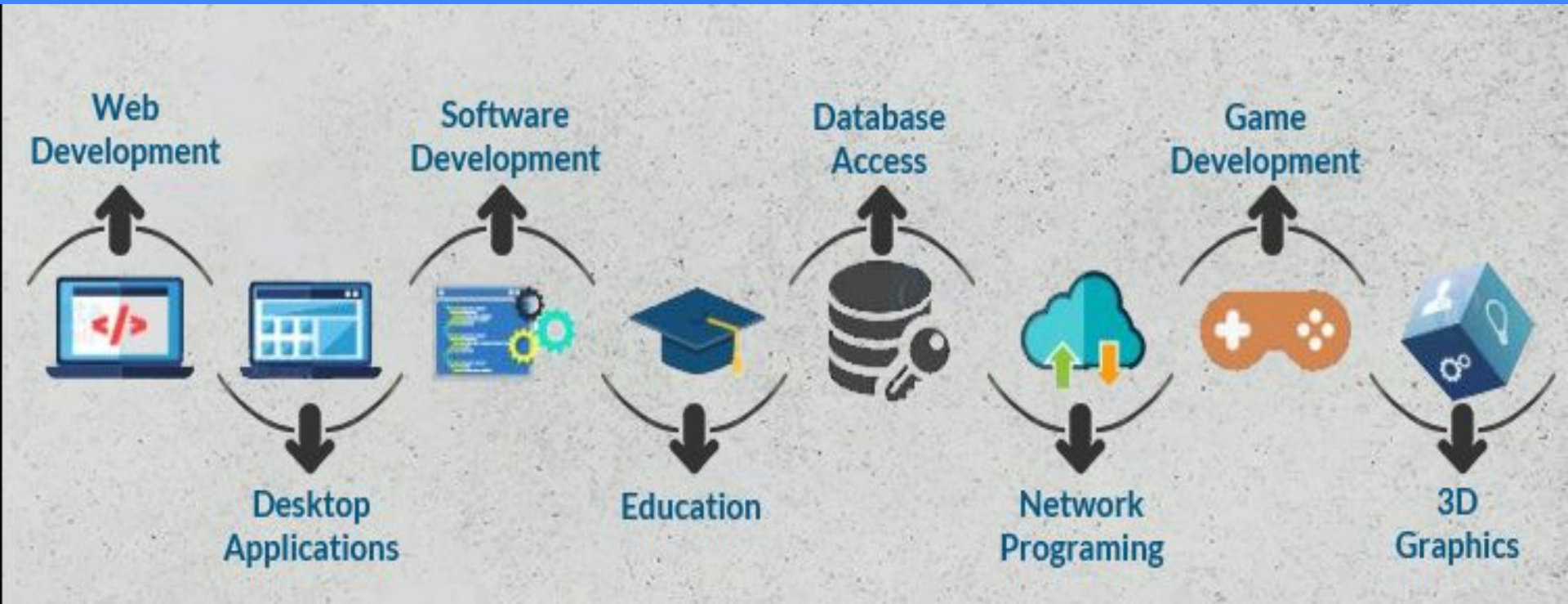
Lesson: Python Development

DS260222A

Python

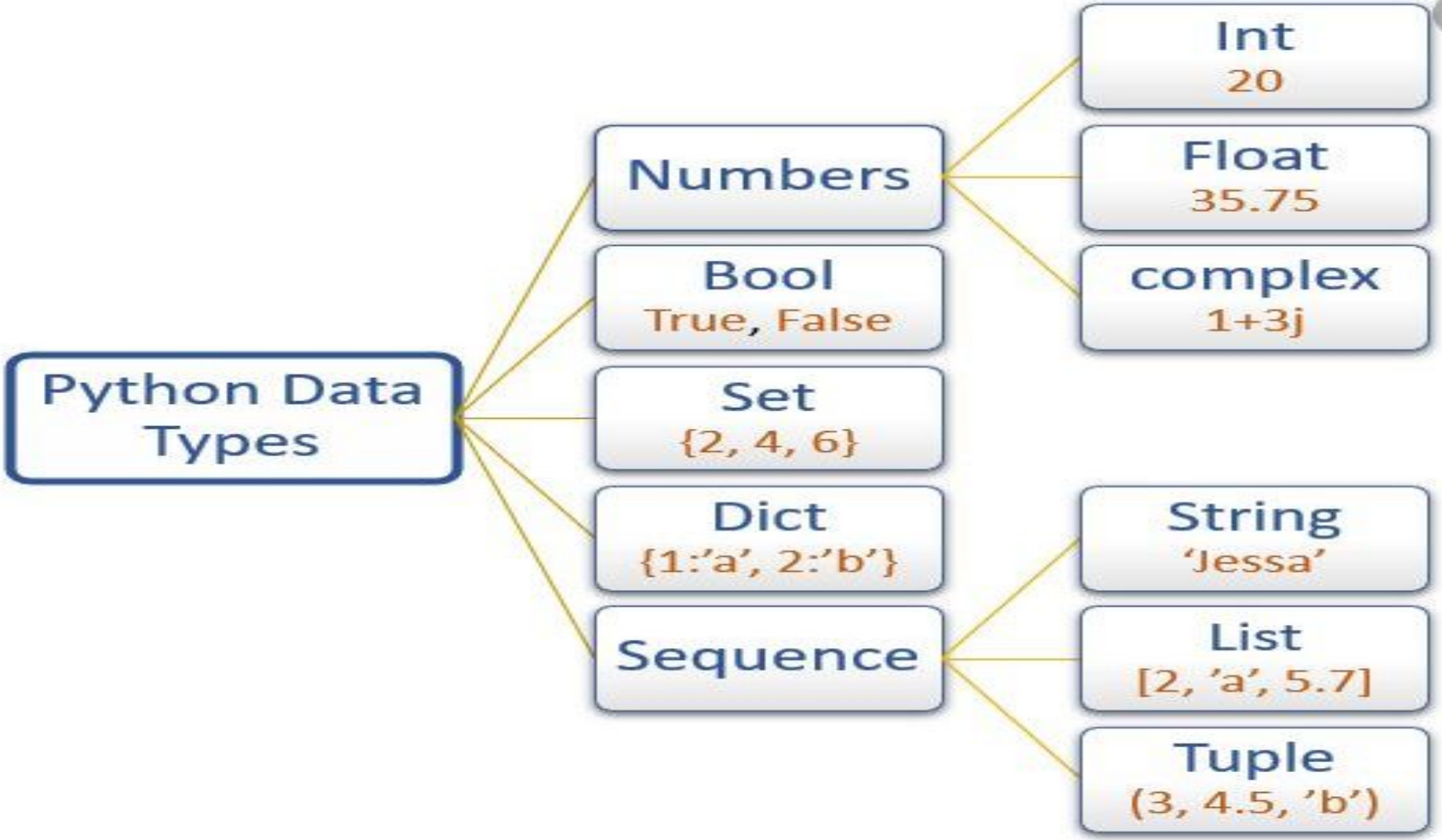
- Python Applications
- Python Features
- Data Types

Python Applications

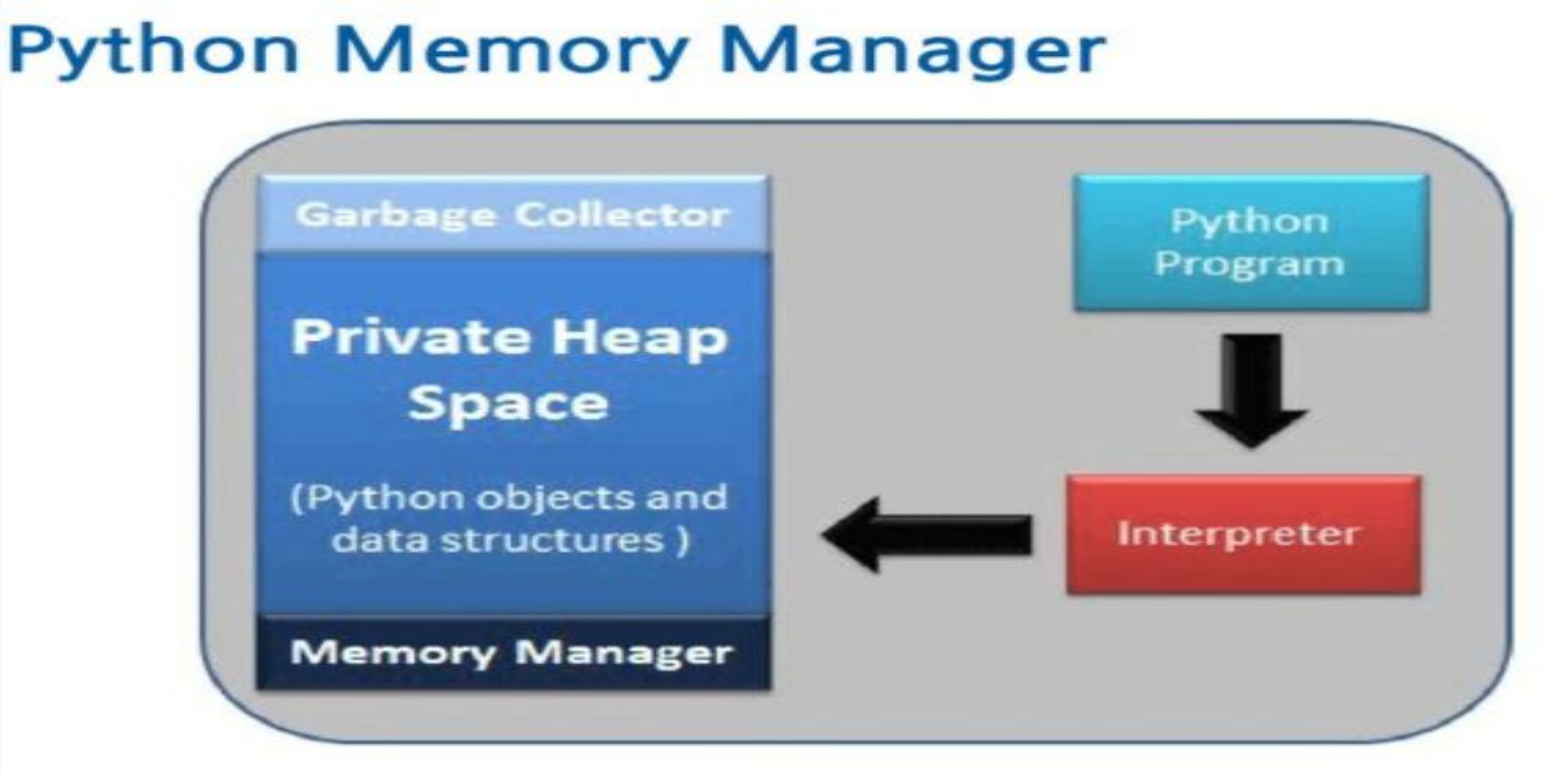


Python Features

- Easy to Learn, Free and Open Source.
- High Level Programming
- Object Oriented
- Dynamically typed
- Data Science and Machine Learning etc..,


Python Data Types
Numbers
Int
20
Float
35.75
complex
1+3j
Bool
True, False
Set
{2, 4, 6}
Dict
{1:'a', 2:'b'}
Sequence
String
'Jessa'
List
[2, 'a', 5.7]
Tuple
(3, 4.5, 'b')

Python Memory Management



Numeric Data Type

Name	Example	Usage
→ Integer Data-type	1, 2, 10, 33, 45, etc	To store the integer values
→ Float Data-type	1.5, 2.3, 6.1, etc	To store float values
→ Complex numbers	2+1j, 9+5j, etc	To store Complex numbers

- Integer data-type is used in many areas like while iterating through loops mathematical operations in real numbers, etc
- Float data-type are used to store the decimal values which are ignored nu int data-type.
- Complex numbers are used to perform some complex operations and scientific calculations

Operators

- Arithmetic operators
- Comparison operators
- Assignment Operators
- Logical Operators
- Bitwise Operators
- Membership Operators
- Identity Operators

Arithmetic Operator

Operator	Description
+ (Addition)	It is used to add two operands. For example, if a = 20, b = 10 => a+b = 30
- (Subtraction)	It is used to subtract the second operand from the first operand. If the first operand is less than the second operand, the value results negative. For example, if a = 20, b = 10 => a - b = 10
/ (divide)	It returns the quotient after dividing the first operand by the second operand. For example, if a = 20, b = 10 => a/b = 2.0
*** (Multiplication)**	It is used to multiply one operand with the other. For example, if a = 20, b = 10 => a * b = 200
% (reminder)	It returns the reminder after dividing the first operand by the second operand. For example, if a = 20, b = 10 => a%b = 0
**** (Exponent)**	It is an exponent operator represented as it calculates the first operand power to the second operand.
// (Floor division)	It gives the floor value of the quotient produced by dividing the two operands.

Comparison Operator

Operator	Description
==	If the value of two operands is equal, then the condition becomes true.
!=	If the value of two operands is not equal, then the condition becomes true.
<=	If the first operand is less than or equal to the second operand, then the condition becomes true.
>=	If the first operand is greater than or equal to the second operand, then the condition becomes true.
>	If the first operand is greater than the second operand, then the condition becomes true.
<	If the first operand is less than the second operand, then the condition becomes true.

Logical Operator

Operator	Description
and	If both the expression are true, then the condition will be true. If a and b are the two expressions, a → true, b → true => a and b → true.
or	If one of the expressions is true, then the condition will be true. If a and b are the two expressions, a → true, b → false => a or b → true.
not	If an expression **a** is true, then not (a) will be false and vice versa.

Bitwise

Operator	Description
& (binary and)	If both the bits at the same place in two operands are 1, then 1 is copied to the result. Otherwise, 0 is copied.
| (binary or)	The resulting bit will be 0 if both the bits are zero; otherwise, the resulting bit will be 1.
^ (binary xor)	The resulting bit will be 1 if both the bits are different; otherwise, the resulting bit will be 0.

Python Membership Operators

Membership operators are used to test if a sequence is presented in an object:

Operator	Description	Example
in	Returns True if a sequence with the specified value is present in the object	x in y
not in	Returns True if a sequence with the specified value is not present in the object	x not in y

Identity Operators

Python Identity Operators

Identity operators are used to compare the objects, not if they are equal, but if they are actually the same object, with the same memory location:

Operator	Description	Example
is	Returns True if both variables are the same object	x is y
is not	Returns True if both variables are not the same object	x is not y