Why Learn to Program without an AI?

Claude Sonnet 4

Is it really necessary to learn how to program given that llm chatbots can provide code based on user prompts?

Programming skills remain valuable even with powerful AI code generation for several key reasons:

Understanding and evaluation: To effectively use AI-generated code, you need to understand what it does, whether it's correct, secure, and efficient. Without programming knowledge, you can't properly evaluate or debug the output.

Problem decomposition: Good programming isn't just about writing syntax - it's about breaking down complex problems into manageable pieces, designing system architecture, and making strategic decisions about data structures and algorithms. AI needs clear, well-structured prompts to generate useful code.

Iteration and refinement: Real software development involves continuous refinement, debugging, and optimization. You need to understand code to modify it, integrate it with existing systems, and maintain it over time.

Domain expertise: AI generates generic solutions, but you need programming knowledge to adapt code to your specific requirements, constraints, and existing codebase.

AI limitations: Current AI models can make mistakes, suggest outdated approaches, or miss important edge cases. They also can't understand your full project context or business requirements without detailed explanation.

Week 1

You should be able to:

- Set up a local Python development environment using VSCode.
- Write basic Python programs to use and manipulate the primitive data types (numerical, string, None)
- Apply valid name binding and related conventions to Python objects
- Build expressions and statements using Python operators (Arithmetic, Boolean, Relational, Logical) that accord to operator precedence

Week 1

You should also be able to:

- Distinguish between Python modules, packages, and libraries
- Identify and explain what a Python object is as well as describe object properties
- Discuss how Python allocates memory to object data (immutable types)

1. Python Code: Organization & Structure

- Modules – a file saved with the .py extension.

NOTE: A Python module is really just a text file.

1. Python Code: Organization & Structure

- Modules – a file saved with the .py extension.

Python Code: Organization & Structure

- Modules – a file saved with the .py extension
- Package – group of modules organized using directories

Python’s JSON Package

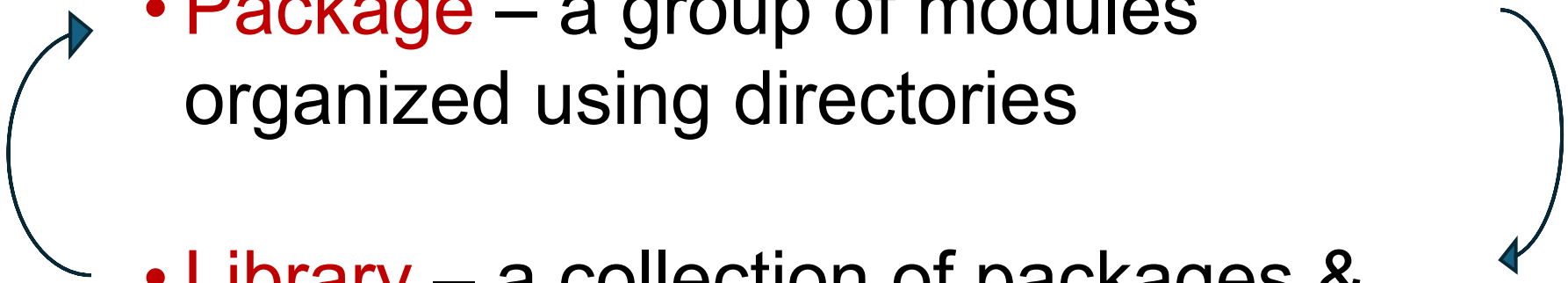
```
json/
├── __init__.py                 # Main module interface
├── decoder.py                  # JSON decoder implementation
│   ├── class JSONDecoder
│   ├── def decode()
│   └── def raw_decode()
├── encoder.py                  # JSON encoder implementation
│   ├── class JSONEncoder
│   ├── def encode()
│   └── def iterencode()
├── scanner.py                  # Low-level JSON scanning
│   ├── def make_scanner()
│   └── def py_make_scanner()
└── tool.py                     # Command-line JSON tool
    ├── def main()
    └── CLI argument parsing
```

Python Code: Organization & Structure

- Modules – a file saved with the .py extension
- Package – a group of modules organized using directories
- Library – a collection of packages & modules that provides functions, classes, and tools (reusable code) to help developers perform specific tasks

Python Code: Organization & Structure

- Modules – a file saved with the .py extension
- Package – a group of modules organized using directories
- Library – a collection of packages & modules



Python Code: Organization & Structure

- Function – a block of organized, resusable code that is used to perform a task

Functions: Topic for Week 2

Python Function

```python
def greet(name):
    return f"Hello, {name}!"

# Usage
message = greet("Alice")
print(message)  # Output: Hello, Alice!
```

Python Code: Organization & Structure

- Function – a block of organized, reusable code that is used to perform a task
- Class – a blueprint for creating objects that encapsulate data and functionality (methods) together

Python Class

```python
class Dog:
    def __init__(self, name):
        self.name = name

    def bark(self):
        return f"{self.name} says woof!"

# Usage
my_dog = Dog("Rex")
print(my_dog.bark())  # Output: Rex says woof!
```

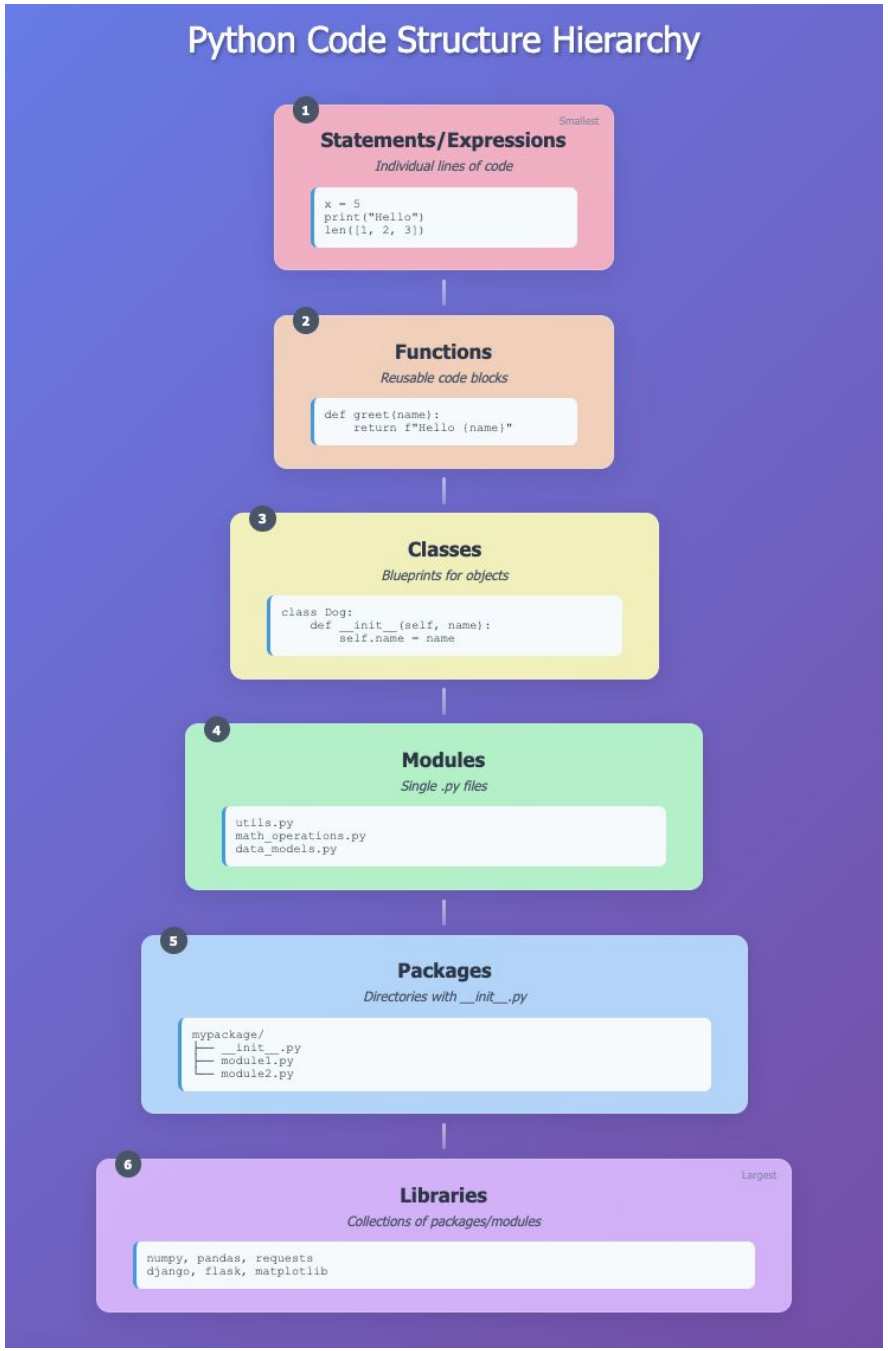
Python Code: Organization & Structure

- Function – a block of organized, reusable code that is used to perform a task
- Class – a blueprint for creating objects that encapsulate data and functionality (methods) together
- Statements/Expressions – individual lines of code

Python Expressions & Statements

```
# EXPRESSIONS (evaluate to values)
5 + 3              # evaluates to 8
"hello".upper() # evaluates to "HELLO"
len([1, 2, 3])  # evaluates to 3

# STATEMENTS (perform actions, no return value)
x = 5              # assignment statement
print("hello")  # function call statement
if x > 0:          # conditional statement
    pass
```


Python Code Structure Hierarchy
1
Smallest
Statements/Expressions
Individual lines of code
x = 5
print("Hello")
len([1, 2, 3])
2
Functions
Reusable code blocks
def greet(name):
return f"Hello {name}"
3
Classes
Blueprints for objects
class Dog:
def __init__(self, name):
self.name = name
4
Modules
Single .py files
utils.py
math_operations.py
data_models.py
5
Packages
Directories with __init__.py
mypackage/
├── __init__.py
├── module1.py
└── module2.py
6
Largest
Libraries
Collections of packages/modules
numpy, pandas, requests
django, flask, matplotlib

Everything in Python is an Object

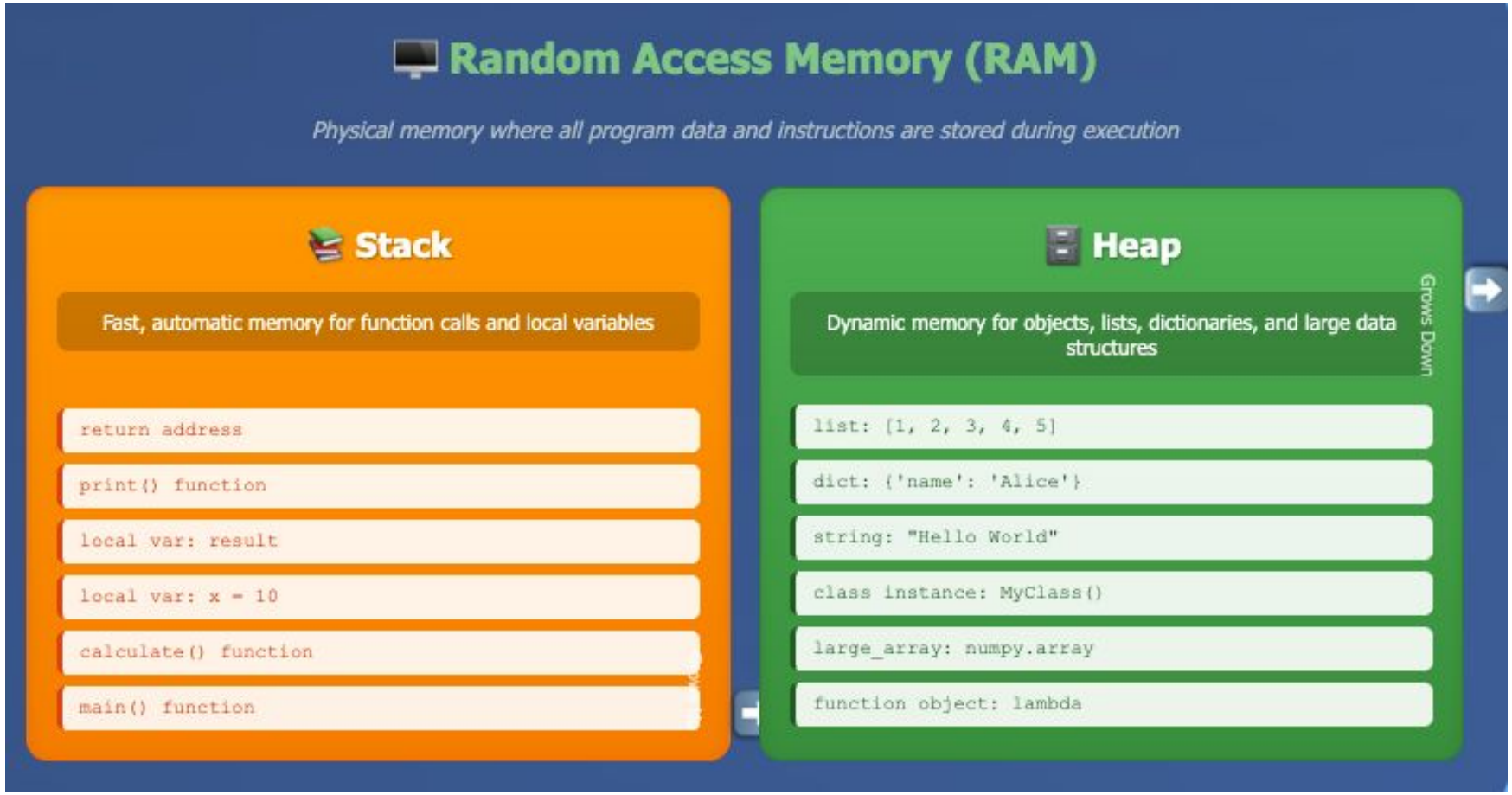
Python Objects: have identity, type, value

"First class objects"

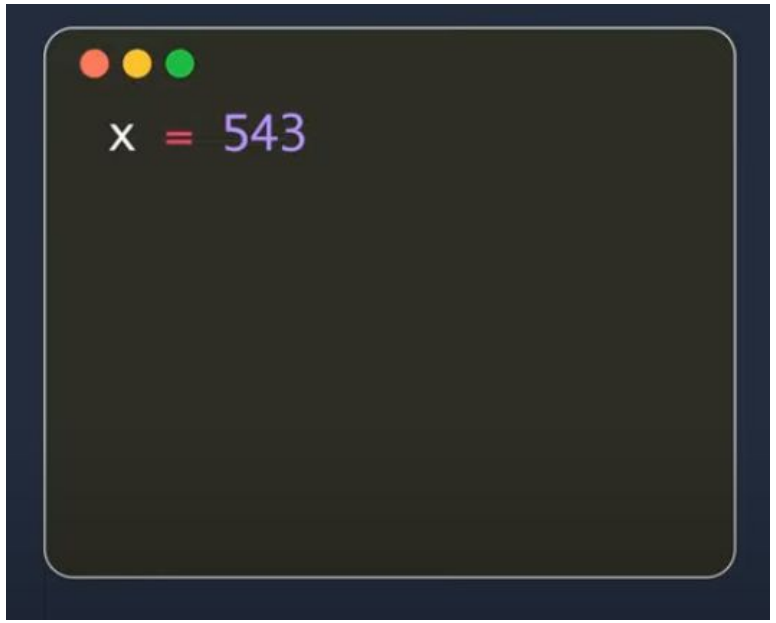
- Can be passed as arguments
- Can be returned from functions
- Can be assigned to variables
- Can be stored in data structures
- Can be created and manipulated at runtime

Example

Python Memory Management

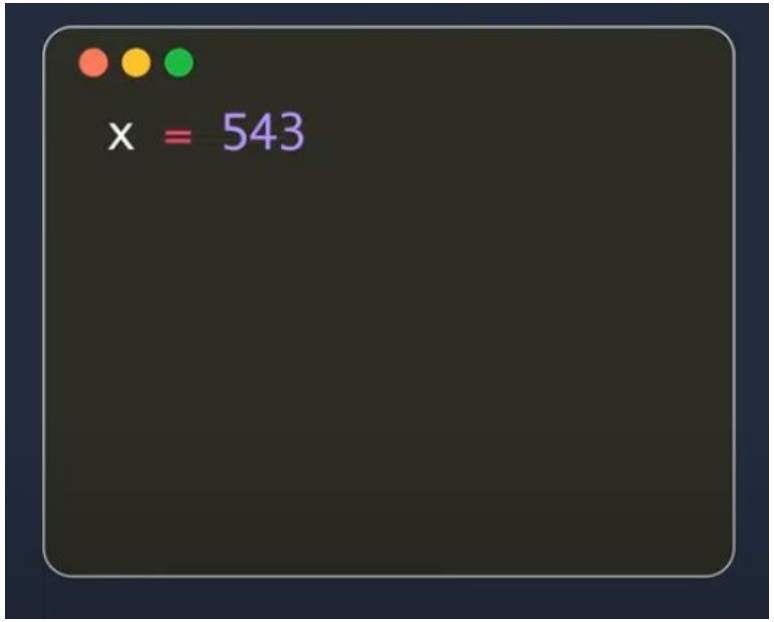


Name Binding & Memory



https://www.youtube.com/watch?v=Bz3ir-vKqkk

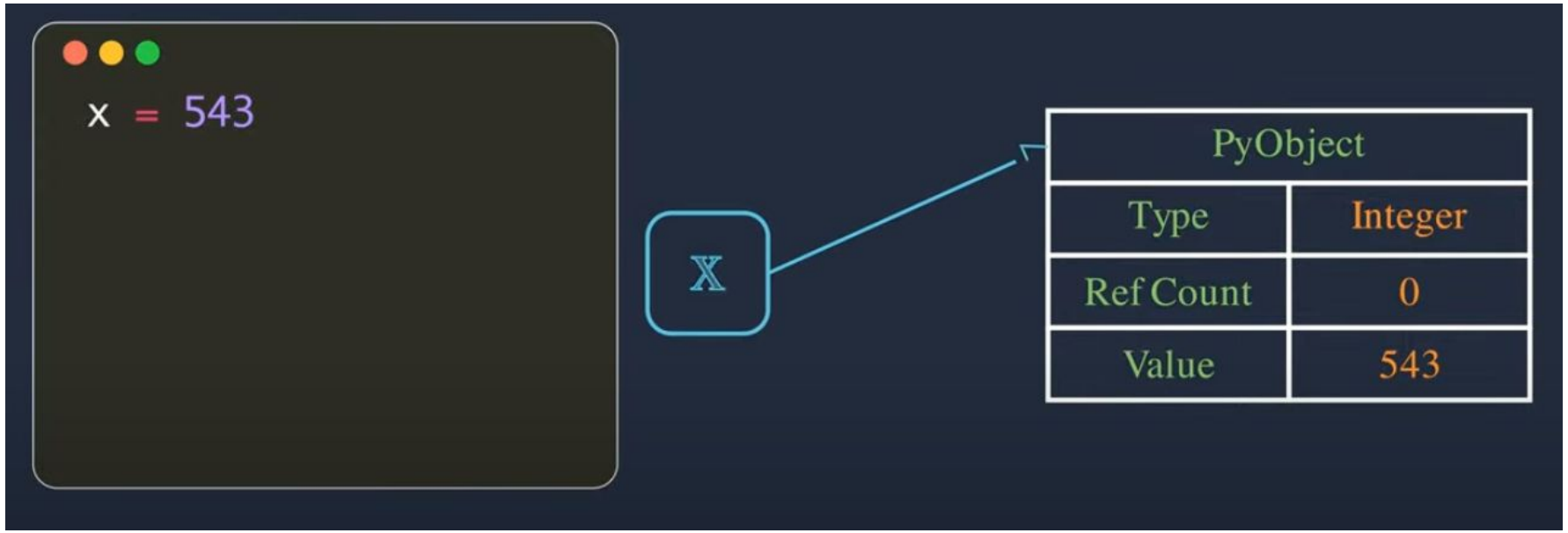
Name Binding & Memory



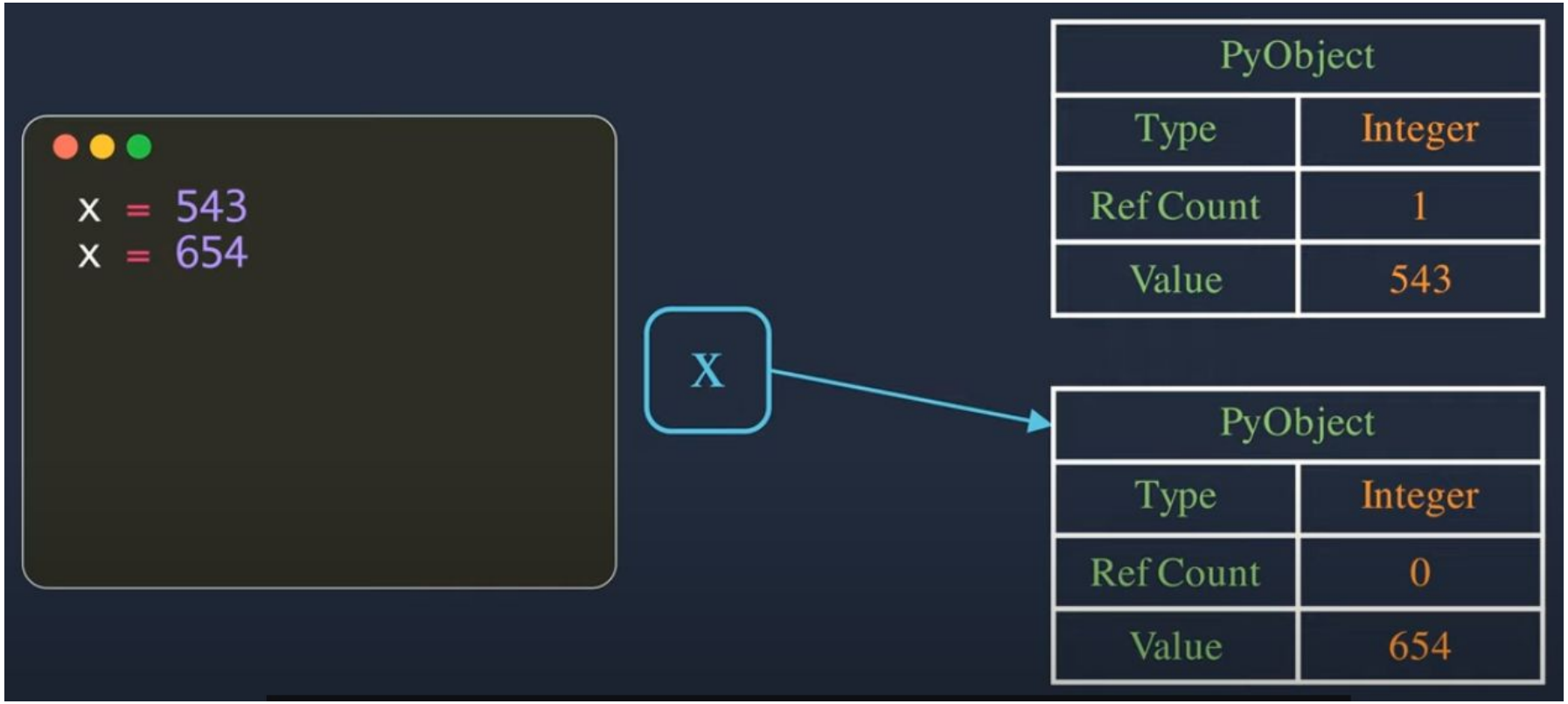
The Stack holds the "names," the Heap holds the actual "things."

https://www.youtube.com/watch?v=Bz3ir-vKqkk

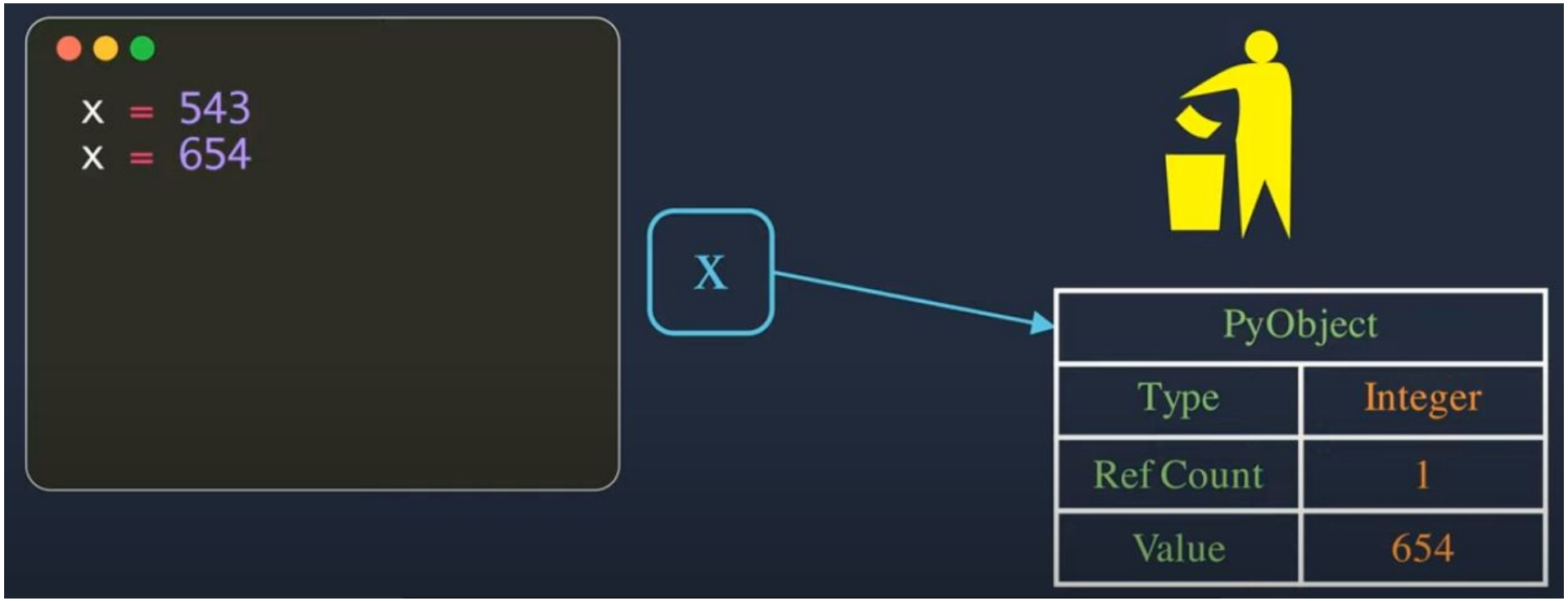
Id = 100073723106504



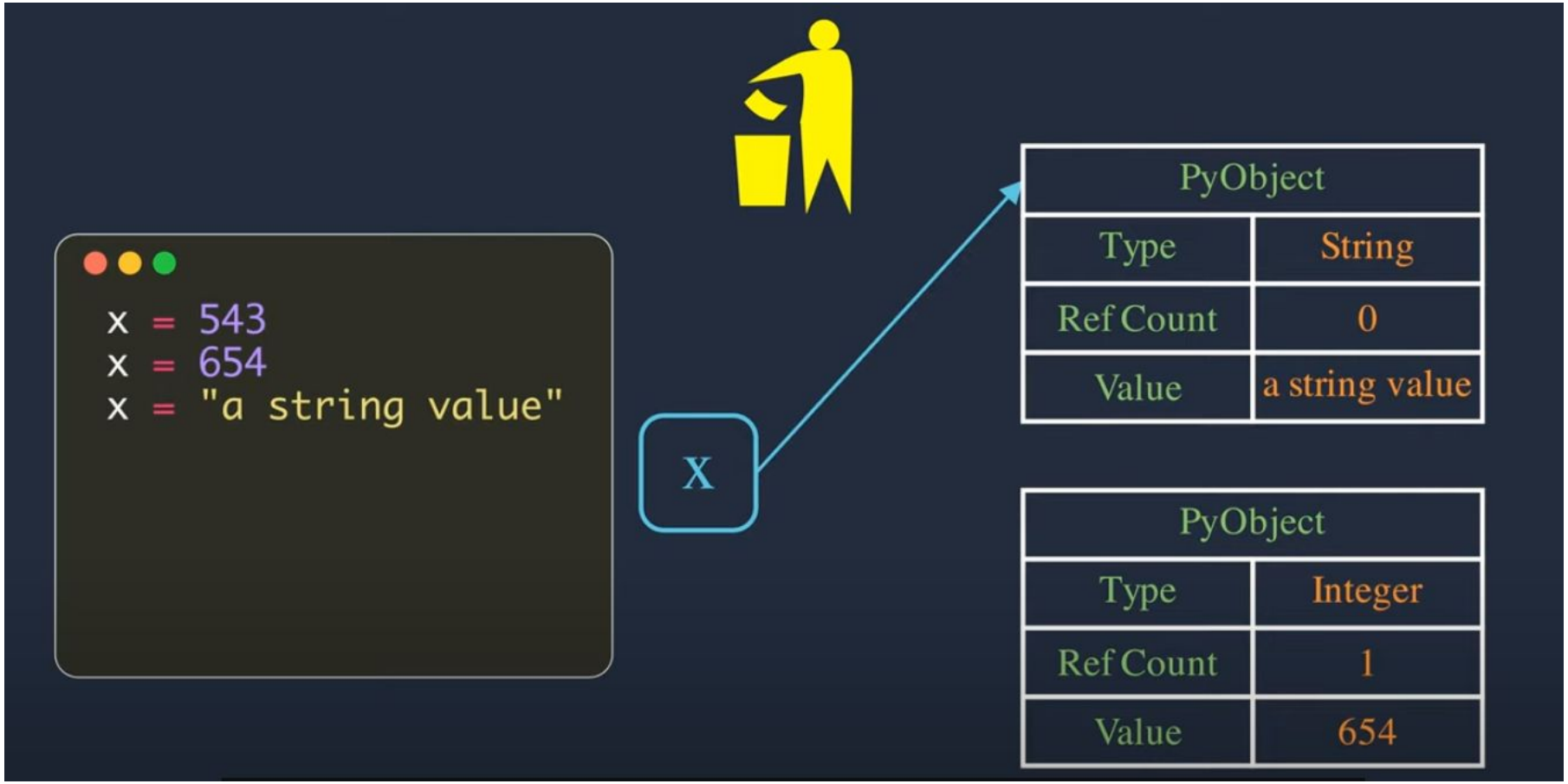
https://www.youtube.com/watch?v=Bz3ir-vKqkk



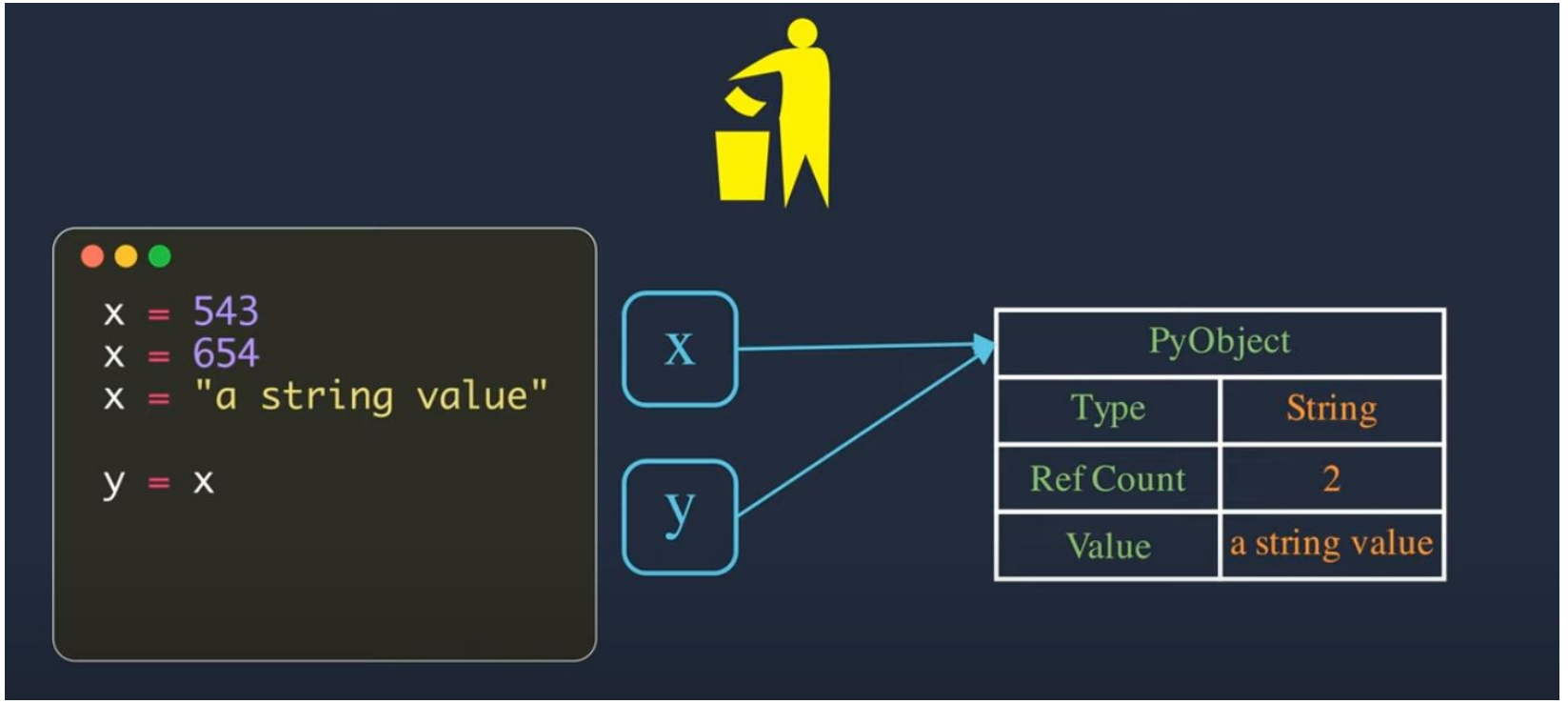
https://www.youtube.com/watch?v=Bz3ir-vKqkk



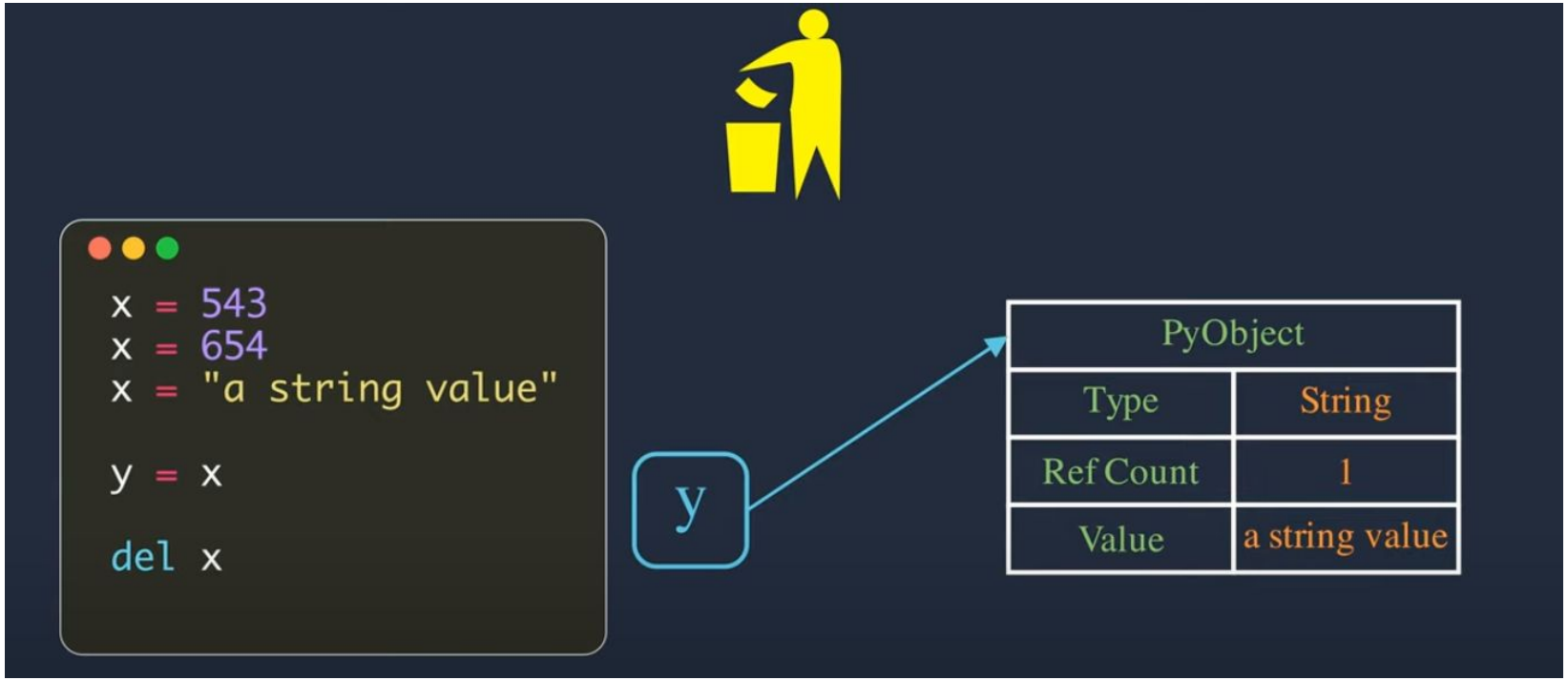
https://www.youtube.com/watch?v=Bz3ir-vKqkk



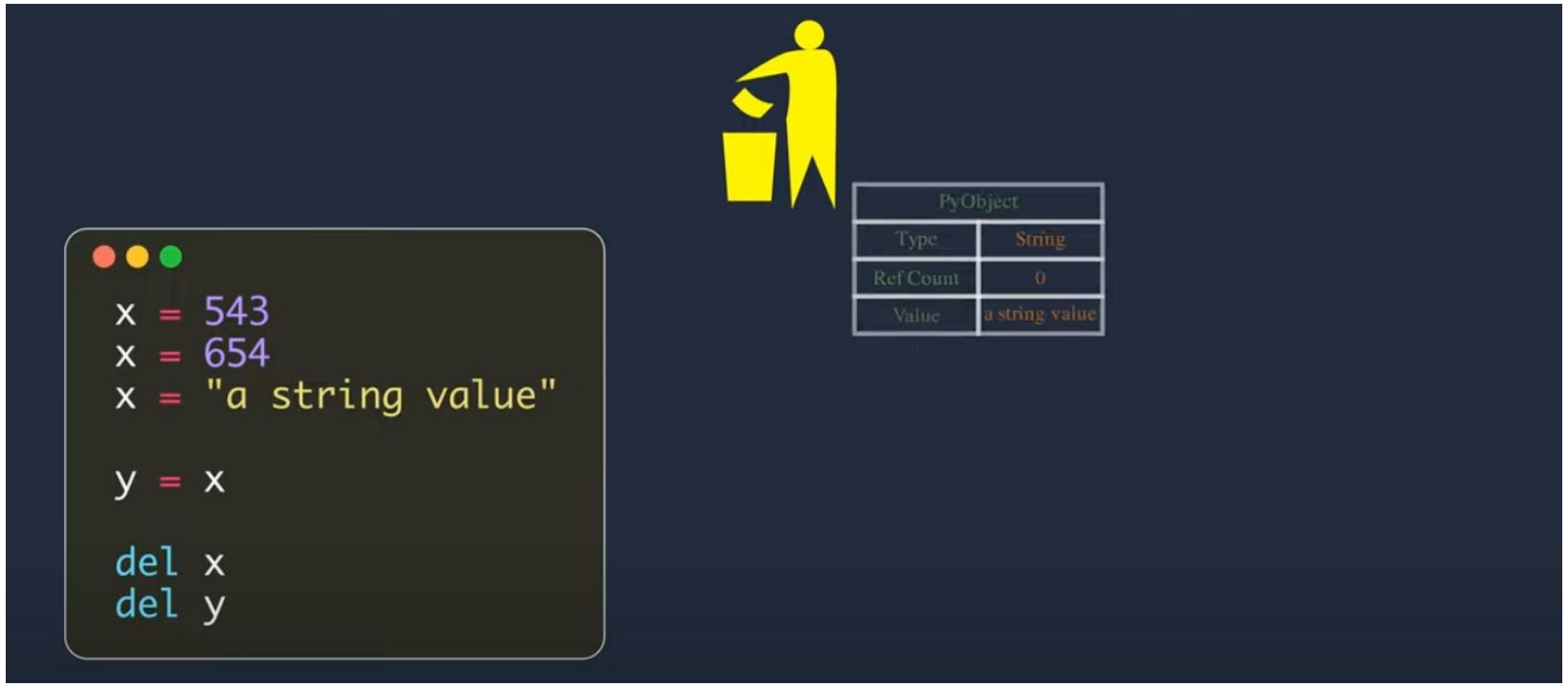
https://www.youtube.com/watch?v=Bz3ir-vKqkk



https://www.youtube.com/watch?v=Bz3ir-vKqkk



https://www.youtube.com/watch?v=Bz3ir-vKqkk



https://www.youtube.com/watch?v=Bz3ir-vKqkk