Variables and Data Types

What is a variable?

Variable is like a container that holds data. Very similar to how our containers in kitchen holds sugar, salt etc Creating a variable is like creating a placeholder in memory and assigning it some value. In Python its as easy as writing:

```
a = 1
b = True
c = "Harry"
d = None
```
These are four variables of different data types.

Local Vs Global Variables

Variables may be either global or local:

- Variables that are defined inside a function body have a local scope, and those defined outside have a global scope.

- This means that local variables can be accessed only inside the function in which they are declared, whereas global variables can be accessed throughout the program body by all functions.

The global Keyword:

We only need to use the **global keyword** in a function if we want to assign or change the global variable instead of creating a local variable.

global is not needed for printing and accessing.

Python “assumes” that we want a local variable due to the assignment to a variable inside of a function. Any variable which is changed or created inside of a function is local if it hasn’t been declared as a global variable. To tell Python, that we want to use

the global variable, we have to use the keyword **global**, as can be seen in the following example:

```python
a = 1

# Uses global because there is no local 'a'
def f():
    print('Inside f() : ', a)

# Variable 'a' is redefined as a local
def g():
    a = 2
    print('Inside g() : ', a)

# Uses global keyword to modify global 'a'
def h():
    global a
    a = 3
    print('Inside h() : ', a)

# Global scope
print('global : ', a)
```

```
f()

print('global : ', a)

g()

print('global : ', a)

h()

print('global : ', a)
```

```What is a Data Type?

Data type specifies the type of value a variable holds. This is required in programming to do various operations without causing an error.
In python, we can print the type of any operator using type function:

```
a = 1
print(type(a))
b = "1"
print(type(b))
```

By default, python provides the following built-in data types:

1. Numeric data: int, float, complex

- int: 3, -8, 0
- float: 7.349, -9.0, 0.0000001
- complex: 6 + 2i

2. Text data: str

str: "Hello World!!!", "Python Programming"

3. Boolean data:

Boolean data consists of values True or False.

4. Sequenced data: list, tuple

list: A list is an ordered collection of data with elements separated by a comma and enclosed within square brackets. Lists are mutable and can be modified after creation.

Example:

```
list1 = [8, 2.3, [-4, 5], ["apple", "banana"]]
print(list1)
```

Output:

```
[8, 2.3, [-4, 5], ['apple', 'banana']]
```

Tuple: A tuple is an ordered collection of data with elements separated by a comma and enclosed within parentheses. Tuples are immutable and can not be modified after creation.

Example:

```
tuple1 = (("parrot", "sparrow"), ("Lion", "Tiger"))
print(tuple1)
```

Output:

```
(('parrot', 'sparrow'), ('Lion', 'Tiger'))
```

5. Mapped data: dict

dict: A dictionary is an unordered collection of data containing a key:value pair. The key:value pairs are enclosed within curly brackets.

Example:

```
dict1 = {"name":"Sakshi", "age":20, "canVote":True}
print(dict1)
```

Output:

```
{'name': 'Sakshi', 'age': 20, 'canVote': True}
```