

`__eq__` compares addresses of two items by default. Overrides `==`.

`__getitem__` gets the item at a specific key `__init__` is Python's constructor, first param is always passed in and is usually called "self" `__iter__` returns the iterable for a Python object `__next__` pops the next item off of an iterable `assert` is good for pre and post-conditions, bad for user input or testing `unittest` is good for testing use `try` and `except` with `StopIteration` exception to build a custom iterator Mutable: List, Set Immutable: Tuple, FrozenSet `floordiv()` is

like Java's integer division `def` declares a function `issubclass` with parameter `classname` checks if a class is a subclass of another class `truediv()` is like Java's double division `pass` is used when you have no code to put inside an `if/else/try/except` Use a comma at the end of a single item tuple `Yield` is used as a return inside generators `Map` applies a function to all items inside of an iterable. The first parameter is the function to be applied, the second is the list that is iterable. `Reduce` takes a two param function and a list, and goes iterating through with

a "cur" and a "next" as the first and second parameter being passed into the function. For `reduce`, the default is either the first value or the third parameter Positional arguments have to go before named arguments For parameters, you basically replace `*` with all the values as positional arguments, and replace `**` (dict values) with named arguments Passing in the same parameter twice into a function causes an error Using `*` then `**` in the function declaration can let the user pass anything into a function