.
29. In a database, a deadlock is an unwanted situation in which two or more transactions are waiting indefinitely for one another to give up locks
30. In a database, a livelock is an unwanted situation in which two or more programs state change constantly therefor the same interaction repeats again and again and the program is running without stopping,

2, Python string methods

METHOD	DESCRIPTION	EXAMPLE
capitalize()	The first character is converted to upper case and the rest are converted to lower case	<pre>txt = "i HATE My life!" x = txt.capitalize() print (x)</pre> <p><u>output:</u> I hate my life!</p>
casefold()	The all string is converted in lower case	<pre>txt = "I HATE MY LIFE SO MUCH" x = txt.casefold() print(x)</pre>

References

W3SCHOOLS: <https://www.w3schools.com/python/>
 GeeksforGeeks: <https://www.geeksforgeeks.org/>
 Programiz : <https://www.programiz.com/python-programming>
 Baeldung : <https://www.baeldung.com/cs/>
 Dataflair : <https://data-flair.training/blogs/>
 Analytics Vidhya : <https://www.analyticsvidhya.com/blog/>

center()	The center() method will center align the string,	<pre>txt = "i hate my life" x = txt.center(20) print(x)</pre>
count()	Count() will count the number of time a value appears	<pre>txt = "I hate my life, my life is a joke" x = txt.count("life") print(x)</pre>
endswith()	The endswith() returns True if the string ends with a specific value, otherwise, it will return False	<pre>txt = "I hate my life !" x = txt.endswith("!") print(x)</pre>
find()	Finds() returns the first occurrence of value of a specific value, otherwise, it will return -1 if the value is not found	<pre>txt = "i hate my life." x = txt.find("hate") print(x)</pre>
format()	Format() allows to put the specified value within a string placeholder	<pre>user_name = 'Jenny_1995' age = 23 output = '{} is {} years old'.format(user_name, age) print(output)</pre>

References

W3SCHOOLS:<https://www.w3schools.com/python/>

GeeksforGeeks:<https://www.geeksforgeeks.org/>

Programiz :<https://www.programiz.com/python-programming>

Baeldung :<https://www.baeldung.com/cs/>

Dataflair :<https://data-flair.training/blogs/>

Analytics Vidhya :<https://www.analyticsvidhya.com/blog/>

index()	index() returns the position at the first occurrence of the specified value in a list	<pre>fruits = ['apple', 'banana', 'cherry'] x = fruits.index("cherry") print(x)</pre>
isalnum()	Isalnum returns true if all characters are alphanumeric	<pre>txt = "ihatemylife" txt2="i hate my life" x = txt.isalnum() x2=txt2.isalnum() print(x) print(x2)</pre>
isalpha()	Isalpha returns true if all characters are alphabet letters	<pre>txt = "Hanitra" txt2="Hanitra5" x = txt.isalpha() x2=txt2.isalpha() print(x) print(x2)</pre>
isdigit()	Isdigit returns true if all characters are digits,	<pre>txt = "55778" txt2="Hanitra5" x = txt.isdigit() x2=txt2.isdigit() print(x) print(x2)</pre>
islower()	islower returns true if all characters are in lower case,	<pre>txt = "I hate my life" txt2="i hate my life" x = txt.islower() x2=txt2.islower()</pre>

References

W3SCHOOLS:<https://www.w3schools.com/python/>

GeeksforGeeks:<https://www.geeksforgeeks.org/>

Programiz :<https://www.programiz.com/python-programming>

Baeldung :<https://www.baeldung.com/cs/>

Dataflair :<https://data-flair.training/blogs/>

Analytics Vidhya :<https://www.analyticsvidhya.com/blog/>

		<pre>print(x) print(x2)</pre>
isnumeric()	isnumeric returns true if all characters are numeric	<pre>txt = "I hate my life" txt2="15555" x = txt.isnumeric() x2=txt2.isnumeric() print(x) print(x2)</pre>
isspace()	isspace returns true if all characters are whitespaces	<pre>txt = "I hate my life" txt2=" " x = txt.isspace() x2=txt2.isspace() print(x) print(x2)</pre>
join()	Takes all items in an iterable and join them into one string,	<pre>myTuple = ("I", "hate", "my", "life") x = " ".join(myTuple) print(x)</pre>
lower()	The lower converts a all string in lower case	<pre>txt = "I HATE my LIFE" x = txt.lower() print(x)</pre>

References

W3SCHOOLS:<https://www.w3schools.com/python/>

GeeksforGeeks:<https://www.geeksforgeeks.org/>

Programiz :<https://www.programiz.com/python-programming>

Baeldung :<https://www.baeldung.com/cs/>

Dataflair :<https://data-flair.training/blogs/>

Analytics Vidhya :<https://www.analyticsvidhya.com/blog/>

lstrip()	Lstrip removes any leading characters as space	<pre>txt = " hate" x = txt.lstrip() print("i", x, "my life")</pre>
replace()	Replace replaces a value by another value	<pre>txt = "I hate my life" x = txt.replace("hate", "love") print(x)</pre>
rsplit()	Rsplrit splits a string into a list	<pre>txt = "i hate my life" x = txt.rsplit() print(x)</pre>
split()	split splits a string into a list	<pre>txt = "i hate my life" x = txt.split() print(x)</pre>
splitlines()	The splitlines() method splits a string into a list.	<pre>txt = "i hate my life"</pre>

References

W3SCHOOLS:<https://www.w3schools.com/python/>

GeeksforGeeks:<https://www.geeksforgeeks.org/>

Programiz :<https://www.programiz.com/python-programming>

Baeldung :<https://www.baeldung.com/cs/>

Dataflair :<https://data-flair.training/blogs/>

Analytics Vidhya :<https://www.analyticsvidhya.com/blog/>

	The splitting is done at line breaks.	<pre>x = txt.splitlines() print(x)</pre>
startswith()	Startswith returns true if the strings start with the specified value	<pre>txt = "i hate my life" txt2 = "Hello, welcome to my world." x = txt.startswith("Hello") x2 = txt2.startswith("Hello") print(x) print(x2)</pre>
strip()	Strip() removes any leading and trailing characters as spaces	<pre>txt = " hate " x = txt.strip() print("I", x, "my life")</pre>
swapcase()	Swapcase converts all characters in lower case to upper case and all characters in upper case to lower case	<pre>txt = "I HATE my life" x = txt.swapcase() print(x)</pre>
title()	Title() converts every first character of a word in upper case	<pre>txt = "i hate my life" x = txt.title()</pre>

References

W3SCHOOLS:<https://www.w3schools.com/python/>

GeeksforGeeks:<https://www.geeksforgeeks.org/>

Programiz :<https://www.programiz.com/python-programming>

Baeldung :<https://www.baeldung.com/cs/>

Dataflair :<https://data-flair.training/blogs/>

Analytics Vidhya :<https://www.analyticsvidhya.com/blog/>

		<code>print(x)</code>
<code>upper()</code>	Upper() converts all characters in a string in upper case	<pre>txt = "i hate my life" x = txt.upper() print(x)</pre>

3,Python list methods

Method	Description	Example
<code>append()</code>	Append() appends an item at the end of a list	<pre>fruits = ["apple", "banana", "cherry"] fruits.append("orange") print(fruits)</pre>
<code>clear()</code>	The <code>clear()</code> method removes all the elements from a list.	<pre>fruits = ['apple', 'banana', 'cherry', 'orange'] fruits.clear()</pre>
<code>copy()</code>	The <code>copy()</code> method returns a copy of the specified list.	<pre>fruits = ['apple', 'banana', 'cherry', 'orange']</pre>

References

W3SCHOOLS:<https://www.w3schools.com/python/>

GeeksforGeeks:<https://www.geeksforgeeks.org/>

Programiz :<https://www.programiz.com/python-programming>

Baeldung :<https://www.baeldung.com/cs/>

Dataflair :<https://data-flair.training/blogs/>

Analytics Vidhya :<https://www.analyticsvidhya.com/blog/>

		<pre>x = fruits.copy() print(x)</pre>
count()	the count() method returns the number of elements with the specified value.	<pre>fruits = ['apple', 'banana', 'cherry'] x = fruits.count("cherry") print(x)</pre>
extend()	The extend() method adds the specified list elements (or any iterable) to the end of the current list.	<pre>fruits = ['apple', 'banana', 'cherry'] cars = ['Ford', 'BMW', 'Volvo'] fruits.extend(cars) print(fruits)</pre>
index()	The index() method returns the position at the first occurrence of the specified value.	<pre>fruits = ['apple', 'banana', 'cherry'] x = fruits.index("cherry") print(x)</pre>
insert()	the insert() method inserts the specified value at the specified position.	<pre>fruits = ['apple', 'banana', 'cherry'] fruits.insert(1, "orange") print(fruits)</pre>
pop()	The pop() method removes the element at the specified position.	<pre>fruits = ['apple', 'banana', 'cherry']</pre>

References

W3SCHOOLS: <https://www.w3schools.com/python/>

GeeksforGeeks: <https://www.geeksforgeeks.org/>

Programiz : <https://www.programiz.com/python-programming>

Baeldung : <https://www.baeldung.com/cs/>

Dataflair : <https://data-flair.training/blogs/>

Analytics Vidhya : <https://www.analyticsvidhya.com/blog/>

		<pre>fruits.pop(1) print(fruits)</pre>
remove()	The remove() method removes the first occurrence of the element with the specified value.	<pre>fruits = ['apple', 'banana', 'cherry'] fruits.remove("banana") print(fruits)</pre>
reverse()	The reverse() method reverses the sorting order of the elements.	<pre>fruits = ['apple', 'banana', 'cherry'] fruits.reverse() print(fruits)</pre>
sort()	The sort() method sorts the list ascending by default.	<pre>cars = ['Ford', 'BMW', 'Volvo'] cars.sort() print(cars)</pre>

4,Python tuple methods

Method	description	example
count()	The count() method returns the number of times a specified value appears in the tuple.	<pre>this tuple = (1, 3, 7, 8, 7, 5, 4, 6, 8, 5) x = this tuple.count(5) print(x)</pre>
index()	The index() method finds	#index

References

W3SCHOOLS:<https://www.w3schools.com/python/>

GeeksforGeeks:<https://www.geeksforgeeks.org/>

Programiz :<https://www.programiz.com/python-programming>

Baeldung :<https://www.baeldung.com/cs/>

Dataflair :<https://data-flair.training/blogs/>

Analytics Vidhya :<https://www.analyticsvidhya.com/blog/>

	the first occurrence of the specified value.	<pre>thistuple = (1, 3, 7, 8, 7, 5, 4, 6, 8, 5) x = thistuple.index(8) print(x)</pre>
--	--	---

5,Python dictionary methods

Methods	Description	Example
clear()	The clear() method removes all the elements from a dictionary.	<pre>car = { "brand": "Ford", "model": "Mustang", "year": 1964 } car.clear() print(car)</pre>
copy()	The copy() method returns a copy of the specified dictionary.	<pre>car = { "brand": "Ford", "model": "Mustang", "year": 1964 } x = car.copy() print(x)</pre>
fromkeys()	The fromkeys() method returns a dictionary with the specified keys and the specified value.	<pre>x = ('key1', 'key2', 'key3') y = 0 thisdict = dict.fromkeys(x, y) print(thisdict)</pre>

References

W3SCHOOLS:<https://www.w3schools.com/python/>

GeeksforGeeks:<https://www.geeksforgeeks.org/>

Programiz :<https://www.programiz.com/python-programming>

Baeldung :<https://www.baeldung.com/cs/>

Dataflair :<https://data-flair.training/blogs/>

Analytics Vidhya :<https://www.analyticsvidhya.com/blog/>

get()	The get() method returns the value of the item with the specified key.	<pre>car = { "brand": "Ford", "model": "Mustang", "year": 1964 } x = car.get("model") print(x)</pre>
items()	The items() method returns a view object. The view object contains the key-value pairs of the dictionary, as tuples in a list.	<pre>car = { "brand": "Ford", "model": "Mustang", "year": 1964 } x = car.items() print(x)</pre>
keys()	The keys() method returns a view object. The view object contains the keys of the dictionary, as a list.	<pre>car = { "brand": "Ford", "model": "Mustang", "year": 1964 } x = car.keys() print(x)</pre>
pop()	The pop() method removes the specified item from the	<pre>car = { "brand": "Ford", "model": "Mustang",</pre>

References

W3SCHOOLS: <https://www.w3schools.com/python/>

GeeksforGeeks: <https://www.geeksforgeeks.org/>

Programiz : <https://www.programiz.com/python-programming>

Baeldung : <https://www.baeldung.com/cs/>

Dataflair : <https://data-flair.training/blogs/>

Analytics Vidhya : <https://www.analyticsvidhya.com/blog/>

	dictionary.	<pre>"year": 1964 } car.pop("model") print(car)</pre>
popitem()	The popitem() method removes the item that was last inserted into the dictionary	<pre>car = { "brand": "Ford", "model": "Mustang", "year": 1964 } car.popitem() print(car)</pre>
setdefault()	The setdefault() method returns the value of the item with the specified key.	<pre>car = { "brand": "Ford", "model": "Mustang", "year": 1964 } x = car.setdefault("model", "Bronco") print(x)</pre>
update()	The update() method inserts the specified items to the dictionary.	<pre>car = { "brand": "Ford", "model": "Mustang", "year": 1964 } car.update({"color": "White" </pre>

References

W3SCHOOLS:<https://www.w3schools.com/python/>

GeeksforGeeks:<https://www.geeksforgeeks.org/>

Programiz :<https://www.programiz.com/python-programming>

Baeldung :<https://www.baeldung.com/cs/>

Dataflair :<https://data-flair.training/blogs/>

Analytics Vidhya :<https://www.analyticsvidhya.com/blog/>

		<pre>"})) print(car)</pre>
values()	<p>values() method returns a view object that displays a list of all values in a given dictionary.</p>	<pre>sales = { 'apple': 2, 'orange': 3, 'grapes': 4 } print(sales.values())</pre>

6,Python set methods

Method	Description	Example
add()	Add an item to a set, using the add() method:	<pre>thisset = {"apple", "banana", "cherry"} thisset.add("orange") print(thisset)</pre>
clear()	The clear() method removes all elements in a set.	<pre>fruits = {"apple", "banana", "cherry"} fruits.clear() print(fruits)</pre>
copy()	The copy() method copies the set.	<pre>#copy fruits = {"apple", "banana",</pre>

References

W3SCHOOLS:<https://www.w3schools.com/python/>

GeeksforGeeks:<https://www.geeksforgeeks.org/>

Programiz :<https://www.programiz.com/python-programming>

Baeldung :<https://www.baeldung.com/cs/>

Dataflair :<https://data-flair.training/blogs/>

Analytics Vidhya :<https://www.analyticsvidhya.com/blog/>

		<pre>"cherry"} x = fruits.copy() print(x)</pre>
difference()	The difference() method returns a set that contains the difference between two sets.	<pre>#difference x = {"apple", "banana", "cherry"} y = {"google", "microsoft", "apple"} z = x.difference(y) print(z)</pre>
intersection()	The intersection() method returns a set that contains the similarity between two or more sets.	<pre>x = {"apple", "banana", "cherry"} y = {"google", "microsoft", "apple"} z = x.intersection(y) print(z)</pre>
issubset()	Return True if all items in set x are present in set y :	<pre>x = {"a", "b", "c"} y = {"f", "e", "d", "c", "b", "a"} z = x.issubset(y)</pre>

References

W3SCHOOLS:<https://www.w3schools.com/python/>

GeeksforGeeks:<https://www.geeksforgeeks.org/>

Programiz :<https://www.programiz.com/python-programming>

Baeldung :<https://www.baeldung.com/cs/>

Dataflair :<https://data-flair.training/blogs/>

Analytics Vidhya :<https://www.analyticsvidhya.com/blog/>

		<code>print(z)</code>
<code>issuperset()</code>	Return True if all items set <code>y</code> are present in set <code>x</code> :	<pre> x = {"f", "e", "d", "c", "b", "a"} y = {"a", "b", "c"} z = x.issuperset(y) print(z) </pre>
<code>pop()</code>	The <code>pop()</code> method removes a random item from the set.	<pre> #pop fruits = {"apple", "banana", "cherry"} fruits.pop() print(fruits) </pre>
<code>remove()</code>	The <code>remove()</code> method removes the specified element from the set.	<pre> fruits = {"apple", "banana", "cherry"} fruits.remove("banana") print(fruits) </pre>
<code>symmetric_difference()</code>	The <code>symmetric_difference()</code> method returns a set that contains all items from both set, but not the items that are present in both sets.	<pre> #symmetricdifference x = {"apple", "banana", "cherry"} y = {"google", "microsoft", "apple"} </pre>

References

W3SCHOOLS:<https://www.w3schools.com/python/>

GeeksforGeeks:<https://www.geeksforgeeks.org/>

Programiz :<https://www.programiz.com/python-programming>

Baeldung :<https://www.baeldung.com/cs/>

Dataflair :<https://data-flair.training/blogs/>

Analytics Vidhya :<https://www.analyticsvidhya.com/blog/>

		<pre>z = x.symmetric_difference(y) print(z)</pre>
union()	The union() method returns a set that contains all items from the original set, and all items from the specified set(s).	<pre>x = {"apple", "banana", "cherry"} y = {"google", "microsoft", "apple"} z = x.union(y) print(z)</pre>

7,Python file methods

Methods	Description	Example
read()	The read() method returns the specified number of bytes from the file	<pre>f = open("demofile.txt", "r") print(f.read())</pre>
readline()	The readline() method returns one line from the file.	<pre>f = open("demofile.txt", "r") print(f.readline())</pre>
readlines()	The readlines() method returns the content of the file line by line	<pre>f = open("demofile.txt", "r") print(f.readlines())</pre>
write()	Create a file and	<pre>poem="this is a file" f = open("demofile.txt", "w")</pre>

References

W3SCHOOLS:<https://www.w3schools.com/python/>

GeeksforGeeks:<https://www.geeksforgeeks.org/>

Programiz :<https://www.programiz.com/python-programming>

Baeldung :<https://www.baeldung.com/cs/>

Dataflair :<https://data-flair.training/blogs/>

Analytics Vidhya :<https://www.analyticsvidhya.com/blog/>

	write inside the file	<code>print(f.write(poem))</code>
writelines()	Rewrite inside the file	<pre>poem="this is a file 2" f = open("demofile.txt", "w") print(f.writelines(poem))</pre>

References

W3SCHOOLS:<https://www.w3schools.com/python/>

GeeksforGeeks:<https://www.geeksforgeeks.org/>

Programiz :<https://www.programiz.com/python-programming>

Baeldung :<https://www.baeldung.com/cs/>

Dataflair :<https://data-flair.training/blogs/>

Analytics Vidhya :<https://www.analyticsvidhya.com/blog/>