We have learned about Basic Modules program -

(1)

```
''' program for importing Modules '''
import os
import tensorflow
print (" using Modules...!")
```

using import Statement -

```
• import <module_Name
```

```
• import <module1 >, <module 2>, ...<modulen
```

and here to access a particular method from any module

```
moduleName.functionName()
```

for ex -

```
import(Math) → (Modules)
math.sqrt (144)
```

output - 12.0

here os module means

operating system module. and it is provide functions for interacting with the operating system.

✱ Tensorflow module.

✱. used in machine learning as well as Open source library for numerical computation

✱ we can handle large data with the help of transflow module.

✱ camelcase module.

program :-

```
''' program for importing a module '''
import camelcase
c = camelcase.camelcase()
txt = "have happy thoughts :!!"
print (c.hump(txt))
```

This method capitalizes the first letter of each worlds.

hump →

convert strings and dictionary keys between camelcase

camelcase means -

ⓐ That one word joined that all start with capital letter.

* variables => used to store values

No command for declaring a variables

variable is created as soon as we assingn a value to it }

for ex=>

x = 5 # Here x is type of int

y = "RAM" # Here y is type of string.

Their are some rules for declaring variables

1) V.name must start with a Letter or Underscore
2) cannot start with Number
3) any contain - alpha-numeric. character underscores (A-Z, 0-9 and _).
4) V.names are case sensitive (age, Age and AGE are 3 diff v)
5) Can't be space among variables. Name.

Basic program -

```
X = 5
Y = "ABC"
print (x)
print (Y)
X = "XYZ"
print (X)
```

variables_Naming program

```
''' Program for valid and invalid variables
    names '''

# valid variables names
myvar = "HELLO...!!"
my_var = "HELLO"
_my_var= "HELLO"
myVar = "HELLO"
MYVAR = "HELLO"
myvar2 = "HELLO"
print (myvar)
print (my_var)
print (_my_var)
print (myVar)
print (MYVAR)
print (myvar2)
'''
# INVALID variables names
2myvar = "HELLO"      } Here start with Num
my_var = "HELLO"      } Here dash can't allowed
my var = "HELLO"      } Here space Not allowe

print (2myvar)
print (my-var)
print (my var)

'''
```

2) case sensitive variabes

It Just Take care aboute uppercase and Lowercase.

* when you don't type the same string the output will not create

program.

```
a = " small a"
A = " CAPTIAL A"
print ( a).
print (A)
```

OUTPUT -

```
small a
CAPTIAL A
```

A Check Datatype of variables

```
''' Program for checking the datatype of
    the variables '''
X = str(100000)
Y = int('500')
Z = Float(5000000)
 print(type(X))
 print (type (y))
 print (type (z))
```

Output

```
< class 'str' >
< class 'int' >
< class 'float' >
```

☆ Assigning values

(1) Program For assigning values to the variables

■ ☆ Assigning multiple values to multiple Variables -

☆
```
a = 50; b= "ABC" ; c = 5.45
print (a)
print (b)
print (c)
```

☆
```
col 1 ; col 2, col 3 = "red", "orang", "gree"
```
{ Here સાથે યુક્ત (combinly assign)

```
print(col 1)
Print (col 2)
print (col 3)
```

■ Assigning one value to multiple variables

☆
```
col 1 = col 2 = col 3 = "Yellow"

print (col 1)
print (col 2)
print (col 3)
```

PAGE NO.

DATE:

Assigning values From List to variables

```
colours = ["Yellow", "Green", "Red"]

print ("Assigning values From list to
        variables")

col1, col2, col3 = colours

print (col1)
print (col2)
print (col3)
```

Multiple variables in one statement

```
X = "Have"
Y = "Happy"
Z = "Thoughts"

print (x + Y + z)  } (combine words to senten
print (x, y, z)
```

output

```
Have Happy Thoughts
Have Happy Thoughts
```

- IN case of '+' operator

```
x = 100
Y = "ABC"
print (x+y)
```

output

```
100 ABC
```

Global variables

- That are created outside a function.
- can be used by everyone both inside fun. and outs

- program - outside of a fun.

```
1. x = "Pranali"
2 def myFunc():
3 print ("my Name is" + x )
4. myfunc()
```

output -

```
my Name is Pranali
```

- program - inside a fun

```
~~x = "Pranali"~~

x = "cold"

def myfunc():
x = "Sweet"
print (" icecream is" + x)

myfunc()

print (" iceream is" + x)
```

output -

```
~~pyt~~ icecream is cold
icecream is Sweet
```

* also use the global keyword if you want to change global variable inside a function

```
x = "cold"
def myfunc():
~~x =~~ global x
x = " sweet"
myfunc()
print ( ~~python~~" icecream is"+x)
```

output

```
~~python is fantas~~ icecream is sweet
```

- Type casting

used to convert the variable data type into certain data type

```
a = 7
print (type (a))
b = 3.0
print (type (b))
c = a + b
print (c)
print (type (c))
d = a * b
print (d)
print(type (d))
```

output

```
<class 'int'>
< class 'Float'>
10.0
< Class 'Float'>
21.0
<class Float'>
```

Task

(1) Type casting int to Float

(2) Type casting Float to into

due date - Friday | 5/8/2022