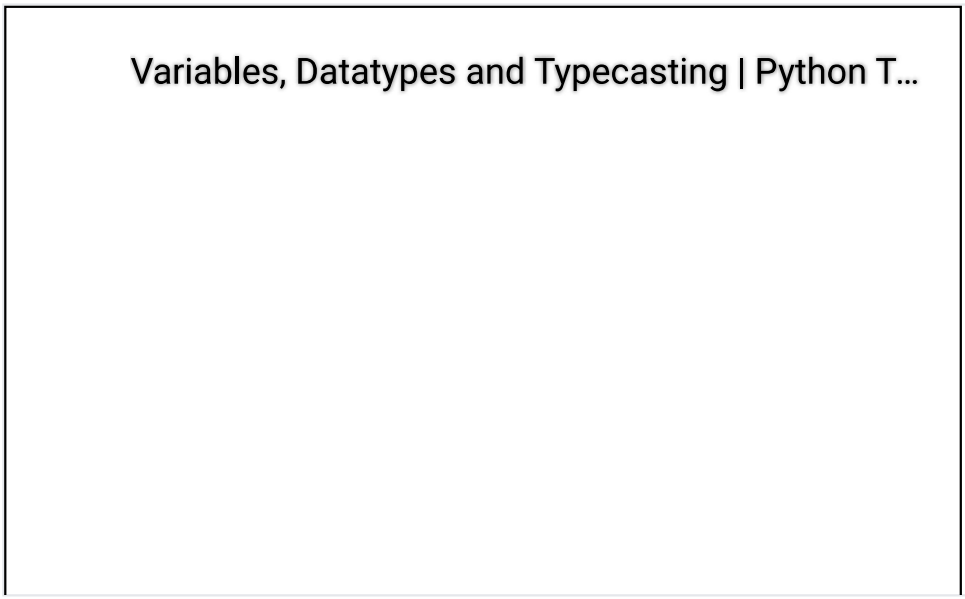

Show Course Contents

Overview Q&A Downloads Announcements

Variables, Datatypes and Typecasting

Variable:

A variable is a name given to any storage area or memory location in a program. In simple words, we can say that a variable is a container that contains some information, and whenever we need that information, we use the name of that container to access it. Let's create a variable:

```
a = 34 # Variable storing an integer
b = 23.2 # Variable storing real number
```

2. Floating point numbers (<class 'float'>): Used to store decimal or floating-point numbers
3. Strings (<class 'str'>): Used to store strings
4. Booleans (<class 'bool'>): Used to store True/False type values
5. None: None is literal to describe 'Nothing' in Python

Rules for defining a variable in Python:

- A variable name can contain alphabets, digits, and underscores (_). For E.g. : demo_xyz = 'It's a string variable'
- A variable name can only start with an alphabet and underscore.
- It can't start with a digit. For example, *5harry* is illegal and not allowed.
- No white space is allowed to be used inside a variable name.
- Also, reserved keywords are not recommended to be used as variable names.

Examples of few valid variable names are *harry*, *_demo*, *de_mo*, etc.
Python is a fantastic language that automatically identifies the type of data for us. It means we need to put some data in a variable, and Python automatically understands the kind of data a variable is holding. Cool, isn't it?
Have a look at the code below:

```
# Variable in Python:
abc = "It's a string variable"
_abcnum = 40 # It is an example of int variable
abc123 = 55.854 # It is an example of float variable
print(_abcnum + abc123) # This will give sum of 40 + 55.854
```

type() Function in Python: type() function is a function that allows a user to find data type of any variable. It returns the data type of any data contained in the variable passed to it.
Have a look at the code below, which depicts the use of type function:

```
print (type(harry)) #It will give output as string type
demo3 = type(demo) #It will return data type as float
print(demo3) #It will print that data type
print(type(demo2)) #It will give output as int type
```

Note – We can't do numbers with strings arithmetic operations, i.e., we can't add a string to any number. Have a look at the example below:

```
var1 = "It's a String"
var2 = 5
print(var1+var2) ''' It will give an error as we can't add string to a
```

Note – We can add (concatenate) two or more strings, and the strings will be concatenated to return another string. Here is the example showing that:

```
var1 = "My Name is "
var2 = "Harry"
var3 = var1+var2+" & I am a Good Boy."
print(var1+var2) # It will give output 'My Name is Harry'
print(var3)
```

Typecasting :

Typecasting is the way to change one data type of any data or variable to another datatype, i.e., it changes the data type of any variable to some other data type.
I know it's a bit confusing but let me tell you in a simple manner. Suppose there is a string "34" Note: String is not integer since it is enclosed in double-quotes) and as we know, we can't add this to an integer number, let's say 6. But to do so, we can typecast this string to int data type, and then we can add 34+6 to get the output as 40. Have a look at the program below:

```
abc3 = 55.95
xyz = 5.0

abc4=int(abc2)

print(abc+abc4) # Output : 50
print(abc+int(abc2)) # Output : 50

print(float(abc)+xyz) # It will add 5.0 + 5.0 and will return 10.0

print(str(abc)+45) # It will give an error as abc has been changed int
```

str() – this function allows us to convert some other data type into a string.
int() – this function allows us to convert some other data type into an integer. For example, str("34") returns 34 which is of type integer (int)
float() – this function allows us to convert some other data type into a floating point number, i.e., a number with decimals.

input() Function – This function allows the user to receive input from the keyboard into the program as a string.
input() function always takes input as a string, i.e., if we ask the user to take a number as input, even then, it will take it as a string, and we will have to typecast it into another data type as per the use case.
If you enter 45 when the input() is called, you will get "45" as a string

```
# Input Function in Python:
print("Enter your name : ")
name = input() #It will take input from user
print(Your Name is",name) # It will show the name
xyz = input(Enter your age : ")
print("Your age is ",xyz)
```

```
# Quiz :
print("Enter First Number : ")
num1= input()
print("Enter Second Number : ")
num2=input()
print("The Sum is",num2) #It will give output as sum of two numbers.
```

Code file as described in the video

```
var1 = "54"
var4 = "32"
var2 = 4
var3 = 36.7
# print(100 * str(int(var1) + int(var4)) )
# print(100 * "Hello world\n")
# print("Enter your number")
# inpnum = input()
#
# print("You entered", int(inpnum)+10)
"""
str()
int()
float()
"""

"""
Quiz - Solved in the video
Exercise - Next video
Project - Some awesome python utility
```

```
n1 = input()
print("Enter second number")
n2 = input()
print("Sum of these two numbers is", int(n1) + int(n2))
```

Previous Next

CodeWithHarry