Lecture 15 – the split() method, updating the elements of lists, lists are mutable objects

Learning outcomes

At the end of this lecture, students should be able to:
• use the index number to access individual elements of a list
• make changes to the elements of a list
• copy the values of a list
• use the split() method on a string to obtain a list of string objects
• lists are mutable objects

Recap

From lecture 14
• we can iterate through the elements of a list using a for...in loop
• calculations can be done using the values in the elements of a list

```
def print_xs(a_list):
    for item in a_list:
        if item == True:
            print("X", end="")
        else:
            print(" ", end="")

def main():
    print("01234567890123456789")
    print("XX X XX X")
    list1 = [True, False, False, True, False, True, False, False, False, True, True, False, False, True]
    print_xs(list1)
    main()
```

Lists Recap - accessing list elements

The elements of a list can be accessed from the end of the list by using a negative index value.

```
my_list = [10, 20, 30, 40, 50]
print(my_list[-4])
my_list[-3] = my_list[-1] + my_list[-2]
print(my_list[-3], my_list[1], my_list[-5])
```

```
my_list 0 10 20 30 40 50
0 10 -5
1 20 -4
2 30 -4
3 40 -2
4 50 -1
```
Mutable Objects, Immutable Objects

Strings and int objects are immutable (look at outputs 1. and 2.).

List objects are mutable (look at output 3.).

```
def main():
    list1 = [6, 4, 7]
    value1 = list1[0]
    value1 = value1 + 3
    print("1.", list1, value1)

    list2 = ["a", "b", "c"]
    value2 = list2[0]
    value2 = value2.upper()
    print("2.", list2, value2)

    list3 = ["a", "b", "c"]
    value3 = list3
    value3[0] = value3[0].upper()
    print("3.", list3, value3)
main()
```

1. [6, 4, 7] 9
2. ['a', 'b', 'c'] A
3. ['A', 'b', 'c'] ['A', 'b', 'c']

Why does the following not work as intended?

In the following for...in loop, each element of the list is accessed but ...

```
def main():
    a_list = [10, 8, 6, 4, 7]
    print("1.", a_list)

    for number in a_list:
        number = number * 2
        print(number, end=" ")

    print()
main()
```

```
[10, 8, 6, 4, 7] 20 16 12 8 14
```

What if the intention was to update the element values in the list?

```
def main():
    my_list = [10, 8, 6, 4, 7]
    for index in range(len(my_list)):
        print(index, my_list[index] * 2)
main()
```

```
1. [10, 8, 6, 4, 7] 20 16 12 8 14
3. [10, 8, 6, 4, 7]
```

Changing a value at an index location updates the element of the list.

Updating the elements in the list

The elements in a list can be updated if we assign to each element of the list using the index of the element, e.g.,

```
def main():
    a_list = [10, 8, 6, 4, 7]
    print("1.", a_list)
    number_of_elements = len(a_list)

    for index in range(number_of_elements):
        a_list[index] = a_list[index] * 2
        print("2.", a_list)
main()
```

```
1. [10, 8, 6, 4, 7] 2. [20, 16, 12, 8, 14]
```
Complete the main() function

Complete the code in the main() function which adds 1 to each list element which has odd value.

```python
import random
def main():
    a_list = []
    for index in range(10):
        a_list = a_list + [random.randrange(1, 100)]
    print("1.", a_list)
    print("2.", a_list)
main()
```

Complete the main() function

Complete the code in the main() function which changes the elements starting from index 1 so that each element is the accumulative total of the previous elements (i.e., element 1 is the sum of the element 0 and element 1, element 2 is the sum of element 1 and element 2, etc.).

```python
import random
def main():
    a_list = []
    for num in range(10):
        a_list = a_list + [random.randrange(1, 10)]
    print("1.", a_list)
    print("2.", a_list)
main()
```

The split() method

The split() method separates a single string into a list of the parts of the string (the tokens) using the separator defined (inside the parentheses). Each element of the list is a string object. This method can be applied to any string object.

If no separator is defined (as in the code below), whitespace is the default separator, e.g.,

```python
def main():
    phrase = 'The best cure for insomnia is to get a lot of sleep'
    words_list = phrase.split()
    print(words_list[0], words_list[4], words_list[7])
main()
```

The split() method - example

```python
def main():
    prompt = "Enter a line of numbers: ",
    line_of_nums = input(prompt)
    line_of_nums = line_of_nums.split()
    for index in range(len(line_of_nums)):
        line_of_nums[index] = int(line_of_nums[index])
    total = 0
    for number in line_of_nums:
        total = total + number
    print("Total:", total)
main()
```

Note that split() function breaks a string up into a list of strings.
def split_message(message):
    words = message.split()
    num = int(words[1])
    num = num + 4
    words[2] = num
    a_word = words[0]
    words[0] = a_word[:3]
    print(words[1], words[0], words[2], sep = "")

def main():
    phrase = 'tuna 4 lunch'
    split_message(phrase)

main()
Examples of Python features used in this lecture

def change_list(a_list):
    number_of_elements = len(a_list)
    for i in range(number_of_elements):
        a_list[i] = a_list[i] * 2

def use_lists(list1, list2):
    list3 = []
    for index in range(len(list1)):
        list3 = list3 + [list1[index] + list2[index]]
    return list3

def split_message(message):
    words = message.split()
    print(words[2], words[0])