Outlines

• Function: id(.)
• Mutable and immutable types
• Quiz walkthrough
• Homework question
Memory Visualization
Function: id(.)

- Return the “identity” of an object.
- Show the address of the object in memory.
- An integer (or long integer) which is guaranteed to be unique and constant for this object during its lifetime.
Immutability

• By immutability, we mean that whenever we start manipulating an immutable item, Python will spawn off another value for us.
Immutability Example

x = 300
y = 300

s = 'a string'

t = ('a', 'tuple')
Mutability

• For mutable types, we can actually change the value where it's stored in memory.
Mutability Example

• \( l = [1, 2, 3] \)
Mutable and Immutable Types

• Mutable types:
  – list, dictionary

• Immutable types:
  – Int, float, string, tuple
Quiz Walkthrough

Assume these definitions:

\[ y = 4 \]

\[ \text{def double}(y): \]
\[ \quad \text{return } y + y \]

Evaluate the following expression. Show every step.

\[ y + \text{double}(3 \times y) \]
y = 3
def double(y):
    return y + y
def quadruple1(y):
    return double(double(y))
def quadruple2(y):
    return double(y) + double(y)
def quadruple3(y):
    return y + y + y + y

double(y) + quadruple1(double(y) + y)
y = 3
def double(y):
    return y + y
def quadruple1(y):
    return double(double(y))
def quadruple2(y):
    return double(y) + double(y)
def quadruple3(y):
    return y + y + y + y
double(y) + quadruple2(double(y) + y)
Quiz Walkthrough

```python
y = 3
def double(y):
    return y + y
def quadruple1(y):
    return double(double(y))
def quadruple2(y):
    return double(y) + double(y)
def quadruple3(y):
    return y + y + y + y

double(y) + quadruple3(double(y) + y)
```