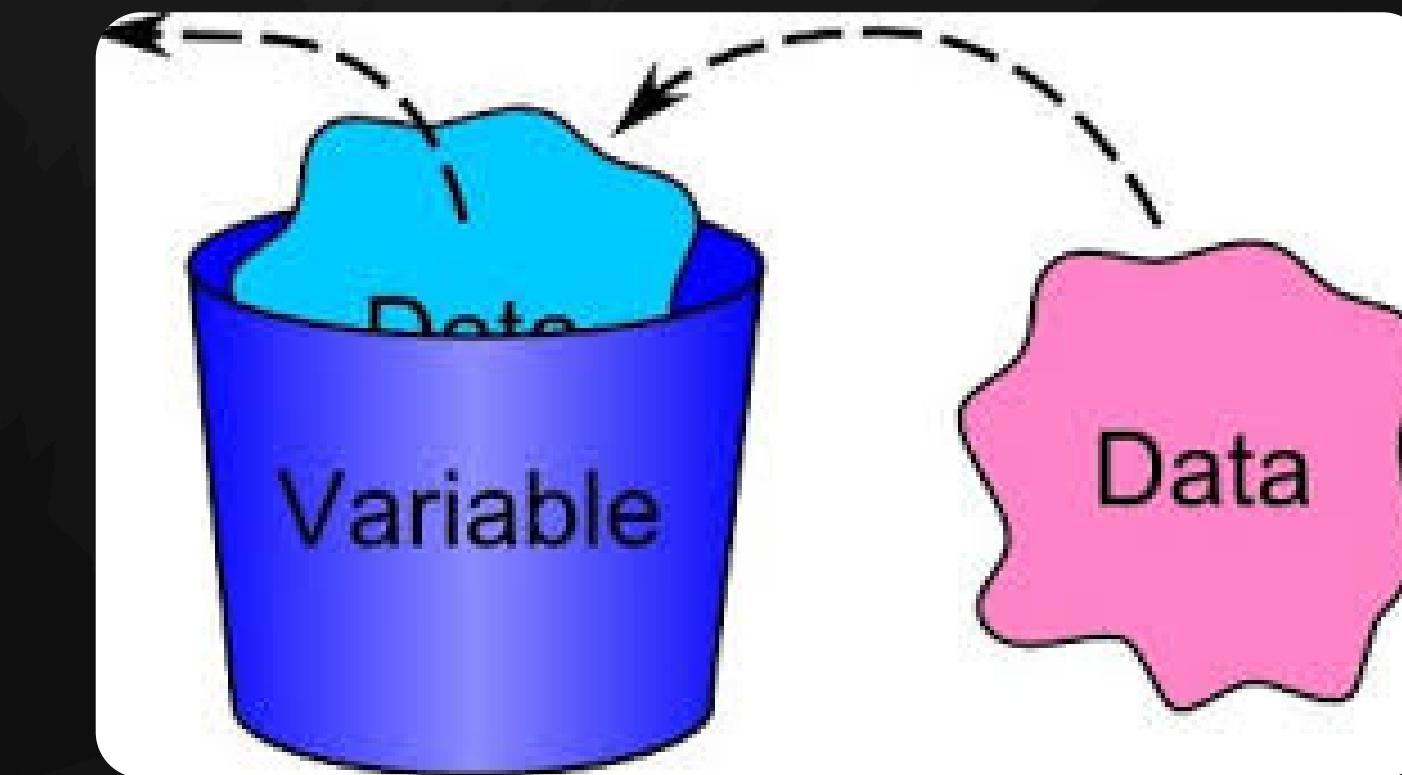


VARIABLES

VARIABLES TYPES

Presented by: Umar Mehmood



Variable

A variable is like a container that stores information in programming. Think of it as a box where you can put something (like a number, text, or value) and give it a name so you can use it later. For example:



Variable Types

1. Integer(int)
2. Float(float)
3. String(str)
4. Boolean(bool)
5. List(list)
6. Tuples(tuple)
7. Dictionary(dict)
8. Set(set)
9. None(None)
10. Complex(complex)



1. Integer(int)

Integer are whole numbers, Positive, Negative, without decimal points

```
1 # Example  
2 Age = 30  
3 print(type(Age))  
# <class 'int'>
```



2.Float(float)

Float Reprecent numbers that have a decimal point or
in a exponential form

```
1 # Example
2 Temperature = 20.2
3 print(type(Temperature))
# <class 'float'>
```

3.String(string)

String are sequences of characters enclosed in single quotes ('') or double quotes ("")

```
1 # Example
2 Greetings = 'Hello, World!'
3 print(type(Greetings))
# <class 'str'>
```



4.Boolean(bool)

Boolean represents one of two values True or False

```
1 # Example
2 isAdult = True
3 print(type(isAdult))
# <class 'bool'>
```

5.List(list)

list are ordered mutable collection of items which can be of any data types list are defined using square brackets[].

```
1 # Example
2 number = [1, 2, 3, 4, 5]
3 print(type(number))
# <class 'list'>
```

6. Tuples(tuple)

Tuple are ordered, immutable collection of items. Tuples are defined using parentheses()

```
1 # Example
2 Temperature = (20.2, 30.5,
3   40.6)
4 print(type(Temperature))
# <class 'tuple'>
```

7.Dictionary(dict)

Dictionary are unordered collection of key-value pairs, defined using curly brackets {}

```
1 # Example
2 person = {'name': 'John',
3            'age': 25}
4 print(type(person))
# <class 'dict'>
```

8. Set(set)

Sets are unordered collection of unique items,
defined using curly brackets {}

```
1 # Example
2 person = {'name': 'John',
3            'age': 25}
4 print(type(person))
# <class 'dict'>
```

9.None(NoneType)

None represent the absence of a value and is an object of its own data type

```
1 # Example
2 Result = None
3 print(type(Result))
# <class 'NoneType'>
```

10. Complex(complex)

Complex numbers are numbers with a real and imaginary part. The imaginary part is denoted by J.

```
1 # Example
2 Complex_number = 2 + 3j
3 print(type(Complex_number
    )) # <class 'complex'>
```

Thank You