

Assignment -1
Python Programming

Assignment Date	19 September 2022
Student Name	Ms.C.Pavithra
Student Roll Number	113219071028
Maximum Marks	2 Marks

Question-1:

Split this string.

Solution:	
	s = "Hi there Sam!"
	print(s.split())
	#-----#
	#-----#



The screenshot shows a Python code editor interface. On the left, there are icons for Python and C. The central area has tabs for 'main.py' and 'Shell'. The code in 'main.py' is:

```
1 s = "Hi there Sam!"  
2 print(s.split())
```

The 'Run' button is highlighted in blue. In the 'Shell' window to the right, the output is shown as:

```
['Hi', 'there', 'Sam!']>
```

Question-2:

Use .format() to print the following string.

Output should be: The diameter of Earth is 12742 kilometers.

Solution:	
	planet = "Earth"
	diameter = 12742
	print("The diameter of {} is {} kilometers.".format(planet,diameter))
	#-----#
	#-----#

```

main.py
1 planet = "Earth"
2 diameter = 12742
3 print("The diameter of {} is {} kilometers.".format(planet,diameter))

```

The diameter of Earth is 12742 kilometers.
> |

Question-3:

In this nest dictionary grab the word "hello"

Solution:
d = {'k1':[1,2,3,['tricky':['oh','man','inception',{'target':[1,2,3,'hello']}]]}]
d['k1'][3]['tricky'][3]['target'][3]
#-----#
#-----#

```

main.py
1 d = {'k1':[1,2,3,['tricky':['oh','man','inception',{'target':[1,2,3,'hello']}]]]}
2 print(d['k1'][3]['tricky'][3]['target'][3])

```

hello
> |

Question-4.1:

Create an array of 10 zeros?

Solution:
import numpy as np
x=np.zeros(10)
print(x)
#-----#
#-----#

```

main.py
1 import numpy as np
2 print(np.zeros(10))
3

```

[0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
> |

Question-4.2:

Create an array of 10 fives?

Solution:

```
import numpy as np  
x=np.ones(10)*5  
print(x)  
#-----#  
#-----#
```

The screenshot shows a Jupyter Notebook interface. On the left, there are two icons: a Python logo and a 'C' logo. Next to them is the file name 'main.py'. Below the file name are three numbered lines of code: 1. import numpy as np, 2. print(np.zeros(10)), and 3. A blank line. To the right of the code area are three buttons: a copy icon, a refresh icon, and a 'Run' button. Further to the right is a 'Shell' tab. In the 'Shell' tab, the output is shown as [0., 0., 0., 0., 0., 0., 0., 0., 0., 0.] followed by a greater than sign (>).

Question-5:

Create an array of all the even integers from 20 to 35

Solution:

```
import numpy as np  
x=np.arange(20,36,2)  
print(x)  
#-----#  
#-----#
```

The screenshot shows a Jupyter Notebook interface. On the left, there are two icons: a Python logo and a 'C' logo. Next to them is the file name 'main.py'. Below the file name are three numbered lines of code: 1. import numpy as np, 2. print(np.arange(20,35,2)), and 3. A blank line. To the right of the code area are three buttons: a copy icon, a refresh icon, and a 'Run' button. Further to the right is a 'Shell' tab. In the 'Shell' tab, the output is shown as [20 22 24 26 28 30 32 34] followed by a greater than sign (>).

Question-6:

Create a 3x3 matrix with values ranging from 0 to 8

Solution:

```
import numpy as np  
x=np.arange(0,9).reshape(3,3)  
print(x)  
#-----#  
#-----#
```

The screenshot shows a Jupyter Notebook interface. On the left, there are icons for Python (Python logo), C/C++ (C/C++ logo), and C# (C# logo). The code cell contains the following Python code:

```
main.py  
1 import numpy as np  
2 print(np.arange(0,9).reshape(3,3))  
3
```

The "Run" button is highlighted in blue. To the right, the "Shell" output area displays the result of the code execution:

```
[[0 1 2]  
 [3 4 5]  
 [6 7 8]]  
> |
```

Question-7:

Concatenate a and b

```
a = np.array([1, 2, 3]), b = np.array([4, 5, 6])
```

Solution:

```
a=np.array([1,2,3])  
b=np.array([4,5,6])  
c=np.concatenate((a,b))  
print(c)  
#-----#  
#-----#
```

The screenshot shows a Jupyter Notebook interface. On the left, there are icons for Python (Python logo), C/C++ (C/C++ logo), and C# (C# logo). The code cell contains the following Python code:

```
main.py  
1 import numpy as np  
2 a = np.array([1, 2, 3])  
3 b = np.array([4, 5, 6])  
4 print(np.concatenate((a,b)))  
5
```

The "Run" button is highlighted in blue. To the right, the "Shell" output area displays the result of the code execution:

```
[1 2 3 4 5 6]  
> |
```

Question-8:

Pandas

Create a dataframe with 3 rows and 2 columns

Solution:

```
import pandas as pd  
d={'a':[1,2,3],  
   'b':[4,5,6]}  
y=pd.DataFrame(d)  
print(y)  
#-----#  
#-----#
```

The screenshot shows a Jupyter Notebook interface. On the left, there are three icons: a Python logo, a C/C++ logo, and a C# logo. The main area has tabs for 'main.py' and 'Shell'. The 'main.py' tab contains the following Python code:

```
1 import pandas as pd  
2 data = {'col_1': [5,7,2],  
3          'col_2': [8,3,6],  
4          }  
5 df=pd.DataFrame(data)  
6 print(df)
```

The 'Run' button is highlighted in blue. To the right, the 'Shell' tab displays the output of the code, which is a DataFrame:

	col_1	col_2
0	5	8
1	7	3
2	2	6

A command prompt symbol (>) is visible at the bottom of the shell window.

Question-9:

Generate the series of dates from 1st Jan, 2023 to 10th Feb, 2023

Solution:

```
import pandas as pd  
x=pd.date_range(start='1-1-2023',  
                 end='2-10-2023')  
for val in x:  
    print(val)  
#-----#  
#-----#
```

The screenshot shows a Jupyter Notebook interface. On the left, there are icons for file operations (New, Open, Save, Run Cell, Kernel, Help) and a cell type selector (Code, Markdown, Raw, Cell). The main area has a tab for 'main.py' containing the following Python code:

```
1 import pandas as pd
2 date=pd.date_range(start='01.01.2023',end='10.02.2023')
3 print(date)
```

Below the code are three small icons: a gear, a refresh, and a coffee cup. To the right is a 'Run' button. Further right is a 'Shell' section with the output of the code:

```
DatetimeIndex(['2023-01-01', '2023-01-02', '2023-01-03', '2023-01-04',
               '2023-01-05', '2023-01-06', '2023-01-07', '2023-01-08',
               '2023-01-09', '2023-01-10',
               ...
               '2023-09-23', '2023-09-24', '2023-09-25', '2023-09-26',
               '2023-09-27', '2023-09-28', '2023-09-29', '2023-09-30',
               '2023-10-01', '2023-10-02'],
              dtype='datetime64[ns]', length=275, freq='D')
```

A 'Clear' button is located in the top right corner of the shell area.

Question-10:

Create 2D list to DataFrame

```
lists = [[1, 'aaa', 22], [2, 'bbb', 25], [3, 'ccc', 24]]
```

Solution:

	lists = [[1, 'aaa', 22], [2, 'bbb', 25], [3, 'ccc', 24]]
	x=pd.DataFrame(lists)
	print(x)
	#-----#
	#-----#

The screenshot shows a Jupyter Notebook interface. On the left, there are icons for file operations (New, Open, Save, Run Cell, Kernel, Help) and a cell type selector (Code, Markdown, Raw, Cell). The main area has a tab for 'main.py' containing the following Python code:

```
1 import pandas as pd
2 lists = [[1, 'aaa', 22], [2, 'bbb', 25], [3, 'ccc', 24]]
3 df=pd.DataFrame(lists)
4 print(df)
```

Below the code are three small icons: a gear, a refresh, and a coffee cup. To the right is a 'Run' button. Further right is a 'Shell' section with the output of the code:

```
0   1   2
0   1   aaa  22
1   2   bbb  25
2   3   ccc  24
>
```