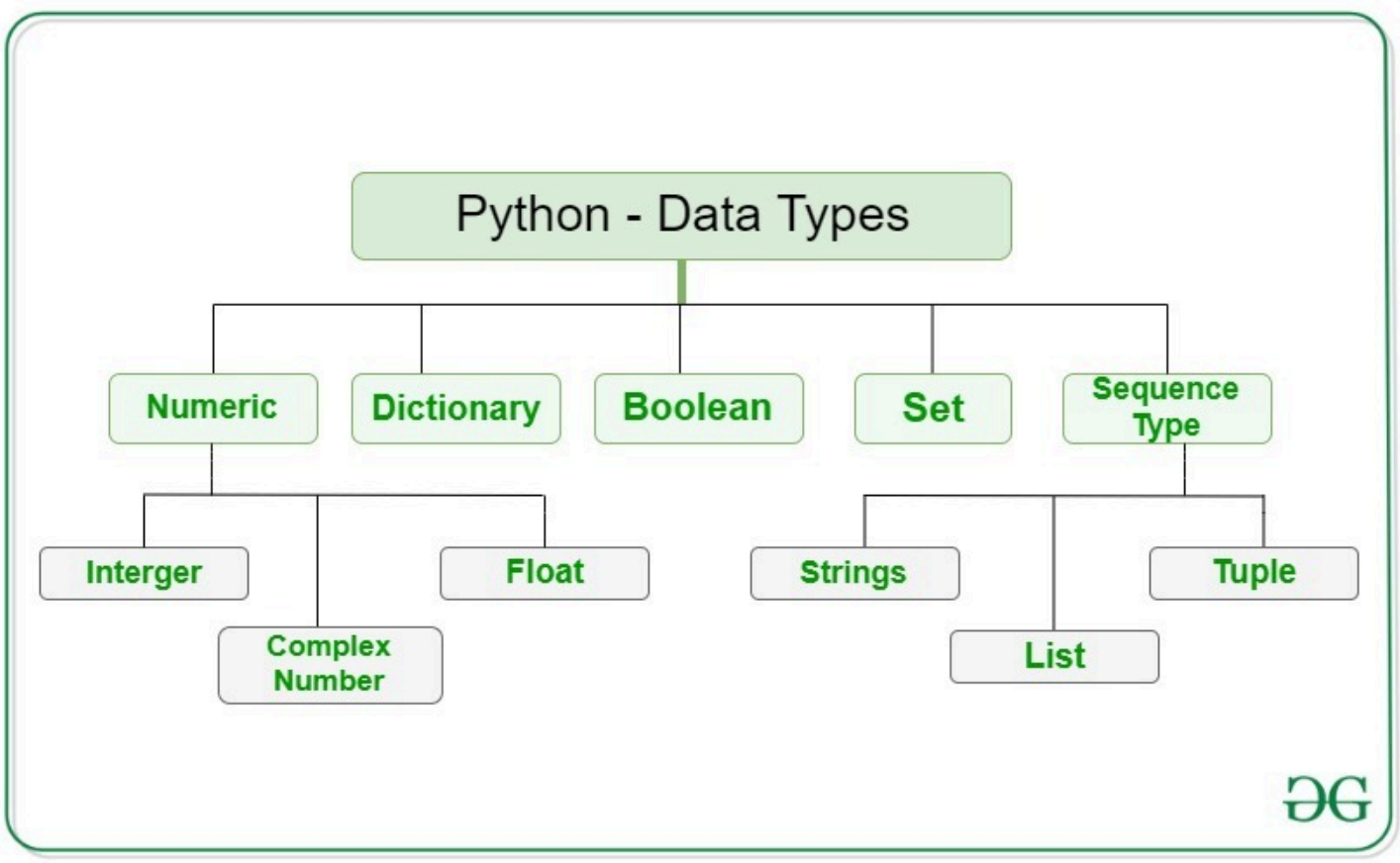
Data Types :-

- **Python Data types are the classification or categorization of data items. I represents the kind of value that tells what operations can be performed on a particular data. Since everything is an object in Python programming, Python data types are classes and variables are instances (objects) of these classes. The following are the standard or built-in data types in Python:**
- **Numeric**
- **Sequence Type**
- **Boolean**
- **Set**
- **Dictionary**



1. Numeric Data Types in Python:-

The numeric data type in Python represents the data that has a numeric value. A numeric value can be an integer, a floating number, or even a complex

number. These values are defined as Python int , Python float , and Python complex classes in Python .

- Integers – This value is represented by int class. It contains positive or negative whole numbers (without fractions or decimals). In Python, there is no limit to how long an integer value can be.
- Float – This value is represented by the float class. It is a real number with a floating-point representation. It is specified by a decimal point. Optionally, the character e or E followed by a positive or negative integer may be appended to specify scientific notation.
- Complex Numbers – A complex number is represented by a complex class. It is specified as *(real part) + (imaginary part)j* . For example – 2+3j

Example of Numeric 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'>
```

2. Sequence Data Types in Python

The sequence Data Type in Python is the ordered collection of similar or different Python data types. Sequences allow storing of multiple

values in an organized and efficient fashion. There are several sequence data types of Python:

String Data Type

Strings in Python 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 Python, a character is a string of length one. It is represented by str class.

Example for Sequence Data:-

```
String1 = 'Welcome to the Geeks World'

print("String with the use of Single Quotes: ")

print(String1)

String1 = "I'm a Geek"

print("\nString with the use of Double Quotes: ")

print(String1)

print(type(String1))

String1 = '''I'm a Geek and I live in a world of "Geeks"'''

print("\nString with the use of Triple Quotes: ")

print(String1)

print(type(String1))


String1 = '''Geeks
        For
        Life'''
```

```
print("\nCreating a multiline String: ")

print(String1)
```

Output:

```
String with the use of Single Quotes:
Welcome to the Geeks World
String with the use of Double Quotes:
I'm a Geek
<class 'str'>
String with the use of Triple Quotes:
I'm a Geek and I live in a world of "Geeks"
<class 'str'>
Creating a multiline String:
Geeks
        For
        Life
```

3. Boolean Data Type in Python

Python Data type with one of the two built-in values, True or False. Boolean objects that are equal to True are truthy (true), and those equal to False are falsy (false). However non-Boolean objects can be evaluated in a Boolean context as well and determined to be true or false. It is denoted by the class bool.

Example of Boolean Data Type:-

```
print(type(True))

print(type(False))

print(type(true))
```

Output:

```
<class 'bool'>
<class 'bool'>

Traceback (most recent call last):
  File "/home/7e8862763fb66153d70824099d4f5fb7.py", line 8, in
    print(type(true))
NameError: name 'true' is not defined
```

4. Set Data Type in Python:-

In Python Data Types, a Set is an unordered collection of data types that is iterable, mutable, and has no duplicate elements. The order of elements in a set is undefined though it may consist of various elements.

Example of Set Data Type :-

```
set1 = set()

print("Initial blank Set: ")

print(set1)

set1 = set("GeeksForGeeks")

print("\nSet with the use of String: ")

print(set1)

set1 = set(["Geeks", "For", "Geeks"])

print("\nSet with the use of List: ")

print(set1)

set1 = set([1, 2, 'Geeks', 4, 'For', 6, 'Geeks'])

print("\nSet with the use of Mixed Values")

print(set1)
```

Output:

```
Initial blank Set:
set()
Set with the use of String:
{'F', 'o', 'G', 's', 'r', 'k', 'e'}
Set with the use of List:
{'Geeks', 'For'}
Set with the use of Mixed Values
{1, 2, 4, 6, 'Geeks', 'For'}
```

5. Dictionary Data Type in Python:-

A dictionary in Python is an unordered collection of data values, used to store data values like a map, unlike other Python Data Types that hold only a single value as an element, a Dictionary holds a key: value pair. Key-value is provided in the dictionary to make it more optimized. Each key-value pair in a Dictionary is separated by a colon : , whereas each key is separated by a 'comma'.

Example of Dictionary Data Type:-

```
Dict = {}

print("Empty Dictionary: ")

print(Dict)

Dict = {1: 'Geeks', 2: 'For', 3: 'Geeks'}

print("\nDictionary with the use of Integer Keys: ")

print(Dict)

Dict = {'Name': 'Geeks', 1: [1, 2, 3, 4]}

print("\nDictionary with the use of Mixed Keys: ")

print(Dict)
```

```
Dict = dict({1: 'Geeks', 2: 'For', 3: 'Geeks'})

print("\nDictionary with the use of dict(): ")

print(Dict)

Dict = dict([(1, 'Geeks'), (2, 'For')])

print("\nDictionary with each item as a pair: ")

print(Dict)
```

Output:

```
Empty Dictionary:
{}
Dictionary with the use of Integer Keys:
{1: 'Geeks', 2: 'For', 3: 'Geeks'}
Dictionary with the use of Mixed Keys:
{1: [1, 2, 3, 4], 'Name': 'Geeks'}
Dictionary with the use of dict():
{1: 'Geeks', 2: 'For', 3: 'Geeks'}
Dictionary with each item as a pair:
{1: 'Geeks', 2: 'For'}
```