Python Lists

- List

  A list is a collection which is ordered and changeable. In Python lists are written with square brackets.

- Example

- Create a List:

  thislist = ["apple", "banana", "cherry"]
  print(thislist)

```
thislist = ["apple", "banana", "cherry"]
print(thislist)
```

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

- **Access Items**

  You access the list items by referring to the index number:

- **Example**

- Print the second item of the list:

  thislist = ["apple", "banana", "cherry"]
  print(thislist[1])

```
thislist = ["apple", "banana", "cherry"]
print(thislist[1])
```

```
banana
```

- **Negative Indexing**

  Negative indexing means beginning from the end, -1 refers to the last item, -2 refers to the second last item etc.

- **Example**

- **Print the last item of the list:**

  thislist = ["apple", "banana", "cherry"]
  print(thislist[-1])

```
thislist = ["apple", "banana", "cherry"]
print(thislist[-1])
```

```
cherry
```

- Range of Indexes

  You can specify a range of indexes by specifying where to start and where to end the range.

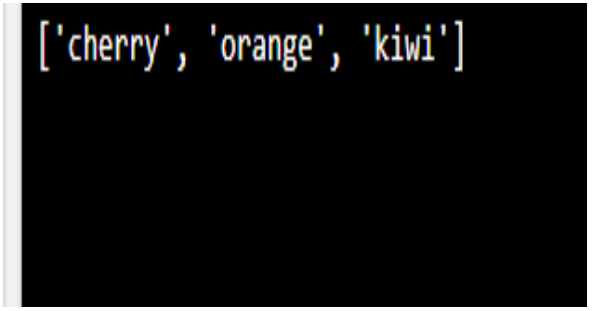
  When specifying a range, the return value will be a new list with the specified items.

- Example
- Return the third, fourth, and fifth item:

  thislist = ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]
  print(thislist[2:5])

```
thislist = ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]
print(thislist[2:5])

#This will return the items from position 2 to 5.
```



- By leaving out the start value, the range will start at the first item:
- Example
- This example returns the items from the beginning to "orange":

```
thislist = ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]
print(thislist[:4])
```

```
thislist = ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]
print(thislist[:4])

#This will return the items from index 0 to index 4.
```

```
['apple', 'banana', 'cherry', 'orange']
```

- **By leaving out the end value, the range will go on to the end of the list:**
- **Example**
- **This example returns the items from "cherry" and to the end:**

```
thislist = ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]
print(thislist[2:])
```

```
thislist = ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]
print(thislist[2:])
```

```
['cherry', 'orange', 'kiwi', 'melon', 'mango']
```

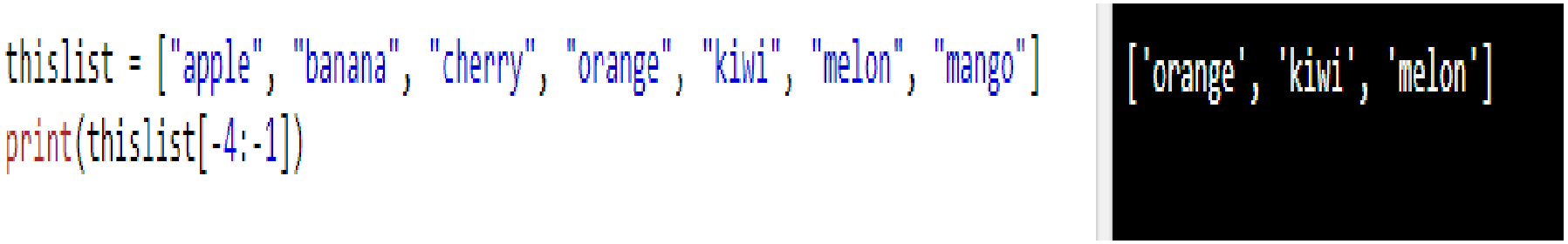
- **Range of Negative Indexes**

  Specify negative indexes if you want to start the search from the end of the list:

- **Example**

- **This example returns the items from index -4 (included) to index -1 (excluded)**

```
thislist = ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]
print(thislist[-4:-1])
```



- Change Item Value

  To change the value of a specific item, refer to the index number:

- Example

- Change the second item:

thislist = ["apple", "banana", "cherry"]
thislist[1] = "blackcurrant"
print(thislist)

```
thislist = ["apple", "banana", "cherry"]
thislist[1] = "blackcurrant"

print(thislist)
```

```
['apple', 'blackcurrant', 'cherry']
```

- Loop Through a List

  You can loop through the list items by using a for loop:

- Example

- Print all items in the list, one by one:

```
thislist = ["apple", "banana", "cherry"]
for x in thislist:
  print(x)
```

```
thislist = ["apple", "banana", "cherry"]
for x in thislist:
  print(x)
```

```
apple
banana
cherry
```

- Check if Item Exists

  To determine if a specified item is present in a list use the in keyword:

- Example
- Check if "apple" is present in the list:

```
thislist = ["apple", "banana", "cherry"]
if "apple" in thislist:
  print("Yes, 'apple' is in the fruits list")
```

- List Length

  To determine how many items a list has, use the len() function:

- Example
- Print the number of items in the list:

```
thislist = ["apple", "banana", "cherry"]
print(len(thislist))
```

```
thislist = ["apple", "banana", "cherry"]
print(len(thislist))
```

```
3
```

- Add Items

  To add an item to the end of the list, use the append() method:

- Example
- Using the append() method to append an item:

```
thislist = ["apple", "banana", "cherry"]
thislist.append("orange")
print(thislist)
```

```
thislist = ["apple", "banana", "cherry"]

thislist.append("orange")

print(thislist)
```

```
['apple', 'banana', 'cherry', 'orange']
```

- **To add an item at the specified index, use the insert() method:**
- **Example**
- **Insert an item as the second position:**

```
thislist = ["apple", "banana", "cherry"]
thislist.insert(1, "orange")
print(thislist)
```

- **Remove Item**
  - **There are several methods to remove items from a list:**
  - **Example**
  - **The remove() method removes the specified item:**

```
thislist = ["apple", "banana", "cherry"]
thislist.remove("banana")
print(thislist)
```

Example

```
thislist = ["apple", "banana", "cherry"]
thislist.remove("banana")
print(thislist)
```

```
['apple', 'cherry']
```

last item if index is not specified):

```
thislist = ["apple", "banana", "cherry"]
thislist.pop()
print(thislist)
```

```
thislist = ["apple", "banana", "cherry"]
thislist.pop()
print(thislist)
```

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

```
thislist = ["apple", "banana", "cherry"]
del thislist[0]
print(thislist)
```

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

- **Example**
- **The del keyword can also delete the list completely**:

```
thislist = ["apple", "banana", "cherry"]
del thislist
```

```
thislist = ["apple", "banana", "cherry"]
del thislist
print(thislist) #this will cause an error because you have succsesfully deleted
"thislist".
```

```
Traceback (most recent call last):
  File "demo_list_del2.py", line 3, in <module>
    print(thislist) #this will cause an error because you have succsesfully del
NameError: name 'thislist' is not defined
```

- Example
- The clear() method empties the list:

  thislist = ["apple", "banana", "cherry"]
  thislist.clear()
  print(thislist)

```
thislist = ["apple", "banana", "cherry"]
thislist.clear()
print(thislist)
```

- Copy a List

  You cannot copy a list simply by typing list2 = list1, because: list2 will only be a *reference* to list1, and changes made in list1 will automatically also be made in list2.

  There are ways to make a copy, one way is to use the built-in List method copy().

- Example
- Make a copy of a list with the copy() method:

  thislist = ["apple", "banana", "cherry"]
  mylist = thislist.copy()
  print(mylist)

```
thislist = ["apple", "banana", "cherry"]
mylist = thislist.copy()
print(mylist)
```

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

- Join Two Lists

  There are several ways to join, or concatenate, two or more lists in Python.

  One of the easiest ways are by using the + operator.

- Example
- Join two list:

```
list1 = ["a", "b" , "c"]
list2 = [1, 2, 3]

list3 = list1 + list2
print(list3)
```

```
['a', 'b', 'c', 1, 2, 3]
```

- **Another way to join two lists are by appending all the items from list2 into list1, one by one**:

```
list1 = ["a", "b" , "c"]
list2 = [1, 2, 3]

for x in list2:
  list1.append(x)

print(list1)
```

```
['a', 'b', 'c', 1, 2, 3]
```

- Or you can use the extend() method, which purpose is to add elements from one list to another list

```
list1 = ["a", "b" , "c"]
list2 = [1, 2, 3]

list1.extend(list2)
print(list1)
```

```
['a', 'b', 'c', 1, 2, 3]
```

- **Python List reverse() Method**
- **Example**

Reverse the order of the fruit list:

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

```
fruits = ['apple', 'banana', 'cherry']

fruits.reverse()

print(fruits)
```

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

- **Python List sort() Method**
- **Example**
- **Sort the list alphabetically:**

  cars = ['Ford', 'BMW', 'Volvo']

  Cars.sort()

```
cars = ['Ford', 'BMW', 'Volvo']

cars.sort()

print(cars)
```

```
['BMW', 'Ford', 'Volvo']
```

Thank You