

BEFORE WE GET  
STARTED PLEASE  
DOWNLOAD THE PYTHON  
FILES FOR TODAY UNDER  
WEEK 09



## OVERVIEW OF LAB THIS WEEK

### Tuesday

Introduction to dictionaries

### Labs

Hands-on review of  
dictionaries

### Thursday

Additional practice with  
dictionaries and  
troubleshooting



## LECTURE OVERVIEW

- ✕ Introduction to Dictionaries
- ✕ Creating, Accessing, Adding and Replacing Values in a Dictionary
- ✕ Working with a Dictionary using loops and dictionary methods



# INTRODUCTION TO PYTHON DICTIONARIES





A vast field of red poppies in full bloom, with green stems and leaves visible. The flowers are densely packed, creating a sea of red. In the center, a white-bordered box with a slightly distressed, hand-drawn appearance contains the text "BUT FIRST." in a white, handwritten-style font.

BUT FIRST.



# LET'S REFRESH

Discuss with a neighbor the benefits and cons of using a list.

Tip: Feel free to use the internet

## Lists

- ✗ Contains only values (of any type)
- ✗ An ordered collection
- ✗ Elements are accessed via an index value

## Dictionaries

- ✗ Accessing items within a dictionary is faster, but uses slightly more memory
- ✗ Requires you to use a unique key





# LISTS TO DICTIONARIES



## LET'S COMPARE

### LIST

```
carType = [  
    "Ford",  
    "Taurus",  
    2019  
]
```

### DICTIONARY

```
carType = {  
    "make": "Ford",  
    "model": "Taurus",  
    "year": 2019  
}
```

DICTIONARIES ARE COMPOSED OF KEY/VALUE PAIRS

Dictionary  
Declaration

Usage of  
a colon      Usage of  
curly braces

carType = {"make": "Ford"}

Key

Value

# OH THE MANY POSSIBILITIES OF THE DICTIONARY VALUE

String

List

Float

Integer

Dictionary

...or any data type

## LET'S RE-EXAMINE

```
carType = {  
  "make": "Ford",  
  "model": "Taurus"  
  "year": 2019  
}
```



Image of a Ford Taurus





You can think of the structure of a dictionary as basically a table with two columns and unlimited rows

## LET'S CREATE AN ADDRESS BOOK

```
person = {  
  "firstname": "Janet",  
  "lastname": "Jackson",  
  "nickname": "Ms.Jackson",  
  "occupation": "singer"  
}
```



Image of Janet Jackson



# ACCESSING ITEMS

## Usage of Brackets



```
getPerson = person["firstname"]
```

Method 1



## Usage of Parenthesis



```
getPerson = person.get("firstname")
```

Method 2 (Built-in Method)

2.

ADDING ITEMS

Dictionary  
Declaration



Usage of  
Bracket



person["location"] = "Las Vegas"



Key



Value



# CHANGING VALUES



## Usage of Bracket



```
person["nickname"] = "Ms. Janet"
```



# ITERATING THROUGH A DICTIONARY

```
for content in person:  
    print(person[content])
```

Method 1

```
for key, value in person.items():  
    print(key, value)
```

Method 2

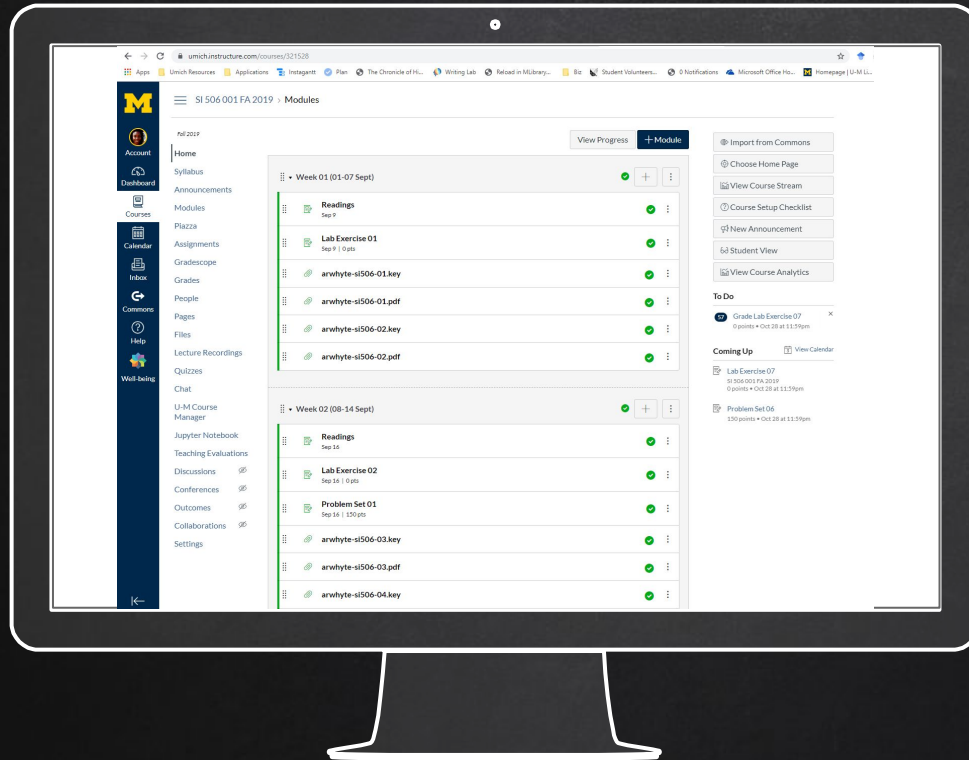


5.

# DICTIONARY METHODS

SOURCE: [WWW.W3SCHOOLS.COM/](http://WWW.W3SCHOOLS.COM/)

Method	Description
<u><a href="#">clear()</a></u>	Removes all the elements from the dictionary
<u><a href="#">copy()</a></u>	Returns a copy of the dictionary
<u><a href="#">fromkeys()</a></u>	Returns a dictionary with the specified keys and values
<u><a href="#">get()</a></u>	Returns the value of the specified key
<u><a href="#">items()</a></u>	Returns a list containing the a tuple for each key value pair
<u><a href="#">keys()</a></u>	Returns a list containing the dictionary's keys
<u><a href="#">pop()</a></u>	Removes the element with the specified key
<u><a href="#">popitem()</a></u>	Removes the last inserted key-value pair
<u><a href="#">setdefault()</a></u>	Returns the value of the specified key. If the key does not exist: insert the key, with the specified value
<u><a href="#">update()</a></u>	Updates the dictionary with the specified key-value pairs
<u><a href="#">values()</a></u>	Returns a list of all the values in the dictionary



LET'S EXPLORE WHAT  
WE LEARNED!



THANKS!

Presentation Created By:

Ihudiya Finda Ogburu

@iogburu

[iogburu@umich.edu](mailto:iogburu@umich.edu)