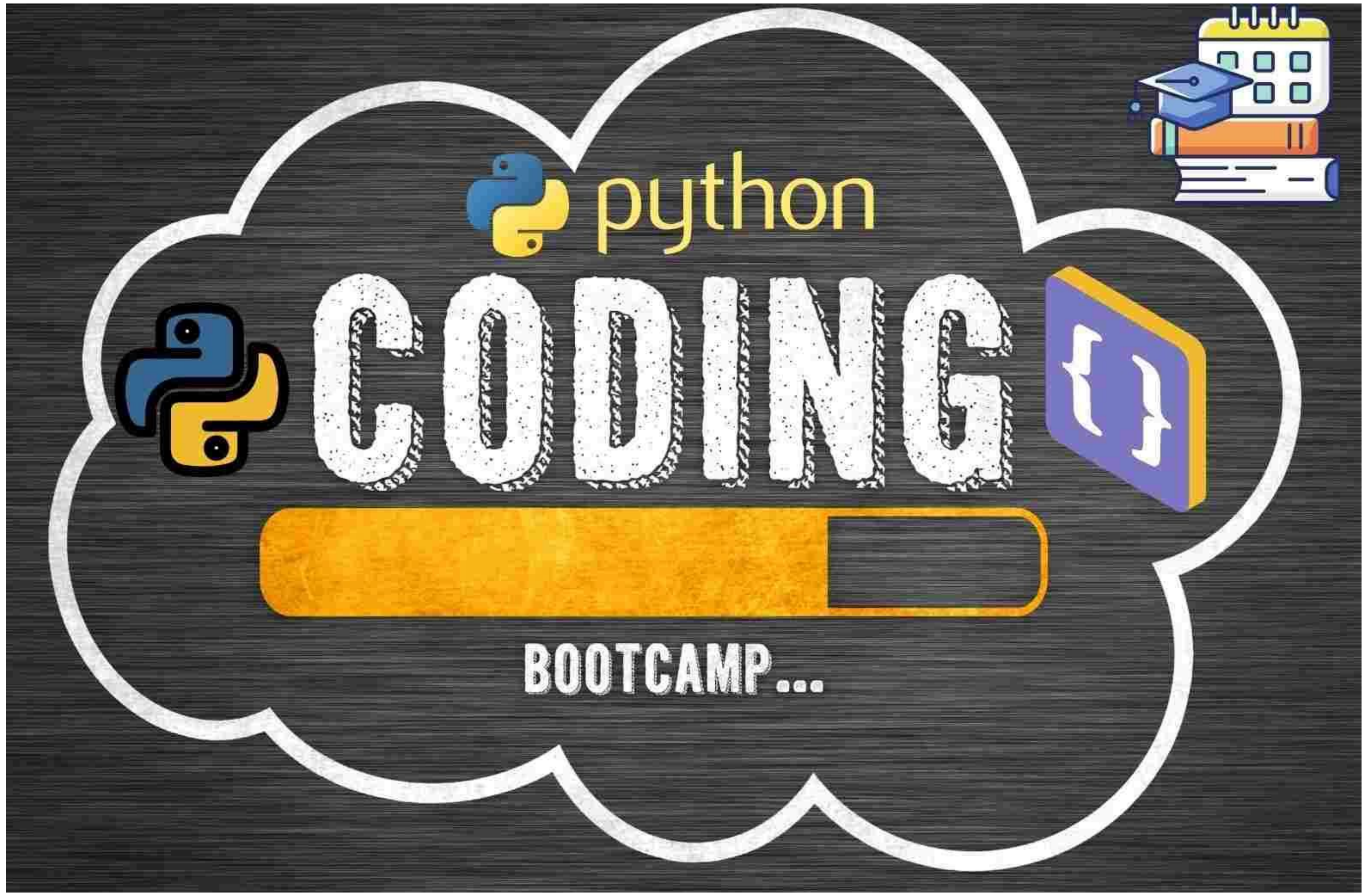

Python Bootcamp 3 Part 2

Dictionaries and Files

Vincent Y. Zhuang

Spring 2025

Dictionary

a collection of {key: value} pairs

dictionaries are indexed by keys (strings and numbers)

Compound data type - allow us to work with multiple items at once

Values in a dictionary can be of any data type and can be duplicated, whereas keys can't be repeated and must be *immutable (string, number)*.

Dictionary syntax

```
age_dict = {"Jo": 60, "Rae": 68, "Tom": 65}
```

curly brackets

common shorthand
for dictionary

Dictionary syntax

```
age_dict = {"Jo": 60, "Rae": 68, "Tom": 65}
```

key colon **value**

```
age_dict = {"Jo": 60,
            "Rae": 68,
            "Tom": 65}
```

Can be written like this for clarity.

Dictionary syntax

A dictionary can also be created by the built-in function `dict()`.

```
Dict = dict({1: ‘PolyU', 2: ‘For’, 3: ‘CityU'})
```

Nested Dictionaries

```
Dict = {1: ‘PolyU’, 2: ‘AF’,
        3: {'A’: ‘Welcome’, ‘B’: ‘To’, ‘C’: ‘AF3214'}}
```

Nested keys

Dictionary examples

Item 1

```
age_dict = {"Jo": 60, "Rae": 68, "Tom": 65}
```

key value

```
person = {"height": 65, "eyes": "hazel",
          "hair": "gray", "age": 75}
```

Put `{key:value}` pairs together inside curly brackets and separated by comma

Dictionary examples

A dictionary with values as a list

```
results = {"test1": [3.4, 0.2, 1.4, 2.2, 8.0],
           "test2": [0.9, 3.4, 2.5, 4.7, 2.6],
           "test3": [4.9, 2.4, 0.4, 8.4, 2.5]}
```

Creating dictionary which contains lists.

To access the items in the list: `dictionary_name[key][index]`

e.g., `results[“test1”][3]` **or**

`results.get(“test1”)[3]` for extracting `2.2`

Dictionary Methods

A list of in-built dictionary functions with their description.

Method	Description
`dict.clear()`	Remove all the elements from the dictionary
`dict.copy()`	Returns a copy of the dictionary
`dict.get(key, default = “None”)`	Returns the value of specified key
`dict.items()`	Returns a list containing a tuple for each key value pair
`dict.keys()`	Returns a list containing dictionary’s keys
`dict.update(dict2)`	Updates dictionary with specified key-value pairs
`dict.values()`	Returns a list of all the values of dictionary
`pop()`	Remove the element with specified key
`popItem()`	Removes the last inserted key-value pair
`dict.setdefault(key,default= “None”)`	Set the key to the default value if the key is not specified in the dictionary
`dict.has_key(key)`	Returns true if the dictionary contains the specified key.
`dict.get(key, default = “None”)`	Used to get the value specified for the passed key.

Working with files

Opening files

Original (rarely use)

```
f = open("my_file.txt", "r")
do something with file
f.close()
```

This leaves the file needlessly open, which takes up memory

Opening files

Original (rarely use)

```
f = open("my_file.txt", "r")
do something with file
f.close()
```

This leaves the file needlessly open, which takes up memory

changes in files do not go into effect
until the file is properly closed

'with/as' statement (almost always use)

```
with open("my_file.txt", "r") as f:
     save file as something else
     or save part of file
```

File automatically closes when you exit the indentation

Opening files syntax

```
with open("my_file.txt", "r") as f:
    save file as something else
    or save part of file
```

File automatically closes when you exit the indentation

Opening files syntax

```
with open("my_file.txt", "r") as f:
    save file as something else
    or save part of file
```

temporary variable,
just like in a for-loop

Opening files

The open() function requires two arguments:

```
open(filename, mode)
```

what you're going to do with the file

Opening files

The open() function requires two arguments:

```
open(filename, mode)
```

mode options:

`"r"`	read
`"w"`	write (wipes the file clean if it already exists) ⚠️
`"a"`	append (add to the end of whatever is already in the file)

Opening files

If you are accessing a file in your current working directory, you can just include the filename, but if the file is in a different directory, you must include the file path.

Let's code!