Recall Week 7 - Sets & Strings

You should be able to:

- Discuss the key characteristics of sets & strings & hashability
- Index and slice strings
- Apply Python methods associated with sets and strings
- Perform standard loops on sets and strings
- Assess membership and perform operations on lists and tuples

Week 8 - Dictionaries

You should be able to:

- Discuss the key characteristics of dictionaries
- Manipulate dictionary keys & values
- Apply Python methods to dictionaries
- Perform standard loops on dictionaries
- Assess membership and perform operations on dictionaries

Sets

{ }

- Mutable
- Unordered - cannot be indexed and sliced
- Unique - no duplicate items
- Immutable items (hashable - int, float, str, tuple)
- Iterable - can loop over
- 17 associated built-in methods

Dictionaries
{k:v}

- Mutable - can change key:value pairs
- Ordered by insertion order but access by keys
- Keys are unique - no duplicate keys
- Keys are immutable (hashable - int, float, str, tuple)
- Iterable - can loop over
- 11 associated dot methods

Dictionaries

keys

```
brew_methods = {
        "pour_over": "medium-fine grind",
        "french_press": "coarse grind",
        "espresso": "fine grind"
}
```

Values

JSON

Javascript Object Notation

A lightweight, string-based data format used for storing and exchanging data

```
json_text = '{"name": "Alice", "age": 30, "city": "Denver"}'
```

JSON & Dictionaries

Feature	JSON	Python Dictionary
Type	Text (string format)	Python object (`dict`)
Purpose	Data exchange between systems	Data storage & manipulation in Python
Syntax Origin	JavaScript	Python
Example	`'{"name": "Alice", "age": 30}'`	`{"name": "Alice", "age": 30}`
Keys	Must be **strings** in double quotes	Must be **immutable** (usually strings) — single or double quotes OK
Values	Must be valid JSON types (string, number, object, array, `true` , `false` , `null`)	Can be any Python object (`str` , `int` , `float` , `list` , `dict` , `bool` , `None` , etc.)
Booleans	Lowercase: `true` , `false`	Capitalized: `True` , `False`
Null value	`null`	`None`
Comments	❌ Not allowed	✅ Allowed in code
Trailing commas	❌ Not allowed	✅ Allowed

JSON & Dictionaries

JSON → text format for transmitting data

Dictionary → a data structure for in memory computation

Python's json module allows us to convert between them in memory using json.dumps() and json.loads()

Direction	Function	Example
Python → JSON	`json.dumps()`	`json.dumps({"a": 1}) → '{"a": 1}'`
JSON → Python	`json.loads()`	`json.loads('{"a": 1}') → {'a': 1}`

JSON & Dictionaries

When reading or writing from or to a json file object, respectively, use:

json.load() → read json from a file

json.dump() → writes directly to file