Topic 1 : Basic Syntax for Markdown

1. Putting Headings:

```
Select Markdown option instead of code and execute the
cell.
```

In [2]:

```
# Heading level 1
## Heading level 2
### Heading level 3
#### Heading level 4
##### Heading level 5
###### Heading level 6


# Write the above lines in a cell, choose the markdown instead of Code in
# compile the markdown and print below code.  DO NOT WRITE THESE LINES TO
```

Heading level 1

Heading level 2

Heading level 3

Heading level 4

Heading level 5

Heading level 6

2. Writing Bold letters

remove # in the below text for markdown

In [ ]:

```
# __two underscores before and after the text__
```

two underscores before and after the text

3. Writing italic letters

In [7]:

```
# _single underscore before and after the text_
```

single underscore before and after the text

4. Dotted Points

In [9]:
```
# - Dash before text
# - will print the text
# - with numbered point
```

- Dot before text
- will print the text
- with numbered point

5. Numbered points

In [12]:
```
# 1. sample
# 2. text
# 3. test
```

1. sample
2. text
3. test

6. Creating Tables:

In [13]:
```
# | Heading 1   | Heading 2   |
# | ----------- | ----------- |
# | 1       | xx       |
# | 2       | yy   |
```

Heading 1	Heading 2
1	xx
2	yy

7. Images:

In [23]:
```
# ![Apple](apple_image.jpg)
```

Apple

8. Underscore text

In [19]:
```
# <u>Underscore text </u>
```

Underscore text

9. Horizontal line

In [22]:
```
# ***
```

In [ ]:

```
```

Topic 2 : Basic Python Programming

Step 1: print "Hello World"

In [1]:

```
print("Hello World")
```

```
Hello World
```

Step 2: Create a Numeric Variable

In [2]:

```
a = 10
```

Step 3: Print above created variable

In [3]:

```
print(a)
```

```
10
```

Step 4: Print the type of above variable

In [4]:

```
print(type(a))
```

```
<class 'int'>
```

Step 5: Create a String Variable & Print

In [5]:

```
string = "Hello"
```

Step 5.1: Access 2nd element in above string

In [6]:

```
print(string[1])
```

```
e
```

Step 5.2: Find the length of above string

In [7]:

```
print(len(string))
```

```
5
```

Step 5.3: Create another String and compare the strings.

In [8]:

```
string2 = "World"
if string == string2:
    print("Both Strings are same.")
else:
    print("Both Strings are not same.")
```

```
Both Strings are not same.
```

Step 5.4: Perform Concatenation operation

In [9]:

```
string3 = string + string2
print(string3)
```

```
HelloWorld
```

Step 6: Create a list

In [10]:

```
sampleList = [3, 5, 6, 18, 10]
```

Step 6.1: Find the length of above list.

In [11]:

```
print(len(sampleList))
```

```
5
```

Step 6.2: Iterate above list using for loop.

In [12]:

```
for i in sampleList:
    print(i)
```

```
3
5
6
18
10
```

Step 6.3: Add another element in to above list

In [13]:

```
sampleList.append(22)
```

Step 6.4: Remove one element from above list

In [14]:

```
sampleList.remove(3)
```

Step 6.5: Find the minimum and maximum numbers exists in above list.

In [15]:

```
print(min(sampleList))
print(max(sampleList))
```

```
5
22
```

Step 7: Getting input as a string from User

In [16]:

```
name = input("Enter your name:")
```

```
Hello Likhitha !
```

Step 8: Displaying output: string

In [19]:

```
print("Hello " + name + " ! ")
```

```
Hello XXX!
```

Step 9: Getting input as a int from User

In [17]:

```
num = int(input("Enter a number: "))
```

```
You entered a number:  360
```

Step 10: Dispay the input

In [22]:

```
print(num)
```

```
360
```

Step 11: Different ways to print int and string

In [20]:

```
a = 20
b = 10

sum = a + b


# Several ways to print
print("The total sum of values a and b is: {}".format(sum))
print("The total sum of values a and b is: %d" %sum)
print("The total sum of values a and b is:", sum)
```

```
The total sum of values a and b is: 30
The total sum of values a and b is: 30
The total sum of values a and b is: 30
```

Step 12: Getting multiple inputs from the user:

In [18]:

```
a, b, c = input("Enter any three values: ").split()


print("First number: ", a)
print("Second number: ", b)
print("Third number: ", c)
```

```
First number:  1
Second number:  2
Third number:  3
```

Step 13: Define a function using "def"

In [24]:

```
def HelloWorld():
    print("Hello World")
```

Step 14: Call the above function

In [26]:

```
HelloWorld()
```

```
Hello World
```

In [ ]:

Topic 3 : Advanced Python Programming

1. pip install Command:

In [26]:

```
## Syntax: !pip install package-name

Example: !pip install pami
```

2. Reading and Processing .CSV file:

In [ ]:

```
# Reading a .csv file
import csv
file = open('fileName.csv')
csvreader = csv.reader(file)


# Printing Column names in csv file
header = next(csvreader)
print(header)
```

3. Reading CSV with pandas:

In [ ]:

```
# Reading csv file into a dataframe
import pandas as pd
df = pd.read_csv('inputFileName.csv')
# If you get error to open the file using the above code give absolute pa
# df = pd.read_csv('Exercise_1/inputFileName.csv')
# You can get absolute path by right click -> copy path.

# Printing the first rows of dataframe
print(df.head())

# Printing the column names
print(df.columns)

# Dropping columns from dataframe
print(df.drop('stationId', axis=1))
```

```
# Printing number of rows and columns in dataframe
print(df.shape)
```

4. How to add code description?

In [ ]:

```
# Add the description about the code using comments.

def hello_world():
    # A simple comment preceding a simple print statement
    print("Hello World")
```