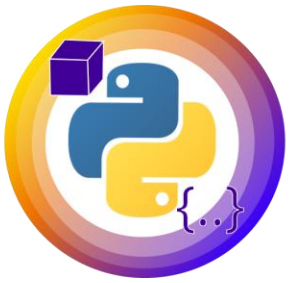
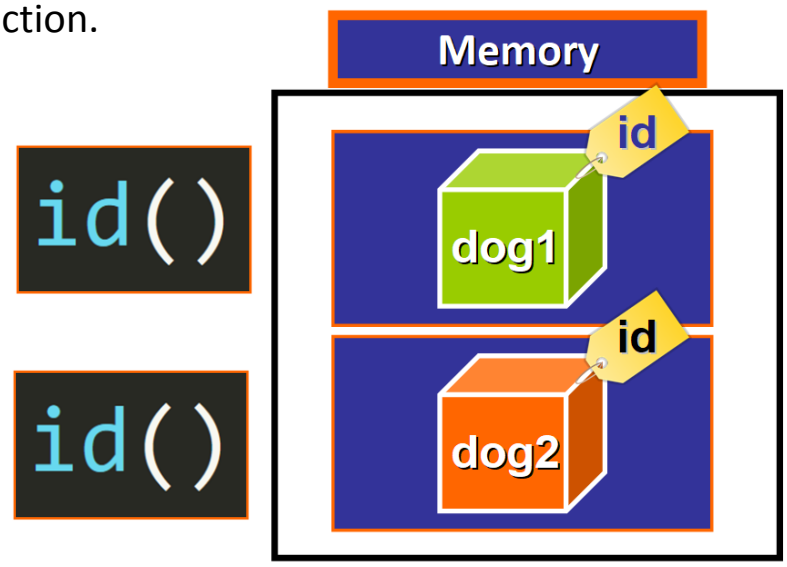


Python OOP: Objects in Memory

Objects in Memory

Key Takeaways

- **Objects in Memory:**
  - "Everything in Python is an Object."
  - Objects are stored in memory.
  - Each object has a **unique number** assigned that represents the memory address where it is currently stored. This number is called the "id" of the object.
  - You can get this value using the **id()** function.
    - Syntax: id(<object_var>)
  - Objects are passed **by reference** to avoid making copies of the objects every time that you pass them as arguments to a function.



Key Takeaways

- **The "is" operator:**
  - Returns **True** when the two operands **point to the same object** in memory.
  - Returns **False** if the operands **point to different objects** in memory.
  - There are **a few exceptions** to optimize memory usage:
    - Small integers in a range from [-5, 256] are retrieved from the same existing object to avoid creating new objects whenever you use an integer.
    - Certain strings are represented with the same object in memory to avoid creating several different objects to represent the same string.