3) Data Types in Python

By Badra

What is data type?

- Data types are the classification or categorization of data items. It represents the kind of value that tells what operations can be performed on a particular Data
- Variables can store data of different types, and different types can do different things.

Note: **Since everything is an object in Python programming, data types are actually classes and variables are instance (object) of these classes.**

Types of data in programming language

Text Type: **str**

Numeric Types: **int**, **float**, **complex**

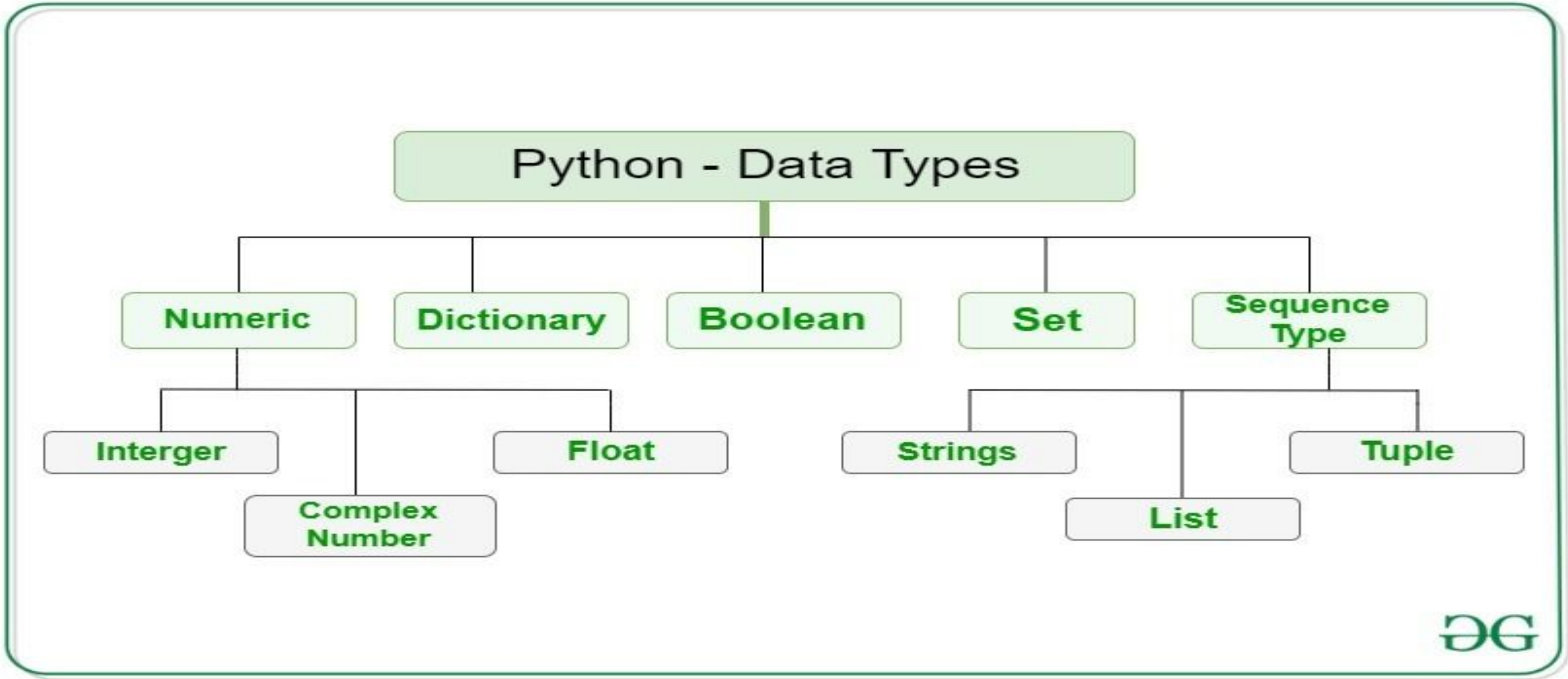
Sequence Types: **list**, **tuple**, **range**

Mapping Type: **dict**

Set Types: **set**, **frozenset**

Boolean Type: **bool**

Data types


Python - Data Types
Numeric
Dictionary
Boolean
Set
Sequence Type
Interger
Complex Number
Float
Strings
List
Tuple

Data Type - numeric (int, float, complex)

- In Python, numeric data type represent the data which has numeric value
- Numeric value can be integer, floating number or even complex numbers. These values are defined as **int**, **float** and **complex** class in Python.

Integers – This value is represented by int class. It contains positive or negative whole numbers (without fraction or decimal). In Python there is no limit to how long an integer value can be. Example: a = 10 (10 is the integer data type in python)

Numeric -

- **Float -** This value is represented by float class. It is a real number with floating point representation. It is specified by a decimal point.
- **Complex Numbers** – Complex number is represented by complex class. It is specified as *(real part) + (imaginary part)j*. For example – 2+3j
- Note: type() function is used to determine the type of data type.

Examples on numeric data types(int, float, complex)

- Create variable to store **int** data type value

  Ex: a = 10

  b = 20

- Create variable to store **float** data type value

  Ex: a = 10.5

  Salary = 1250.50

- Create variable to store **complex** data type
  Ex: a = 2 + 4j

Find the datatype by type() built-in function

# Write a program to find the data type of data

```
a = 5

print("Type of a: ", type(a))

b = 5.0

print("\nType of b: ", type(b))

c = 2 + 4j

print("\nType of c: ", type(c))
```

# output

```
Type of a:  <class 'int'>


Type of b:  <class 'float'>


Type of c:  <class 'complex'>
```

Sequence type in Python

String , List, Tuple

What is sequence in python?

- In Python, sequence is the ordered collection of similar or different data types.
- Sequences allows to store multiple values in an organized and efficient fashion.
- There are several sequence types in Python

1) string

2) list

3) tuple

Sequence type - string data type

- In Python, Strings are arrays of bytes representing Unicode characters.
- A string is a collection of one or more characters put in a single quote, double-quote or triple quote.
- In python there is no character data type, a character is a string of length one. It is represented by str class.

Creating String:

Strings in Python can be created using single quotes or double quotes or even triple quotes.

```
 # Create string with single quote
 Location = ‘Hyderabad’

# Create string with double quotes
Location = “Hyderabad”

# Create string with triple quotes
Location = “””Hyderabad”””
```

Example on strings

```
string1 = ‘welcome to python world’

print(string1)

name = “Badra”

print(name)

location = “””Hyderabad”””

print(location)
```

Output:

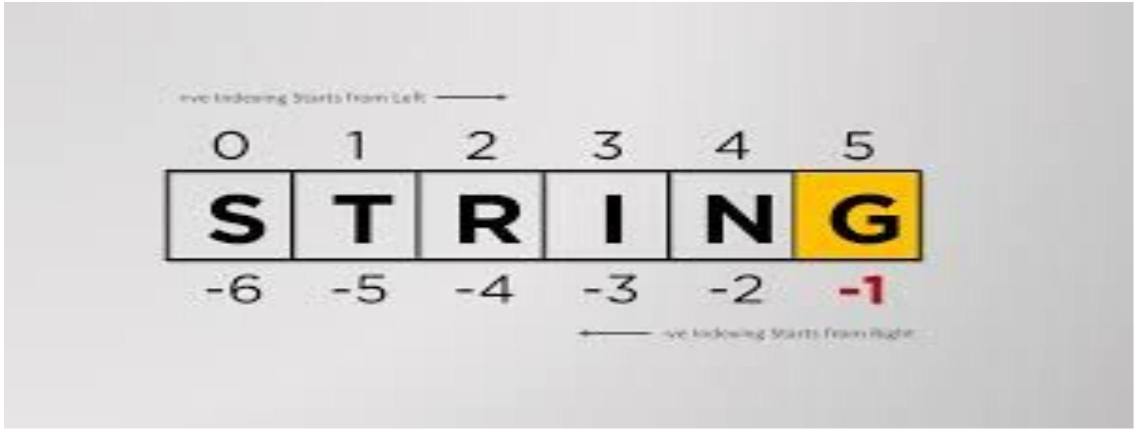
```
welcome to python world

Badra

Hyderabad
```

Accessing Elements of String

- String is a sequence data type, that means we can access the chars by using index position
- We can access the string in forward(left to right (+ve index value) and backward direction(right to left (-ve index value)



Python List Data Type

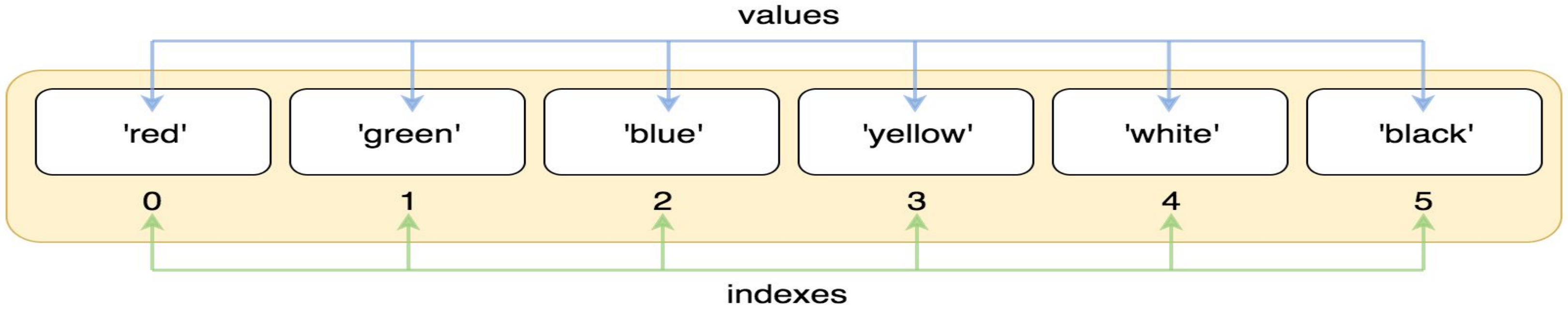
By Badra

Python List Data Type

- Lists are just like the arrays, declared in other languages which is a ordered collection of data.
- It is very flexible as the items in a list do not need to be of the same type.
- List is an mutable data types(can alter)
- List stores data in the form of index position as it an sequence data type
- Access elements by using index(+ve/-ve value)
- Lists in Python can be created by just placing the sequence inside the square brackets[].

  Example: colors = [‘red’, ‘green’, ‘blue’, ‘yellow’, ‘white’, ‘black’]

Accessing elements from list



colors[index] returns the value in that index position.

```
# access red color

print(colors[0])

# access blue color

print(colors[2])
```

Python Tuple data type

By Badra

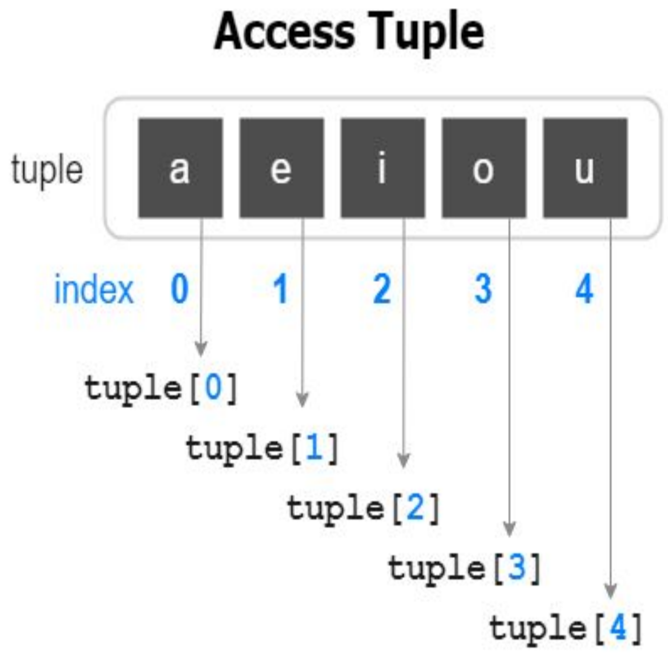
What is tuple?

- Python tuples are **a data structure that store an ordered sequence of values**.
- Tuples are immutable. This means you cannot change the values in a tuple
- Individual values in a tuple are called *items.* Tuples can store any data type.
- Tuple stores item in the form index position as it is also sequence type
- Access the elements/items from tuple by index value(+ve/-ve)
- A tuple is a comma-separated sequence of items. This sequence is surrounded by parenthesis *(())*. Means create tuple by using parenthesis ()

Create tuple:

```
vowels = ('a', 'e', 'i', 'o', 'u')
```

Access elements from tuple



```
vowels = ('a', 'e', 'i', 'o', 'u')

# Access first item from vowels tuple

print(vowels[0])

# access second item

print(vowels[1])

# access char ‘o’ from tuple

print(vowels[3])
```

Boolean data type

By Badra

Boolean data type(True/False)

- The Python Boolean type is one of Python’s built-in datatypes. It’s used to represent the truth value of an expression.

  For example, the expression `1 <= 2` is `True`, while the expression `0 == 1` is `False`. Understanding how Python Boolean values behave is important to programming well in Python.

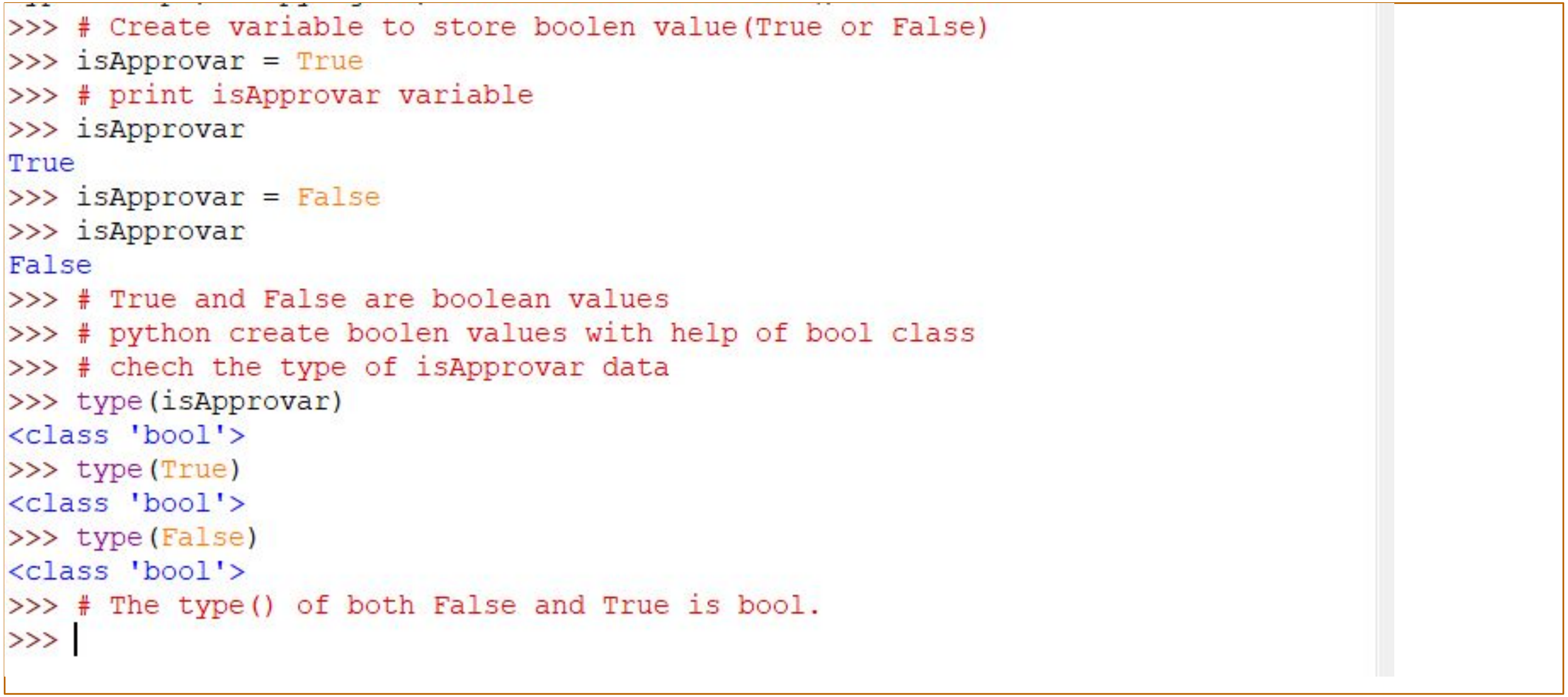
- The Python Boolean type has only two possible values:

  1) True

  2) False

  Note – True and False with capital ‘T’ and ‘F’ are valid booleans otherwise python will throw an error.

Examples on boolean data type

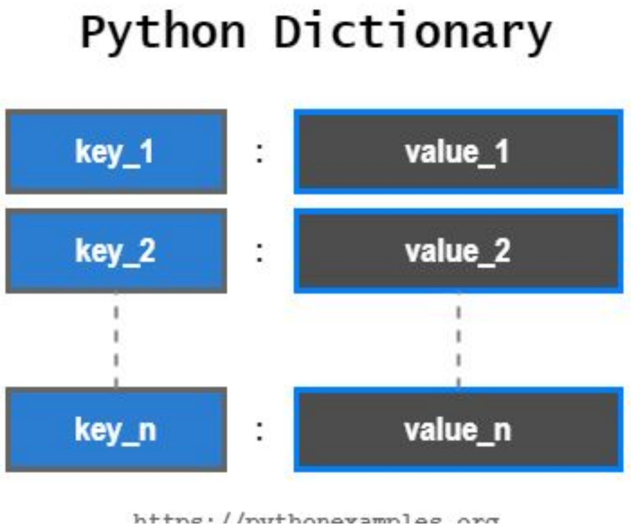


```
>>> # Create variable to store boolen value(True or False)
>>> isApprovar = True
>>> # print isApprovar variable
>>> isApprovar
True
>>> isApprovar = False
>>> isApprovar
False
>>> # True and False are boolean values
>>> # python create boolen values with help of bool class
>>> # chech the type of isApprovar data
>>> type(isApprovar)
<class 'bool'>
>>> type(True)
<class 'bool'>
>>> type(False)
<class 'bool'>
>>> # The type() of both False and True is bool.
>>> 
```

Python Dictionary Data Type

By Badra

Python dictionary data type ({ } or dict())



- Dictionary is a collection of key:value pairs.
- In a dictionary, keys are unique. Each key has an associated value. You can access value using key in a Dictionary.
- Key can be any immutable object in python like str, num . etc
- In Python, Dictionary can be created using curly brackets and comma separated key:value pairs.

Example of dict

Syntax:

```
<variable_name> = {‘key_1”:”Value_1”, ‘key_2’: ‘Value_2’}
```

Or

```
<variable_name> = dict(name=’badra’, age=22)
```

```
# Create a dict to store name and age
person_details = {
                     ‘name’: “Badra”,
                      ‘age’: 33
                    }
# Create a dict to store address data
my_address = { ‘plot_num’:  ‘1-88’,
                 ‘pin_code’: 589896,
                 ‘Landmark’:  ‘Near to bank’,
                ‘city’:  ‘hyderabad’
                }
```

Create dictionary and print it

```
# Create/initialize the dict


companyDetails = { ‘companyName’:  ‘Wipro’,
                   ‘location’:  ‘Hyderabad’,
                   ‘department’: ‘Operations’
                 }
# print data of companyDetails variable
print(companyDetails)
```

Output:

```
{ ‘companyName’:  ‘Wipro’,
  ‘location’:  ‘Hyderabad’,
   ‘department’: ‘Operations’
 }
```

Accessing data from dict

```
collegeDetails = {

        'collegeName': "cbit",

        'address': 'Hyderabad',

        'phoneNumber': "+91 9225123533"

}

# print the college name from above dict

college_name = collegeDetails["collegeName"]

print(college_name)
```

```
# Print College address

address = collegeDetails["address"]

print(address)


# Print the college phone number

phone_number =
collegeDetails["phoneNumber"]

print(phone_number )
```

Sets in Python

By Badra

Set data type

- A set is an unordered collection of items. Every item is unique in it. i.e., the Python set doesn’t allow duplicates.
- In Python, we can create set by using curly brackets {}

SYNTAX:

Myset = {1,2,3,3,4,5,6} # valid

MyInvalid = {}   # this treats as dict

Exercise 1 on set : Add a list of elements to a given set

```
sampleSet = {"Yellow", "Orange", "Black"}

sampleList = ["Blue", "Green", "Red"]

sampleSet.update(sampleList)

print(sampleSet)
```

Output:

```
{'Orange', 'Black', 'Yellow', 'Red', 'Blue', 'Green'}
```

Exercise 2: Return a new set of identical items from a given two set

Input:

```
set1 = {10, 20, 30, 40, 50}

set2 = {30, 40, 50, 60, 70}
```

Expected output:

```
{40, 50, 30}
```

Solution:

```
set1 = {10, 20, 30, 40, 50}
set2 = {30, 40, 50, 60, 70}


print(set1.intersection(set2))
```

Exercise 3: Returns a new set with all items from both sets by removing duplicates

Input:

```
set1 = {10, 20, 30, 40, 50}
set2 = {30, 40, 50, 60, 70}
```

Expected output:

```
{70, 40, 10, 50, 20, 60, 30}
```

Solution:

```
set1 = {10, 20, 30, 40, 50}

set2 = {30, 40, 50, 60, 70}

print(set1.union(set2))
```