



In Python, variables are used to store data values. Each variable has a data type, which defines the type of data it can hold. Here's a brief overview of variables, data types, and typecasting in Python:

Variables:

Variables are containers for storing data values. You can think of them as named storage locations that can hold different types of data. In Python, you don't need to explicitly declare the data type of a variable; the interpreter infers it based on the assigned value.

```
python Copy code

# Variable assignment
x = 10
y = "Hello, World!"
```

Data Types:

Python supports various built-in data types. Some common ones include:

Numeric Types:

- `int`: Integer type (e.g., 5, -10)
- `float`: Floating-point type (e.g., 3.14, -2.5)

Text Type:

- `str`: String type (e.g., "Hello", 'Python')

Boolean Type:

- `bool`: Boolean type (e.g., True, False)

Sequence Types:

- `list`: List type (e.g., [1, 2, 3])
- `tuple`: Tuple type (e.g., (1, 2, 3))
- `range`: Range type (e.g., range(0, 5))

Mapping Type:

- `dict`: Dictionary type (e.g., {'key': 'value'})

Set Types:

- `set`: Set type (e.g., {1, 2, 3})



Typecasting:

Typecasting involves converting a variable from one data type to another. Python provides several built-in functions for typecasting:

`int()`, `float()`, `str()`:

- Convert to integer, float, or string, respectively.

python

Copy code

```
x = 5.6
y = int(x) # y will be 5
```

`list()`, `tuple()`, `set()`:

- Convert to list, tuple, or set, respectively.

python

Copy code

```
my_list = [1, 2, 3]
my_tuple = tuple(my_list)
```

`bool()`:

- Convert to boolean.

python

Copy code

```
value = 0
boolean_value = bool(value) # boolean_value will be False
```

Remember to use typecasting carefully, as it may result in loss of data or unexpected behavior if not done correctly. It's always a good practice to check the type of a variable before performing typecasting.

Code:

```
var1 = "hello world"
var2 = 4
var3 = 36.7
print(type(var1))
print(type(var2))
```




Calculator Code:

```
print("Enter no 1:")
aa = input()
print("Enter no 2:")
bb = input()
cc = int(aa) + int(bb)
print("The sum of aa & bb :", cc)
```

Output:

```
C:\Users\test\PycharmProjects\project_1\.venv\Scripts\python.exe C:\Users\test\PycharmProjects\project_1\calculator.py
Enter no 1:
21
Enter no 2:
21
The sum of aa & bb : 42

Process finished with exit code 0
```

Code:

```
import turtle
import time

A = 5
print(A)

B = "hello"
print(B)

Credit_Card = 44445555666645434343

print(Credit_Card)

Test_turtle = turtle.Turtle()
Test_turtle.forward(100)
Test_turtle.right(90)
Test_turtle.forward(100)
```



```
Test_turtle.right(90)
Test_turtle.forward(100)
Test_turtle.right(90)
Test_turtle.forward(100)
Test_turtle.right(90)

time.sleep(10)
```

Output:

```
C:\Users\test\PycharmProjects\project_1\.venv\Scripts\python.exe C:\Users\test\PycharmProjects\project_1\variable.py
5
hello
44445555666645434343
```

