EXTERNSHIP ASSIGNMENT 1

Done By:

Lalitha Sri. B

20BCE7127.

1. Assign your Name to variable name and Age to variable age. Make a Python program that prints your name and age.

```python
def personal_details():
    name, age = "Lalitha", 20
    location = "visakhapatnam, Andhra Pradesh, India"
    print("Name: {}\nAge: {}\nAddress: {}".format(name, age, location))

personal_details()
```

```
Name: Lalitha
Age: 20
Address: visakhapatnam, Andhra Pradesh, India
```

2. X="Datascience is used to extract meaningful insights." Split the string

```python
X="Datascience is used to extract meaningful insights."

DataSplit = X.split()

print(DataSplit)
```

```
['Datascience', 'is', 'used', 'to', 'extract', 'meaningful', 'insights.']
```

3. Make a function that gives multiplication of two numbers

```python
no_1 = 4
no_2 = 6

product = no_1 * no_2

print(product)
```

```
24
```

4. Create a Dictionary of 5 States with their capitals. also print the keys and values.

```python
capital = {"Uttar Pradesh":"Lucknow", "Andhra Pradesh":"Amaravathi", "Kerala":"Thiruvananthapuram","Tamil Nadu":"Chennai","West Bengal":"Kol

for key in capital:
    print("Key = " +key+ ", Value = " +capital[key])
```

```
Key = Uttar Pradesh, Value = Lucknow
Key = Andhra Pradesh, Value = Amaravathi
Key = Kerala, Value = Thiruvananthapuram
Key = Tamil Nadu, Value = Chennai
Key = West Bengal, Value = Kolkata
```

5. Create a list of 1000 numbers using range function.

```python
ran = list(range(1,1001))
print(ran)
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32,
2, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70
80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106
4, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 13
145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 16
75, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197
205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226
234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254,
262, 263, 264, 265, 266, 267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282,
290, 291, 292, 293, 294, 295, 296, 297, 298, 299, 300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310,
8, 319, 320, 321, 322, 323, 324, 325, 326, 327, 328, 329, 330, 331, 332, 333, 334, 335, 336, 337, 338, 339
6, 347, 348, 349, 350, 351, 352, 353, 354, 355, 356, 357, 358, 359, 360, 361, 362, 363, 364, 365, 366, 36
4, 375, 376, 377, 378, 379, 380, 381, 382, 383, 384, 385, 386, 387, 388, 389, 390, 391, 392, 393, 394, 39
2, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413, 414, 415, 416, 417, 418, 419, 420, 421, 422, 423, 4
31, 432, 433, 434, 435, 436, 437, 438, 439, 440, 441, 442, 443, 444, 445, 446, 447, 448, 449, 450, 451, 4!
59, 460, 461, 462, 463, 464, 465, 466, 467, 468, 469, 470, 471, 472, 473, 474, 475, 476, 477, 478, 479, 4
87, 488, 489, 490, 491, 492, 493, 494, 495, 496, 497, 498, 499, 500, 501, 502, 503, 504, 505, 506, 507, 5
516, 517, 518, 519, 520, 521, 522, 523, 524, 525, 526, 527, 528, 529, 530, 531, 532, 533, 534, 535, 536, 5
44, 545, 546, 547, 548, 549, 550, 551, 552, 553, 554, 555, 556, 557, 558, 559, 560, 561, 562, 563, 564, 5
72, 573, 574, 575, 576, 577, 578, 579, 580, 581, 582, 583, 584, 585, 586, 587, 588, 589, 590, 591, 592, 5
```

6. Create an identity matrix of dimension 4 by 4

```python
import numpy as np

id= np.identity(4, dtype = float)
print("Matrix id : \n", id)
```

```
Matrix id :
 [[1. 0. 0. 0.]
 [0. 1. 0. 0.]
 [0. 0. 1. 0.]
 [0. 0. 0. 1.]]
```

7. Create a 3x3 matrix with values ranging from 1 to 9

```python
import numpy as np
matrix =  np.arange(1, 10).reshape(3,3)
print(matrix)
```

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

8. Create 2 similar dimensional array and perform sum on them.

```python
import numpy as np
matrix_1 =  np.arange(1, 10).reshape(3,3)
matrix_2 =  np.arange(2, 11).reshape(3,3)

matrix = matrix_1 + matrix_2
print(matrix)
```

```
[[ 3  5  7]
 [ 9 11 13]
 [15 17 19]]
```

9. Generate the series of dates from 1st Feb, 2023 to 1st March, 2023 (both inclusive)

```python
from datetime import timedelta, date

def daterange(date1, date2):
    for n in range(int ((date2 - date1).days)+1):
        yield date1 + timedelta(n)

start_dt = date(2023, 2, 1)
end_dt = date(2023, 3, 1)
for dt in daterange(start_dt, end_dt):
    print(dt.strftime("%Y-%m-%d"))
```

```
2023-02-01
2023-02-02
2023-02-03
2023-02-04
2023-02-05
2023-02-06
2023-02-07
2023-02-08
2023-02-09
2023-02-10
2023-02-11
2023-02-12
2023-02-13
2023-02-14
2023-02-15
2023-02-16
2023-02-17
2023-02-18
```

10. Given a dictionary, convert it into corresponding dataframe and display it
dictionary = {'Brand': ['Maruti', 'Renault', 'Hyndai'], 'Sales' : [250, 200, 240]

```python
import pandas as pd

dictionary = {'Brand': ['Maruti', 'Renault', 'Hyndai'], 'Sales' : [250, 200, 240]}
df = pd.DataFrame(list(dictionary.items()),columns = ['Brand','sales'])

print (df)
```

```
   Brand              sales
0  Brand  [Maruti, Renault, Hyndai]
1  Sales         [250, 200, 240]
```

****