UCLA College | Social Sciences Economics

Day 3: Lists & Dictionaries, Functions

Paul Schumacher, MSc Quantitative Economics

Announcements

Attendance

Monday PLF Session??

Midterm

- Take Home + Open notes
- NOT Proctored → need camera
- 4:00 – 6:00 pm on Monday
- Upload your code

Quick Intro – 3 Students

1. Name, School, Hobbies, …
2. What are your future plans (university, work, hobbies, travel)
3. Experience in Python? Why do you want to learn it? What do you wish to get out of this course?
4. Which apps do you use most on your phone?
5. What was the best/ worst thing that happened to you last month?

Class exercise: For Loops & FV of Money

- “PLF Day 2 Worksheet A.ipynb”
- Breakout rooms
- Time: 20 min

Class Recap: I. List & Dictionaries

Store data

- List: stores by index, [ ]
- Dictionaries: stores by keys {‘key’: }

Functions:

- .reverse()
- .pop()

Functions:

- .insert(index, ”value")
- .remove(“value”)

Del vs. pop vs. remove

```
fruits = ['apple', 'banana', 'cherry']
fruits.remove('apple')
print(fruits)
```

```
['banana', 'cherry']
```

Never used it in my life!!

```
my_list = [1, 2, 3, 4, 5, 6, 7, 8, 9]

# deleting the third item
del my_list[2]

# Output: [1, 2, 4, 5, 6, 7, 8, 9]
print(my_list)

# deleting items from 2nd to 4th
del my_list[1:4]

# Output: [1, 6, 7, 8, 9]
print(my_list)

# deleting all elements
del my_list[:]

# Output: []
print(my_list)
```

removes item/ slices at a **given index**

For both (del, pop):
The index of item is needed!!!
Not direct removal of value

```
fruits = ['apple', 'banana', 'cherry']
x = fruits.pop(2)
print(fruits)
```

```
['apple', 'banana']
```

pop() method removes the element at the **specified position**

same

= del fruits[2]

```
# create a list
prime_numbers = [2, 3, 5, 7, 9, 11]

# remove 9 from the list
prime_numbers.remove(9)

# Updated prime_numbers List
print('Updated List: ', prime_numbers)

# Output: Updated List:  [2, 3, 5, 7, 11]
```

For remove: no index of item is needed!!! → direct removal of value

Removes first "matching" item that is given in the brackets

Class Recap: I. List & Dictionaries

.extend() vs .append()

- Add elements to list

multiple

```
>>> nums = [5.6, 7.44, 6.75, 4.56, 2.3]
>>> new_values = [2.3, 9.6, 4.564, 7.56]

# This is where the magic occurs! No more for loops
>>> nums.extend(new_values)

# The list was updated with individual values
>>> nums
[5.6, 7.44, 6.75, 4.56, 2.3, 2.3, 9.6, 4.564, 7.56]
```

single

```
# List that we want to modify
>>> nums = [5.6, 7.44, 6.75, 4.56, 2.3]

# Appending the items
>>> nums.append(2.3)
>>> nums.append(9.6)
>>> nums.append(4.564)
>>> nums.append(7.56)

# Updated list
>>> nums
[5.6, 7.44, 6.75, 4.56, 2.3, 2.3, 9.6, 4.564, 7.56]
```

Class Recap: II. Tuples () vs. List []

- Store Data
- List: Mutable → changeable
- Tuple → not changeable

```
seasons = ("Summer", "Spring", "Winter", "Fall")

print(seasons)

seasons[1] = "Winter"

print(seasons)
```

```
('Summer', 'Spring', 'Winter', 'Fall')
    seasons[1] = "Winter"
TypeError: 'tuple' object does not support item assignment
```

Class Recap: III. Dictionaries {}

- Store of Data + Changeable
- DictionaryName = {“key”: “value”, 123}
- Functions
  - .keys()
  - .values()
  - Direct calls to check and change → DictionaryName[“key”] = “value”/1234

Class exercise: Lists & Dictionaries

- “PLF Day 2 Worksheet B.ipynb”
- Breakout rooms
- Time: 20 min

Class Recap: IV. Functions

- Block of reusable code

Class Recap: IV. Functions

- Block of reusable code

```
print("Happy Birthday")
print("You are old")
print("Happy Birthday")
print()
```

```
Happy Birthday
You are old
Happy Birthday
```

```
def happy_birthday():
    print("Happy Birthday")
    print("You are old")
    print("Happy Birthday")
    print()
```

```
happy_birthday()
```

```
Happy Birthday
You are old
Happy Birthday
```

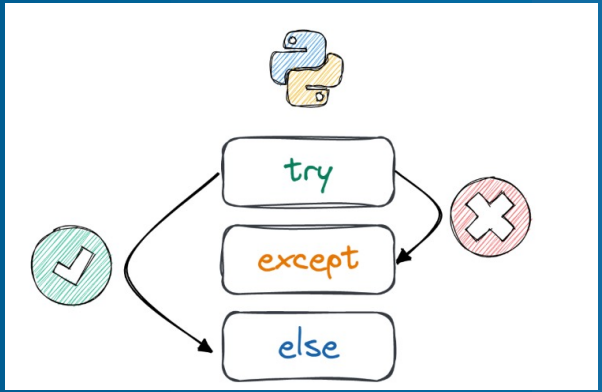
```
def happy_birthday(name, age):
    print(f"Happy Birthday, {name}")
    print(f"You are {age} years old")
    print("Happy Birthday")
    print()
```

```
happy_birthday("Mom", 43)
```

```
Happy Birthday, Mom
You are 43 years old
Happy Birthday
```

Class Recap: IV. Functions

- print(): final output of a code on the console
- return: final value of a function execution



Any Questions?