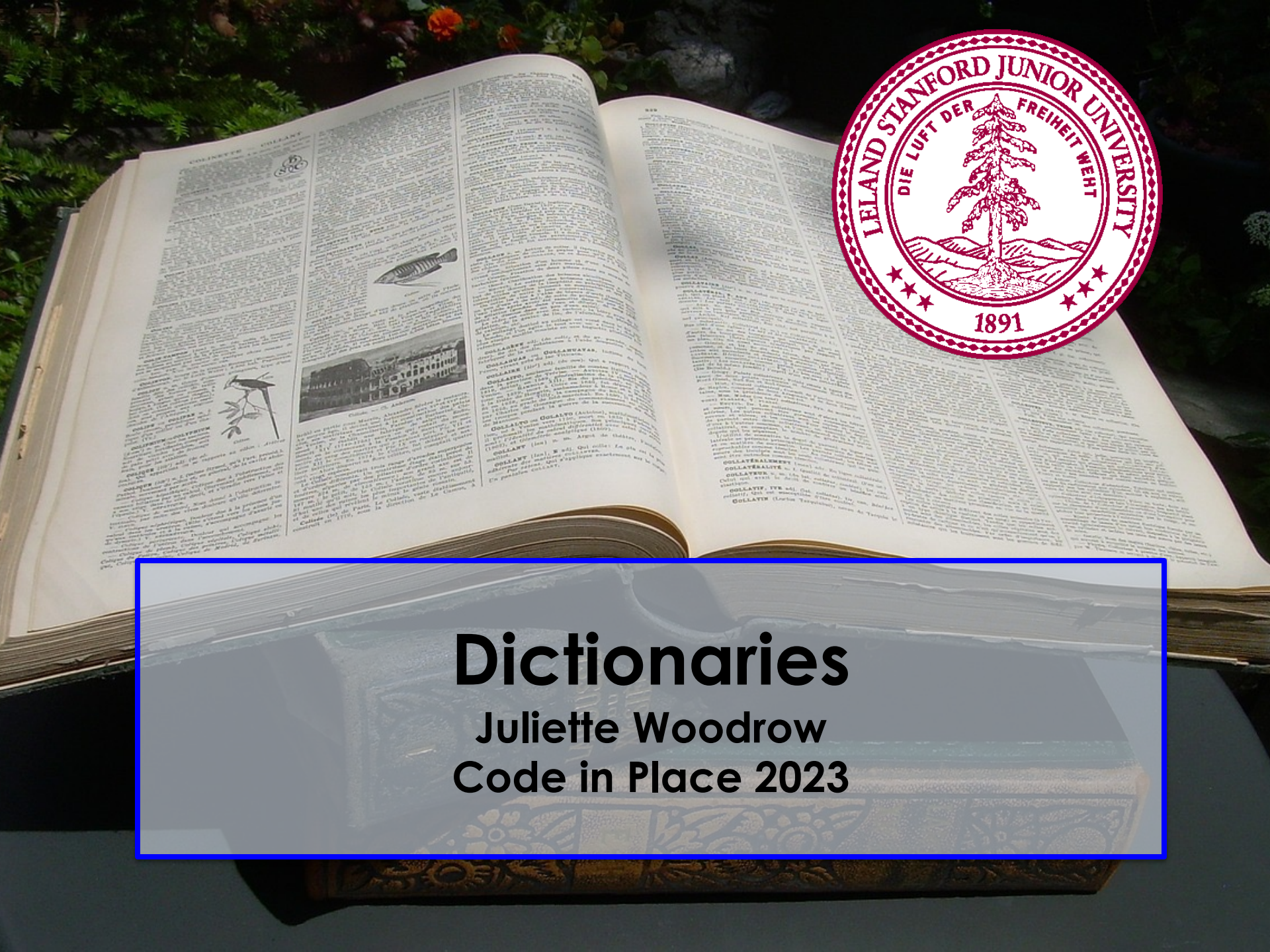


Dictionaries

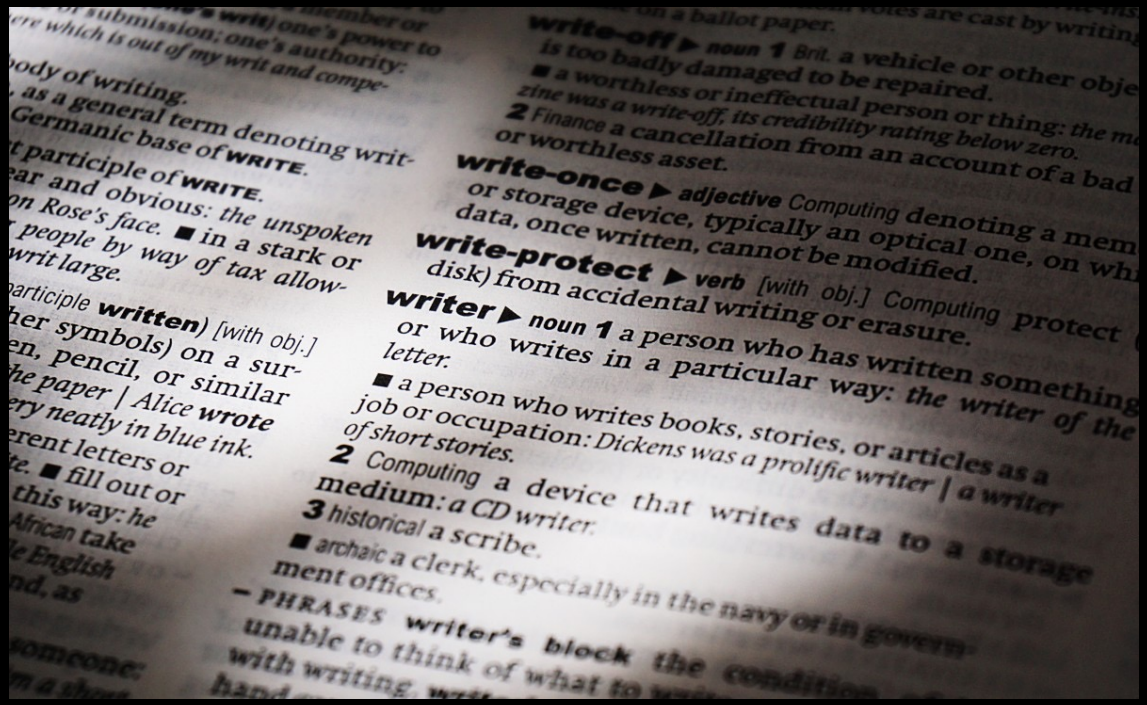
Juliette Woodrow

Code in Place 2023

Learning Goals

1. Learning about dictionaries
2. Building programs using dictionaries

Dictionaries



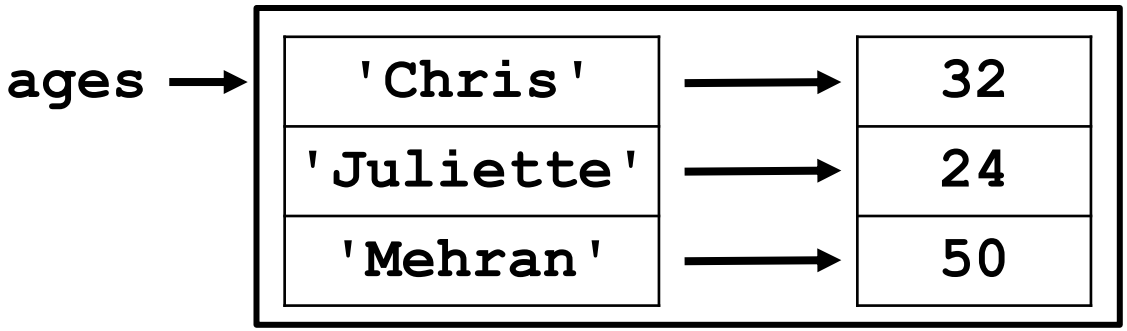
What are Dictionaries?

- Dictionaries associate a ***key*** with a ***value***
  - Key is a *unique* identifier
  - Value is something we associate with that key
- Examples in the real world:
  - Phonebook
    - Keys: names
    - Values: phone numbers
  - Dictionary
    - Keys: words
    - Values: word definitions
  - Social Media Platforms
    - Keys: usernames
    - Values: Information like account details, posts, and followers

Dictionaries in Python

- Creating dictionaries
  - Dictionary start/end with braces
  - Key:Value pairs separated by colon
  - Each pair is separated by a comma

```
ages = {'Chris': 32, 'Juliette': 24, 'Mehran': 50}
squares = {2: 4, 3: 9, 4: 16, 5: 25}
phone = {'Pat': '555-1212', 'Jenny': '867-5309'}
empty_dict = {}
```

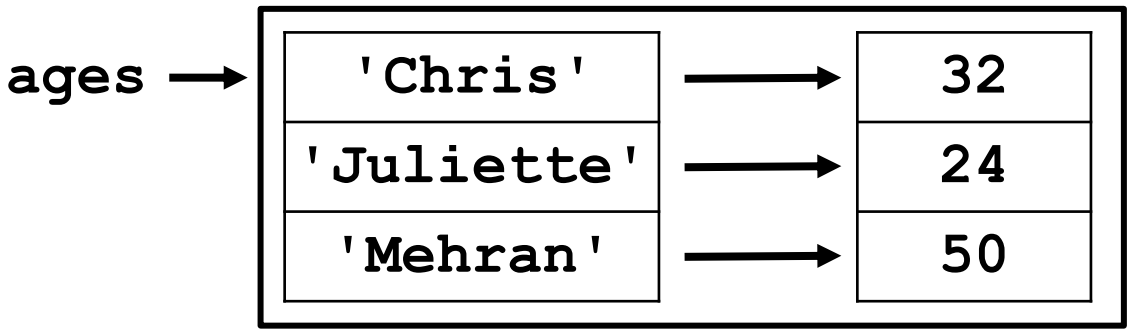


Accessing Elements of Dictionary

- Consider the following dictionary:

```
ages = {'Chris': 32, 'Juliette': 24, 'Mehran': 50}
```

- Like a set of variables that are indexed by keys



- Use key to access associated value:

`ages['Chris']` is `32`

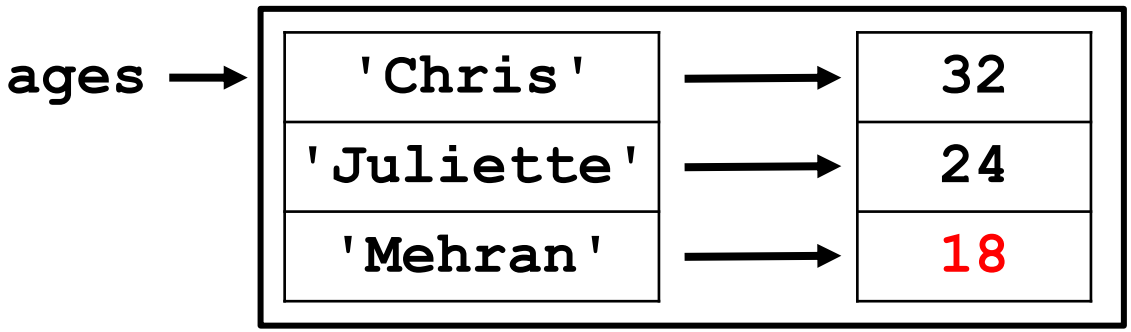
`ages['Mehran']` is `50`

Accessing Elements of Dictionary

- Consider the following dictionary:

```
ages = {'Chris': 32, 'Juliette': 24, 'Mehran': 50}
```

- Like a set of variables that are indexed by keys



- Use key to access associated value:

```
ages['Chris'] is 32
ages['Mehran'] is 50
```

- Can set values like regular variable:

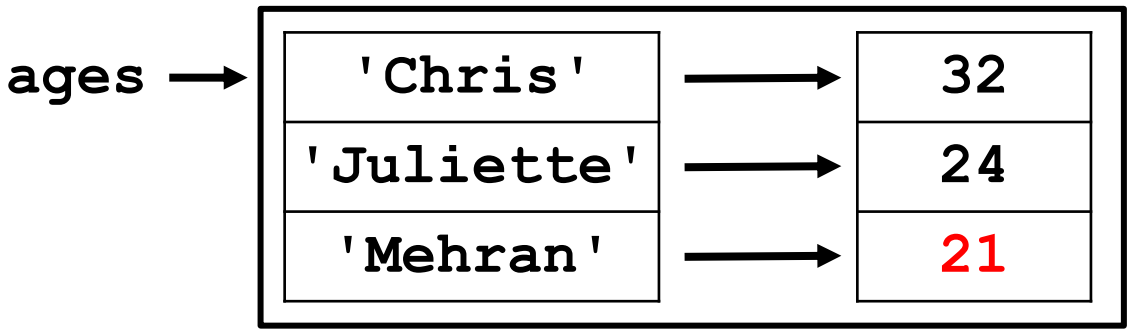
```
ages['Mehran'] = 18
```

Accessing Elements of Dictionary

- Consider the following dictionary:

```
ages = {'Chris': 32, 'Juliette': 24, 'Mehran': 50}
```

- Like a set of variables that are indexed by keys



- Use key to access associated value:

```
ages['Chris'] is 32
ages['Mehran'] is 50
```

- Can set values like regular variable:

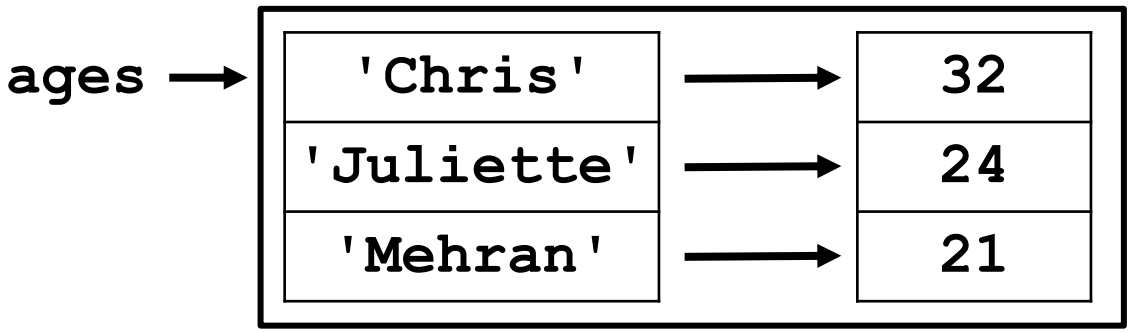
```
ages['Mehran'] = 18
ages['Mehran'] += 3
```

Accessing Elements of Dictionary

- Consider the following dictionary:

```
ages = {'Chris': 32, 'Juliette': 24, 'Mehran': 50}
```

- Like a set of variables that are indexed by keys



- Good and bad times with accessing pairs:

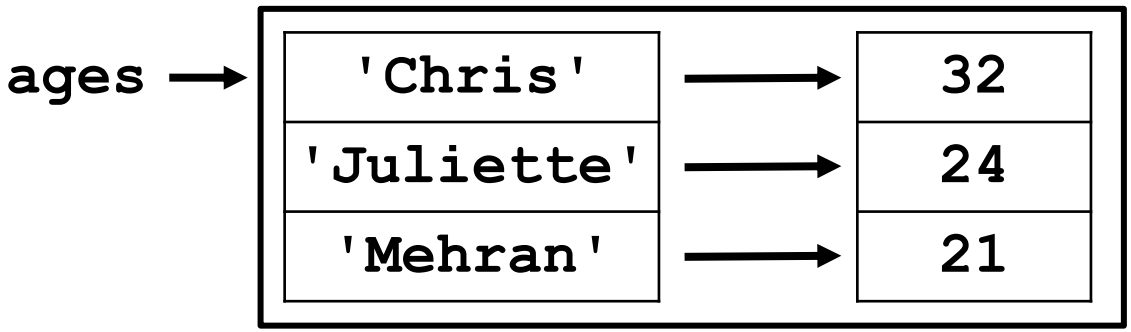
```
>>> chris_age = ages['Chris']
>>> chris_age
32
>>> karels_age = ages['Karel the Robot']
KeyError: 'Karel the Robot'
```

Accessing Elements of Dictionary

- Consider the following dictionary:

```
ages = {'Chris': 32, 'Juliette': 24, 'Mehran': 50}
```

- Like a set of variables that are indexed by keys



- Checking membership

```
>>> 'Mehran' in ages
True
>>> 'Karel the Robot' not in ages
True
```

Adding Elements to Dictionary

- Can add pairs to a dictionary:

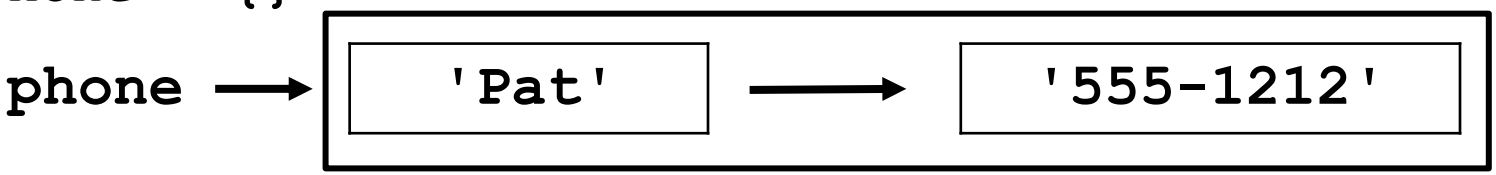
```
phone = {}
```

phone ⟶ *Empty dictionary*

Adding Elements to Dictionary

- Can add pairs to a dictionary:

```
phone = {}
```

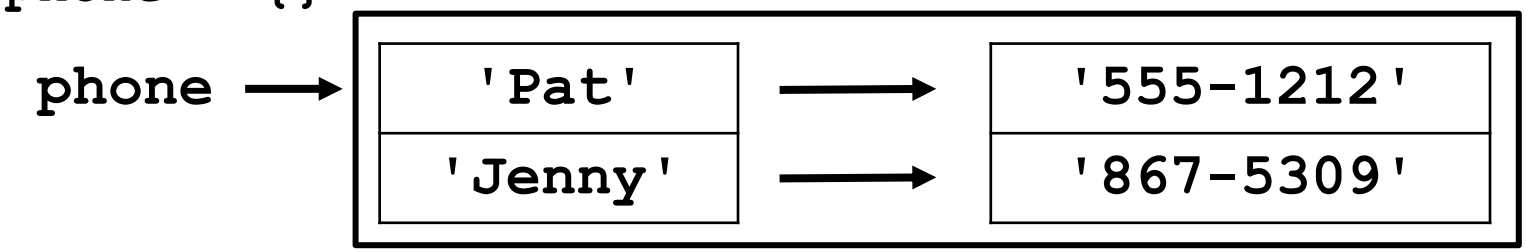



```
phone['Pat'] = '555-1212'
```

Adding Elements to Dictionary

- Can add pairs to a dictionary:

```
phone = {}
```

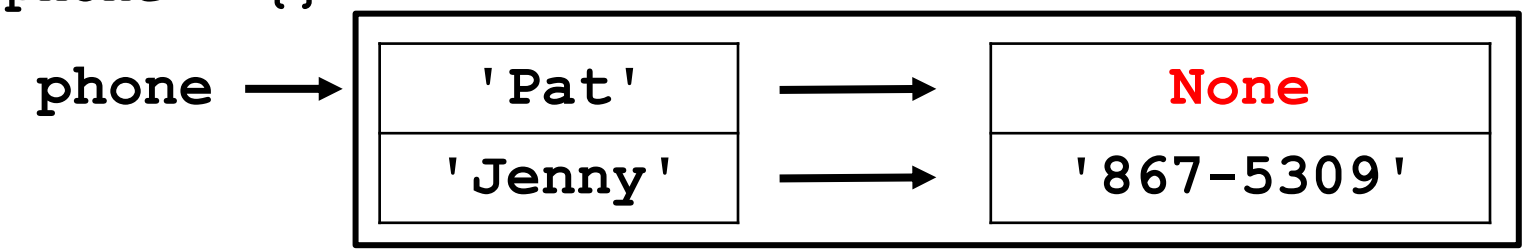



```
phone['Pat'] = '555-1212'
phone['Jenny'] = '867-5309'
```

Adding Elements to Dictionary

- Can add pairs to a dictionary:

```
phone = {}
```

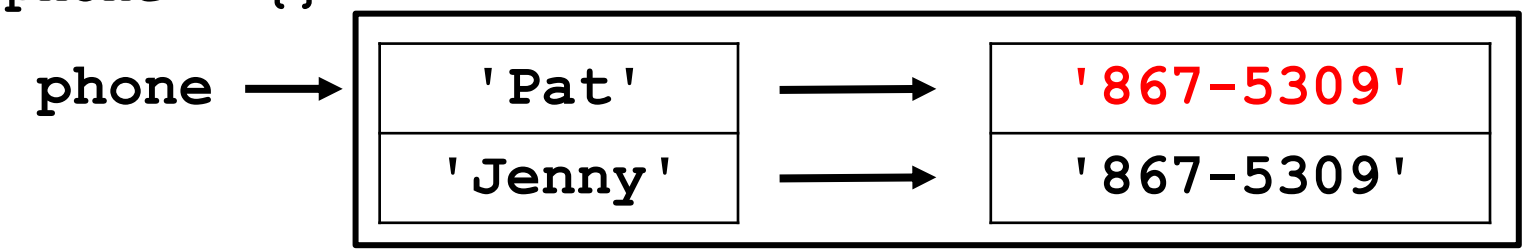



```
phone['Pat'] = '555-1212'
phone['Jenny'] = '867-5309'
phone['Pat'] = None
```

Adding Elements to Dictionary

- Can add pairs to a dictionary:

```
phone = {}
```




```
phone['Pat'] = '555-1212'
phone['Jenny'] = '867-5309'
phone['Pat'] = None
phone['Pat'] = '867-5309'
```

Looping Over Dictionary

```
ages = {'Chris': 32, 'Juliette': 24, 'Mehran': 50}
```

- Can also loop over a dictionary using for-each loop just using name of dictionary:

```
for key in ages:
    value = ages[key]
    print(str(key) + ' -> ' + str(value))
```

Terminal:

```
Chris -> 32
Juliette -> 24
Mehran -> 50
```

Lets Play!

“Mehran Sahami, as a pixar character, reading a dictionary”

Lets Play!

“Professor Mehran Sahami, as a pixar character, reading a dictionary”

List
index -> value

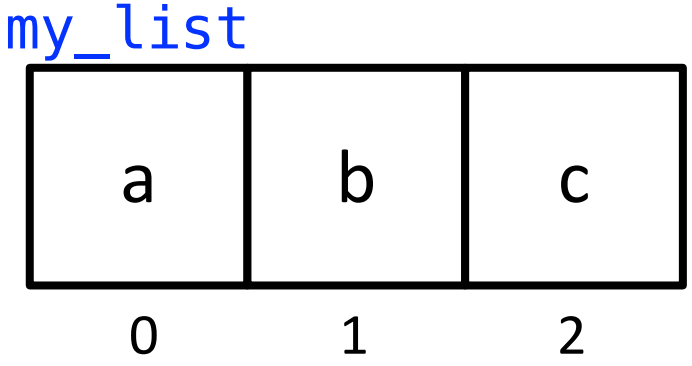
Dictionary
key -> value

List

```
my_list = ['a', 'b', 'c']


print(my_list[1])

for i in range(len(my_list)):
    value = my_list[i]
    print(i, value)
```

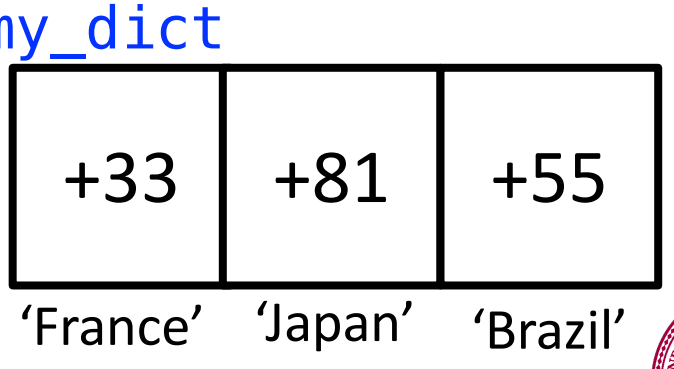


Dictionary

```
my_dict = {
   'France': '+33',
   'Japan': '+81',
   'Brazil':'+55'
}

print(my_list['France'])

for key in my_dict:
    value = my_dict[key]
    print(key, value)
```

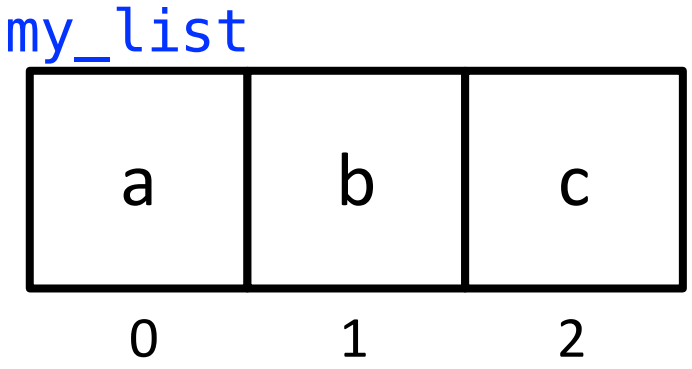


List

```
my_list = [
    'a',
    'b',
    'c'
]

print(my_list[1])

for i in range(len(my_list)):
    value = my_list[i]
    print(i, value)
```

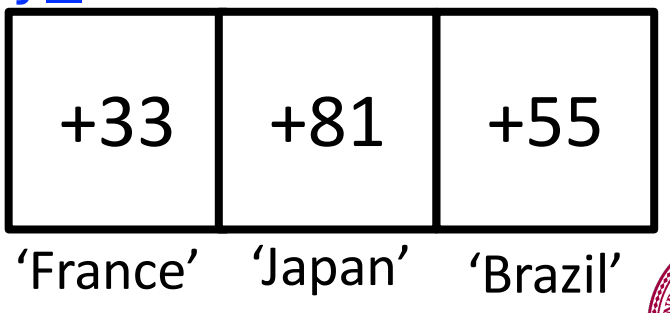


Dictionary

```
my_dict = {
  'France': '+33',
  'Japan': '+81',
  'Brazil':'+55'
]

print(my_list['France'])

for key in my_dict:
    value = my_dict[key]
    print(key, value)
```

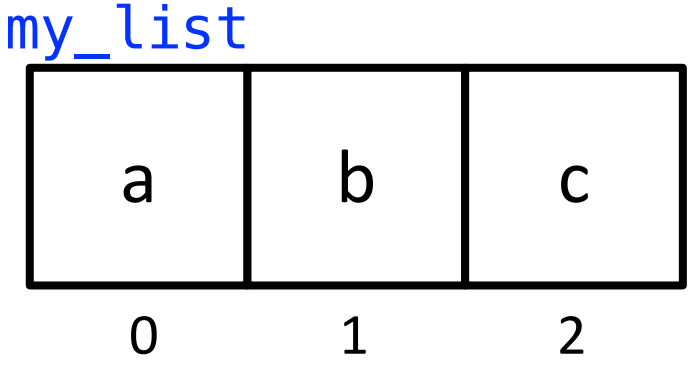


List

```
my_list = ['a', 'b', 'c']


print(my_list[1])

for i in range(len(my_list)):
    value = my_list[i]
    print(i, value)
```

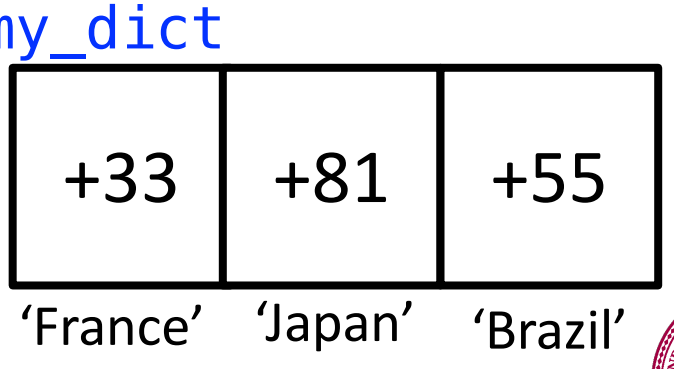


Dictionary

```
my_dict = {
  'France': '+33',
  'Japan': '+81',
  'Brazil':'+55'
}

print(my_list['France'])

for key in my_dict:
    value = my_dict[key]
    print(key, value)
```

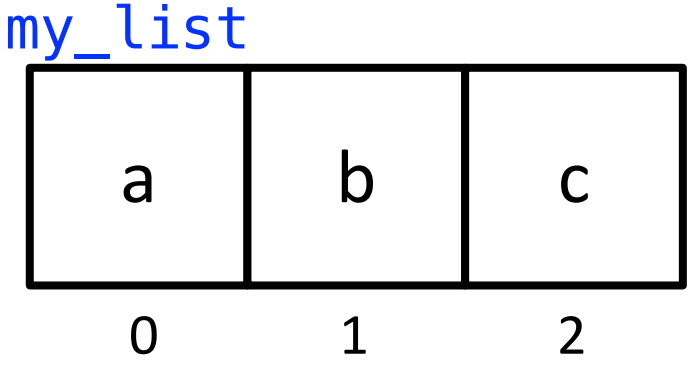


List

```
my_list = ['a', 'b', 'c']


print(my_list[1])

for i in range(len(my_list)):
    value = my_list[i]
    print(i, value)
```

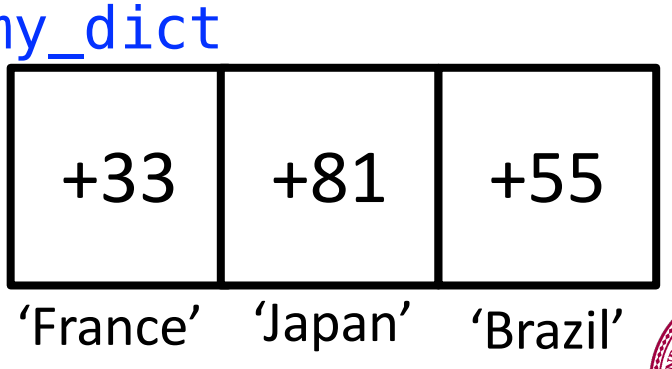


Dictionary

```
my_dict = {
  'France': '+33',
  'Japan': '+81',
  'Brazil':'+55'
}

print(my_list['France'])

for key in my_dict:
    value = my_dict[key]
    print(key, value)
```

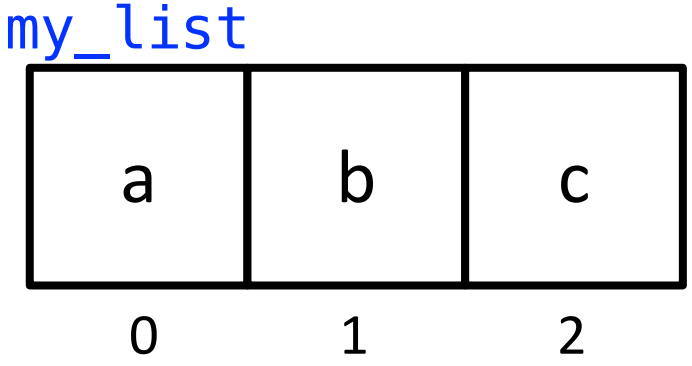


List

```
my_list = ['a', 'b', 'c']


print(my_list[1])

for i in range(len(my_list)):
    value = my_list[i]
    print(i, value)
```

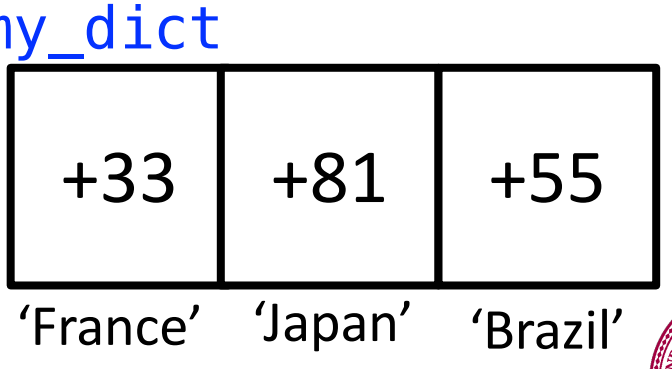


Dictionary

```
my_dict = {
  'France': '+33',
  'Japan': '+81',
  'Brazil':'+55'
}

print(my_list['France'])

for key in my_dict:
    value = my_dict[key]
    print(key, value)
```

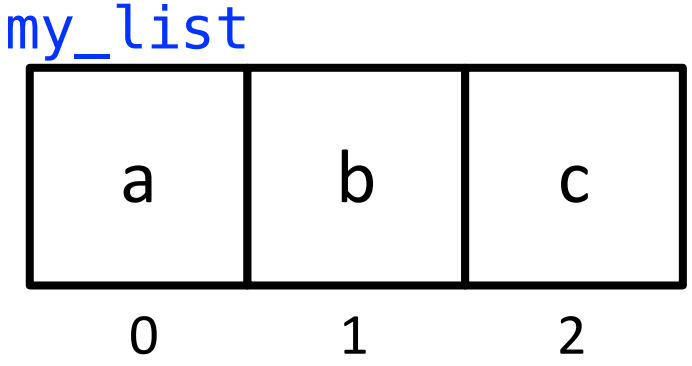


List

```
my_list = [‘a’, ‘b’, ‘c’]


print(my_list[1])

for i in range(len(my_list)):
    value = my_list[i]
    print(i, value)
```

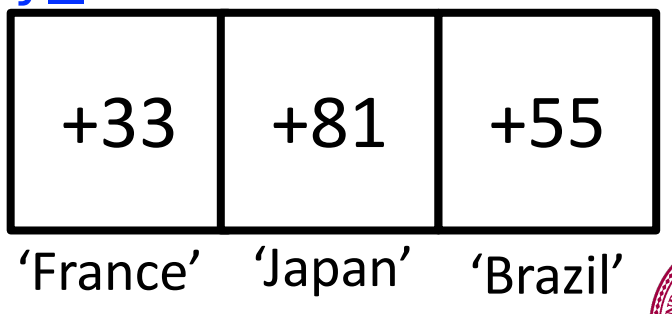


Dictionary

```
my_dict = {
   ‘France’: ’+33’,
   ‘Japan’: ’+81’,
   ‘Brazil’:’+55’
}

print(my_list[‘France’])

for key in my_dict:
    value = my_dict[key]
    print(key, value)
```



More Dictionary Fun! (Part 1)

```
ages = {'Chris': 32, 'Juliette': 24, 'Mehran': 50}
```

- Function: ***dict*.keys()**
  - Returns something similar to a range of the keys in dictionary
  - Can use that to loop over all keys in a dictionary:

```
for key in ages.keys():
    print(str(key) + " -> " + str(ages[key]))
```

Terminal:

```
Chris -> 32
Juliette -> 24
Mehran -> 50
```

  - Can turn **keys()** into a list, using the **list** function

```
>>> list(ages.keys())
['Chris', 'Juliette', 'Mehran']
```

More Dictionary Fun! (Part 2)

```
ages = {'Chris': 32, 'Juliette': 24, 'Mehran': 50}
```

- Function: ***dict*.values()**
  - Returns something similar to a range of the values in dictionary
  - Can use that to loop over all keys in a dictionary:

```
for value in ages.values():
    print(value)
```

Terminal:

```
32
24
50
```

  - Can turn **values()** into a list, using the **list** function

```
>>> list(ages.values())
[32, 24, 50]
```

More Dictionary Fun! (Part 3)

```
ages = {'Chris': 32, 'Juliette': 24, 'Mehran': 50}
```

- Function: ***dict*.items()**
  - Returns
  - Can use that to loop over all key value pairs in a dictionary:

```
for key, value in ages.items():
    print(f"{key}, {value}")
```

Terminal:

```
Chris, 32
Juliette, 24
Mehran, 50
```

More Dictionary Fun! (Part 4/5)

```
ages = {'Chris': 32, 'Juliette': 24, 'Mehran': 50}
```

- Function: ***dict*.pop(key)**
  - Removes key/value pair with the given key. Returns value from that key/value pair.

```
>>> ages
>>> {'Chris': 32, 'Juliette': 24, 'Mehran': 50}
>>> ages.pop('Mehran')
50
>>> ages
{'Chris': 32, 'Juliette': 24}
```

- Function: ***dict*.clear()**
  - Removes all key/value pairs in the dictionary.

```
>>> ages.clear()
>>> ages
{}
```


Functions You Can Apply

```
ages = {'Chris': 32, 'Juliette': 24, 'Mehran': 50}
```

- Function: `len(`*dict*`)`
  - Returns number of key/value pairs in the dictionary

```
>>> ages
{'Chris': 32, 'Juliette': 24, 'Mehran': 50}
>>> len(ages)
3
```

- Function: `del` *dict*`[key]`
  - Removes key/value pairs in the dictionary.
  - Similar to `pop`, but doesn't return anything.

```
>>> ages
{'Chris': 32, 'Juliette': 24, 'Mehran': 50}
>>> del ages['Mehran']
>>> ages
{'Chris': 32, 'Juliette': 24}
```


Example:
count_nums.py

Example:
phonebook.py

Learning Goals

1. Learning about dictionaries
2. Building programs using dictionaries