**Basic Python**

**1. Split this string**

In [11]:

s **=** "Hi there Sam!"

In [12]:

*# Splits at space*

s**.**split()

Out[12]:

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

**2. Use .format() to print the following string.**

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

In [15]:

planet **=** "Earth"

diameter **=** 12742

In [16]:

*# Reverse the index numbers with the*

*# parameters of the placeholders*

'The diameter of {0} is {1} kilometer'**.**format(planet,diameter)

Out[16]:

'The diameter of Earth is 12742 kilometer'

**3. In this nest dictionary grab the word "hello"**

In [13]:

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

In [14]:

*#In this nest dictionary grabing the word "hello"*

print(d["k1"][3]["tricky"][3]["target"][3])

hello

**Numpy**

In [17]:

**import** numpy **as** np

**4.1 Create an array of 10 zeros?**

**4.2 Create an array of 10 fives?**

In [18]:

*#array of 10 zeros*

array1**=**np**.**zeros(10)

print(array1)

[0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]

In [19]:

*# array of 10 fives*

array2**=**np**.**ones(10)**\***5

print(array2)

[5. 5. 5. 5. 5. 5. 5. 5. 5. 5.]

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

In [20]:

*#array of all the even integers from 20 to 35*

array3**=**np**.**arange(20,36,2)

print(array3)

[20 22 24 26 28 30 32 34]

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

In [21]:

*#3x3 matrix with values ranging from 0 to 8*

x **=** np**.**arange(0, 9)**.**reshape(3,3)

print(x)

[[0 1 2]

[3 4 5]

[6 7 8]]

**7. Concatenate a and b**

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

In [22]:

a **=** np**.**array([1, 2, 3])

b **=** np**.**array([4, 5, 6])

*#Concatenate*

np**.**concatenate((a,b),axis**=None**)

Out[22]:

array([1, 2, 3, 4, 5, 6])

**Pandas**

**8. Create a dataframe with 3 rows and 2 columns**

In [23]:

**import** pandas **as** pd

In [24]:

A **=** np**.**random**.**randint(10, size**=**(3,2))

*#dataframe*

df **=** pd**.**DataFrame(A,columns**=**['cola', 'colb'])

df

Out[24]:

|  | **cola** | **colb** |
| --- | --- | --- |
| **0** | 4 | 3 |
| **1** | 8 | 5 |
| **2** | 4 | 5 |

In [25]:

dict\_a **=** {

'col\_a':[1,2,3],

'col\_b': [2,5,6],

}

*#dataframe*

df **=** pd**.**DataFrame(dict\_a)

df

Out[25]:

|  | **col\_a** | **col\_b** |
| --- | --- | --- |
| **0** | 1 | 2 |
| **1** | 2 | 5 |
| **2** | 3 | 6 |

In [26]:

lst\_a **=** [['John', 23], ['Jane', 25], ['Mary', 21]]

*#dataframe*

df **=** pd**.**DataFrame(lst\_a,columns**=**['Name', 'Age'])

df

Out[26]:

|  | **Name** | **Age** |
| --- | --- | --- |
| **0** | John | 23 |
| **1** | Jane | 25 |
| **2** | Mary | 21 |

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

In [27]:

**import** pandas **as** pd

*# calling DataFrame constructor*

df **=** pd**.**DataFrame()

*# Create 6 dates*

df['time'] **=** pd**.**date\_range(start**=**"1/1/2023",end**=**"2/10/2023", freq **=**'24H')

*# print dataframe*

*# Extract features - year, month, day, hour, and minute*

df['year'] **=** df['time']**.**dt**.**year

df['month'] **=** df['time']**.**dt**.**month

df['day'] **=** df['time']**.**dt**.**day

*# Show six rows*

df**.**head(len(df["time"]))

Out[27]:

|  | **time** | **year** | **month** | **day** |
| --- | --- | --- | --- | --- |
| **0** | 2023-01-01 | 2023 | 1 | 1 |
| **1** | 2023-01-02 | 2023 | 1 | 2 |
| **2** | 2023-01-03 | 2023 | 1 | 3 |
| **3** | 2023-01-04 | 2023 | 1 | 4 |
| **4** | 2023-01-05 | 2023 | 1 | 5 |
| **5** | 2023-01-06 | 2023 | 1 | 6 |
| **6** | 2023-01-07 | 2023 | 1 | 7 |
| **7** | 2023-01-08 | 2023 | 1 | 8 |
| **8** | 2023-01-09 | 2023 | 1 | 9 |
| **9** | 2023-01-10 | 2023 | 1 | 10 |
| **10** | 2023-01-11 | 2023 | 1 | 11 |
| **11** | 2023-01-12 | 2023 | 1 | 12 |
| **12** | 2023-01-13 | 2023 | 1 | 13 |
| **13** | 2023-01-14 | 2023 | 1 | 14 |
| **14** | 2023-01-15 | 2023 | 1 | 15 |
| **15** | 2023-01-16 | 2023 | 1 | 16 |
| **16** | 2023-01-17 | 2023 | 1 | 17 |
| **17** | 2023-01-18 | 2023 | 1 | 18 |
| **18** | 2023-01-19 | 2023 | 1 | 19 |
| **19** | 2023-01-20 | 2023 | 1 | 20 |
| **20** | 2023-01-21 | 2023 | 1 | 21 |
| **21** | 2023-01-22 | 2023 | 1 | 22 |
| **22** | 2023-01-23 | 2023 | 1 | 23 |
| **23** | 2023-01-24 | 2023 | 1 | 24 |
| **24** | 2023-01-25 | 2023 | 1 | 25 |
| **25** | 2023-01-26 | 2023 | 1 | 26 |
| **26** | 2023-01-27 | 2023 | 1 | 27 |
| **27** | 2023-01-28 | 2023 | 1 | 28 |
| **28** | 2023-01-29 | 2023 | 1 | 29 |
| **29** | 2023-01-30 | 2023 | 1 | 30 |
| **30** | 2023-01-31 | 2023 | 1 | 31 |
| **31** | 2023-02-01 | 2023 | 2 | 1 |
| **32** | 2023-02-02 | 2023 | 2 | 2 |
| **33** | 2023-02-03 | 2023 | 2 | 3 |
| **34** | 2023-02-04 | 2023 | 2 | 4 |
| **35** | 2023-02-05 | 2023 | 2 | 5 |
| **36** | 2023-02-06 | 2023 | 2 | 6 |
| **37** | 2023-02-07 | 2023 | 2 | 7 |
| **38** | 2023-02-08 | 2023 | 2 | 8 |
| **39** | 2023-02-09 | 2023 | 2 | 9 |
| **40** | 2023-02-10 | 2023 | 2 | 10 |

**10. Create 2D list to DataFrame**

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

In [28]:

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

In [29]:

*#2D list to DataFrame*

df **=** pd**.**DataFrame(lists, columns **=**['col1',"col2","col3"])

df

Out[29]:

|  | **col1** | **col2** | **col3** |
| --- | --- | --- | --- |
| **0** | 1 | aaa | 22 |
| **1** | 2 | bbb | 25 |
| **2** | 3 | ccc | 24 |