Sets

Creating new sets:
- `set()` creates an empty set
- `{element1, element, ...}` creates a set containing the given elements
- `set(iterable)` creates a set from any iterable object (e.g., range, list, string)

Built-in set functions:
- `len(list)`: Returns the number of elements in the list

Set object methods:
- `add(elt)`: Adds `elt` to the set
- `clear()`: Removes all elements from the set
- `pop()`: Removes an arbitrary element from the set and returns it
- `remove(elt)`: Removes `elt` from the set

Set operators:
- `elt in set`: Returns `True` if `elt` is an element of `set`, `False` otherwise
- `set1 <= set2`: Returns `True` if `set1` is a subset of `set2` (every element of `set1` is in `set2`), `False` otherwise
- `set1 | set2`: Returns union of the two sets (new set with elements from both sets)
- `set1 & set2`: Returns intersection of the two sets (new set with only elements common to both sets)
- `set1 - set2`: Returns set difference (new set with elements from `set1` not in `set2`)

Dictionaries

Creating new dictionaries:
- `{}` creates an empty dictionary
- `{key1:value1, key2:value2, ...}` creates a new dictionary with key-value pairs

Built-in dictionary functions:
- `len(dict)`: Returns the number of entries (key-value pairs) in the dictionary

Dictionary object methods:
- `clear()`: Removes all entries from the dictionary
- `keys()`: Returns an iterable object of the keys in the dictionary
- `values()`: Returns an iterable object of the values in the dictionary
- `items()`: Returns an iterable object of all (key, value) tuples in the dictionary
- `get(key[, item])`: Returns value associated with `key` if in dictionary, `item` otherwise. `item` defaults to `None`

Dictionary operators:
- `item in dict`: Returns `True` if `item` is in the keys of `dict`, `False` otherwise

Tuples

Creating new tuples:
- `()` creates an empty tuple
- `(object1, object2, ...)` creates a new tuple containing objects

Built-in tuple functions:
- `len(tuple)`: Returns the number of elements in the tuple