**Data Structures**

Also known as objects, data structures are collections of values and functions.

Tuples, lists, dictionaries, sets.

Unlike strings, these data structures can contain any kind of data.

The primary two we’ll focus on are lists and dictionaries.
Lists

[ ]

A list is a mutable sequence of 0 or more values

Enclosed in square brackets, items separated by a comma

List elements can be added, removed, or modified

Very common and useful

Use indexing and slicing to access individual elements
Lists

Functions

- `s.append()`
- `s.count(x)`
- `s.extend(list)`
- `s.index(x)`
- `s.insert(i, x)`
- `s.pop(i)`
- `s.remove(x)`
- `s.reverse()`
- `s.sort()`
Dictionaries

{}  

A dictionary is a data structure for storing pairs of values

Values can be accessed by their keys

Like lists, dictionaries are mutable

Keys are unique and cannot be repeated within a dictionary

Keys must be immutable and cannot be a list or dictionary

Values are, however, mutable
Dictionaries

Functions

d.items()
d.keys()
d.values()
d.get(key)
d.pop(key)
d.popitem()
d.clear()
d.copy()
d.fromkeys(s, t)
Tuples

(1, 2.0, 'three')

A tuple is an immutable sequence of 0 or more values

Enclosed in round brackets, items separated by a comma

Tuples with a single item must be followed by a comma: (x,)

Once created, it cannot be changed, which can help prevent errors

Use indexing and slicing to access individual elements
**Tuples**

**Functions**

- `x in tuple`
- `len(tuple)`
- `tuple.count(x)`
- `tuple.index(x)`
Sets

A set is a collection of 0 or more items with no duplicates.

A good way to remove duplicates from a sequence.

Two categories: mutable sets and immutable frozensets.