Classes and Pointers

Classes

- **Classes** are like **cookie cutters**—they define a blueprint for creating objects.
- **A constructor** (`__init__` method) defines what happens when we create an instance of a class.
- **Methods vs. Functions:**
  - If it has `self`, it's a **method**, not a regular function!
  - `self` refers to the instance of the class.
  - `self` is **Python-specific**—other languages use different ways to reference instance attributes.
- We can define additional **methods** in a class beyond the constructor.

Pointers

- **Integers are immutable!**
  - If you assign a number to a variable (`var1 = 10`), then create another variable and point it to `var1` (`var2 = var1`), **both point to the same memory location**.
  - However, if you change `var2` (`var2 = 20`), it will **allocate new memory**, while `var1` remains unchanged.
- **Dictionaries (and other mutable types) behave differently!**
  - If you create a dictionary (`dict1 = {}`) and then assign another variable to it (`dict2 = dict1`), both variables **point to the same memory location**.
  - **Modifying** `dict2` **will also modify** `dict1` because they reference the same object in memory.
  - If you assign a new dictionary to `dict2` (`dict2 = {}`), it now **points to a new object**, leaving `dict1` unchanged.