

```
#Basic python codes.
```

```
#18/05/2020
```

```
#program to print current date and time
```

```
import datetime
```

```
print("The current Date and Time is:")
```

```
d=datetime.datetime.now()
```

```
print(d)
```

```
#use variables in python
```

```
x=10
```

```
y="20"
```

```
z=30.5
```

```
print(x,y,z)
```

```
sum1=x+x
```

```
sum2=y+y
```

```
sum3=z+z
```

```
sum4=x+z
```

```
print(sum1,sum2,sum3,sum4)
```

```
print(type(x),type(y),type(z))
```

```
#lists in python.
```

```
#Ex1
```

```
list=[1,2,3,4,5]
```

```
mysum=sum(list)
```

```
mylen=len(list)
```

```
mean=mysum / mylen
```

```
print(mean)
```

```
#Ex2
```

```
li=[1,2,3,3,4,5]
```

```
c=li.count(3)
```

```
print(c)
```

```
string="Hello007"
```

```
print(string.upper())
```

```
print(string.lower())
```

```
#dictionaries in python.
```

```
temp={"monday": 28,"tuesday": 29.5,"wednesday": 30,"thursday": 29.6,"friday":  
30.5,"saturday": 32.9,"sunday": 1000}
```

```
print(temp.values())
```

```
print(temp.keys())
```

```
#tuples in python.
```

```
tuple=(1,2,3)
```

```
print(tuple)
```

```
day_temperatures={'morning': (28.5,30.7,29.0),"noon": (30.9,32.0,31.2),"evening": (30.0,29.0,28.0)}
print(day_temperatures.values())
print(day_temperatures.keys())
```

#19/05/2020

#slicing,accessing items in list

```
letters = ['a', 'b', 'c', 'd', 'e', 'f', 'g']
print(letters[1:4])
letters = ['a', 'b', 'c', 'd', 'e', 'f', 'g']
print(letters[:3])
letters = ['a', 'b', 'c', 'd', 'e', 'f', 'g']
print(letters[-3:])
day_temperatures ={'morning': (28.5,30.7,29.0),"noon": (30.9,32.0,31.2),"evening": (30.0,29.0,28.0)}
print(day_temperatures["morning"])
```

#creating user defined functions in python.

#Ex1

```
def mean(mylist):
    mymean=sum(mylist) / len(mylist)
    print("Mean=")
    return mymean
print(mean([1,2,3,4]))
print(type(mean),type(sum))
```

#Ex2

```
def mean(value):
    if type(value) == dict:
        the_mean = sum(value.values()) / len(value)
    else:
        the_mean = sum(value) / len(value)
    return the_mean
temp = [29,30,31,32]
book = {"A":100,"B":250,"C":500}
mymean1 = mean(temp)
mymean2 = mean(book)
print(mymean1,"and",mymean2)
```

#Ex3

```
def mean(value):
    if isinstance(value,dict):
        the_mean = sum(value.values()) / len(value)
    else:
        the_mean = sum(value) / len(value)
    return the_mean
temp = [29,30,31,32]
```

```

book = {"A":100,"B":250,"C":500}
mymean1 = mean(temp)
mymean2 = mean(book)
print(mymean1,"and",mymean2)

#Ex4
def string(a):
    if(len(a)<8):
        return False
    else:
        return True
a=string("msdedfghfdgf")
print(a)

#Ex5
def temp(value):
    if (value > 25):
        return "Hot"
    elif (25>= value >=15):
        return "Warm"
    else:
        return "cold"
user_input = float(input("Enter the temperature:"))
print(temp(user_input))

#String formatting.
#Ex1
user_input = input("Enter your name:")
print("Hello " + user_input + "!")
print("Hello %s!" %user_input)
print(f"Hello {user_input}!") #Supports only for python 3.6 and above

#Ex2
F_name = input("Enter your first name:")
L_name = input("Enter your last name:")
print("Hello %s %s!" %(F_name,L_name))
print(f"Hello {F_name} {L_name}. What's up")

#program to print items in list,strings and dictionaries.
#Ex1
temperature = [29.4,30.6,28.3]
for temp in temperature:
    print(round(temp))
for letter in 'Hello':
    print(letter.title())
di = {"A":40, "B":50, "C":70}
for d in di.items():
    print(d)

```

```

for d in di.keys():
    print(d)

#Ex2
username = ''
while username != 'pypy':
    username = (input("Enter the username:"))

while True:
    username = input("Enter the username:")
    if(username == "pypy"):
        break
    else:
        continue

#program to take input from user and give formatted output.
def sentence_maker(phrase):
    interrogatives = ("how", "what", "why")
    caps = phrase.capitalize()
    if phrase.startswith(interrogatives):
        return "{}?".format(caps)
    else:
        return "{}.".format(caps)

result = []
while True:
    user_input = input("Say something:")
    if user_input == "\end":
        break
    else:
        result.append(sentence_maker(user_input))
print(" ".join(result))

#20/05/2020

#simple list comprehension
temps = [245, 304, 345, 298]
new_temps = [temp/10 for temp in temps]
print(new_temps)

#list comprehension with if conditionals.
def fun(li):
    new_li = [i for i in li if type(i) != str]
    return new_li
print(fun([9, 6, 'sxcd', 4]))

#list comprehension with if else conditionals.
#Ex1
def fun(li):

```

```

    new_li = [i if type(i) != str else 0 for i in li]
    return new_li
print(fun([45,'no data',45]))

#Ex2
def fun(li):
    new_li = [float(i) for i in li]
    return sum(new_li)
print(fun(['1','4','7']))

#Function arguments and parameters.
#Ex1
def fun(a,b):
    return a+b;
print(fun(4,5)) #non-keyword arguments.

#Ex2
def fun(a,b):
    return a/b;
print(fun(b=6,a=7)) #keyword arguments.

#Ex3
def fun(a,b=2): #default parameter.
    return a/b;
print(fun(4))

#Ex4
def mean(*args): #infinite non-keyword arguments can be passed.
    return sum(args) / len(args)
print(mean(1,2,3,4,6,7,8))

#program which takes indefinite no.of strings as parameters and returns a list
    containing all strings in uppercase and sorted alphabetically.
#Ex1
def fun(*args):
    args = [x.upper() for x in args]
    return sorted(args)
print(fun('AAa','Ccc','bbb'))

#Ex2
def fun(**kwargs): #infinite keyword arguments can be passed.
    return kwargs
print(fun(a=1,b=3,c=7)) #output is a dictionary.

#Ex3
def find_sum(**kwargs):
    return sum(kwargs.values())
print(find_sum(a=3,b=5,c=1))

```

```
#File processing operations.
#To read the file.
#Ex1
myfile = open("fruits.txt")
content = myfile.read()
myfile.close()
print(content)
print(content)

#Ex2
with open("fruits.txt") as myfile: #with context manager will close file implicitly
    content = myfile.read()
print(content)

#To write the file
with open("vegetables.txt","w") as myfile:
    myfile.write("cucumber\nonion\npotato\n")
    myfile.write("garlic")

#using os module to check whether file exists or not.
import time
import os
while True:
    if os.path.exists("fruits.txt"):
        with open("fruits.txt") as myfile:
            print(myfile.read())
    else:
        print("File does not exist")
        time.sleep(10)

#using pandas module to access the file.
import time
import os
import pandas
while True:
    if os.path.exists("temps_today.csv"):
        with open("temps_today.csv") as myfile:
            data = pandas.read_csv("temps_today.csv")
            print(data.mean())
    else:
        print("File does not exist")
        time.sleep(10)
```