List

Dr. Dongchul Kim

List

A list is a collection that is ordered and changeable. In Python, lists are defined with square brackets.

For example,

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

List

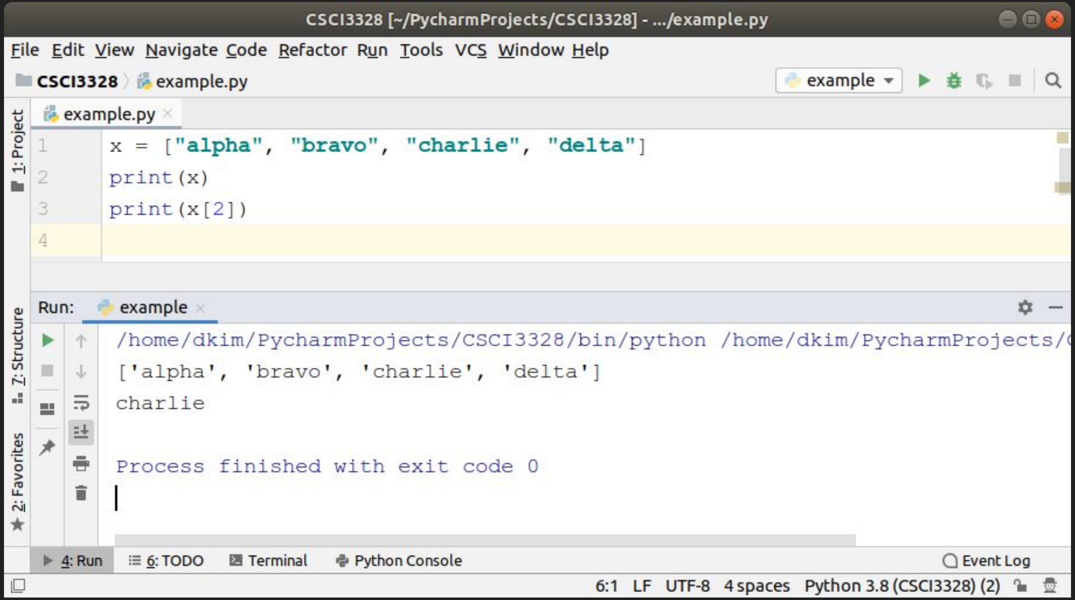
Unlike arrays in some languages, lists can hold elements of different data types.

For example,

```
myList = [3, 'hello', True, [1, 2, 3]]
```

How to access

Accessing Array Elements: Utilize zero-based indexing starting from zero


CSCI3328 [~/PycharmProjects/CSCI3328] - .../example.py
File Edit View Navigate Code Refactor Run Tools VCS Window Help
CSCI3328 example.py
example
example.py
Run: example
4: Run 6: TODO Terminal Python Console Event Log
6:1 LF UTF-8 4 spaces Python 3.8 (CSCI3328) (2)

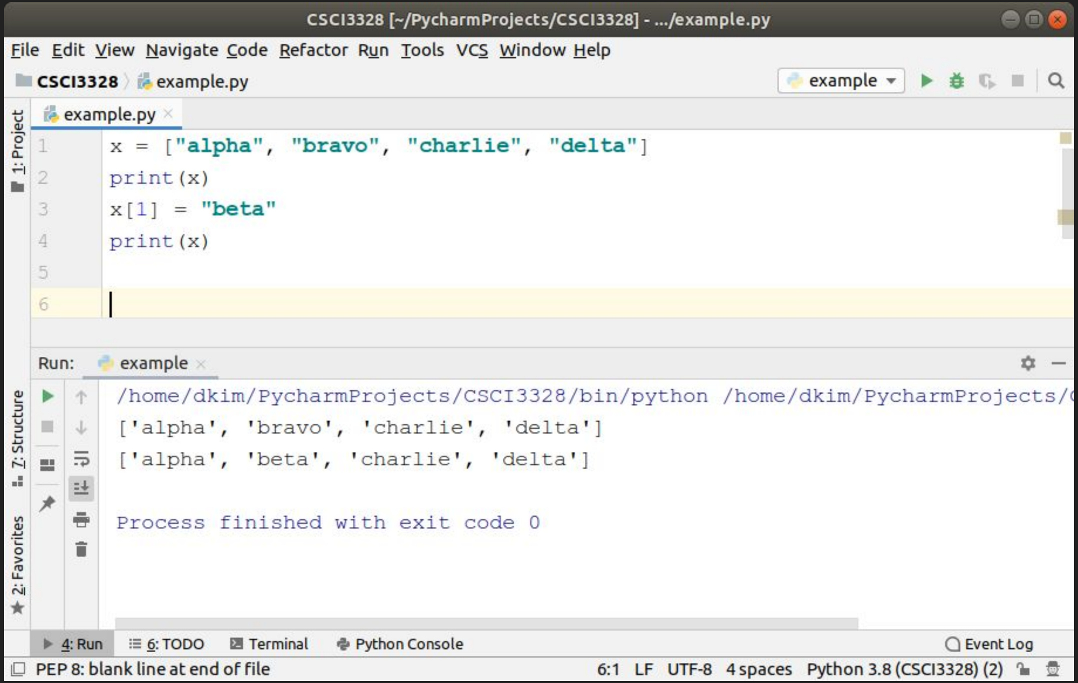
```python
x = ["alpha", "bravo", "charlie", "delta"]
print(x)
print(x[2])
```

```
/home/dkim/PycharmProjects/CSCI3328/bin/python /home/dkim/PycharmProjects/
['alpha', 'bravo', 'charlie', 'delta']
charlie

Process finished with exit code 0
```

How to modify

Modifying Array Values: Reference elements by their index numbers


CSCI3328 [~/PycharmProjects/CSCI3328] - .../example.py
File Edit View Navigate Code Refactor Run Tools VCS Window Help
CSCI3328 example.py
example
example.py
Run: example
4: Run 6: TODO Terminal Python Console Event Log
PEP 8: blank line at end of file 6:1 LF UTF-8 4 spaces Python 3.8 (CSCI3328) (2)

```python
x = ["alpha", "bravo", "charlie", "delta"]
print(x)
x[1] = "beta"
print(x)

```

```
/home/dkim/PycharmProjects/CSCI3328/bin/python /home/dkim/PycharmProjects/
['alpha', 'bravo', 'charlie', 'delta']
['alpha', 'beta', 'charlie', 'delta']

Process finished with exit code 0
```

To create a subset of a list, you can define a range of indexes, indicating the starting and ending points. I

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

If you omit the start index, the range begins from the first item in the list.

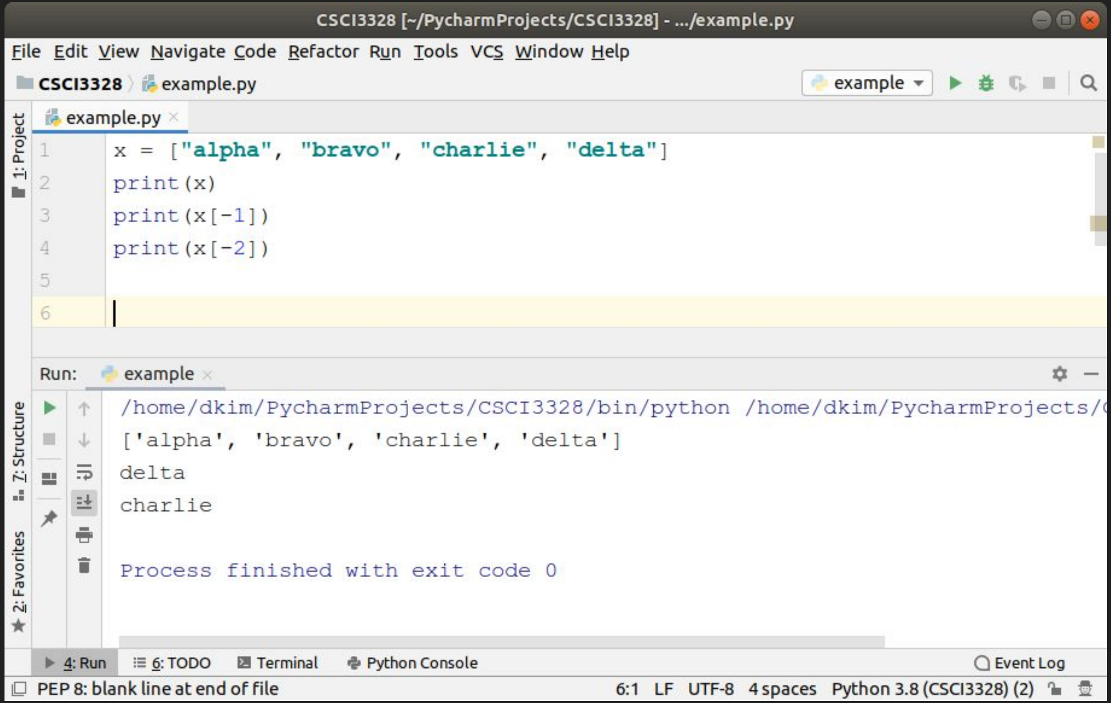
```
thislist = ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]
print(thislist[:4])
```

Conversely, omitting the end index will extend the range to include all items from the start index to the end of the list.

```
thislist = ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]
print(thislist[3:])
```

Negative Index

In Python lists, negative indexing starts from the end of the list, where -1 represents the last item, -2 the second-to-last item, and so on.



CSCI3328 [~/PycharmProjects/CSCI3328] - .../example.py
File Edit View Navigate Code Refactor Run Tools VCS Window Help
CSCI3328 example.py
example
example.py
Run: example
4: Run 6: TODO Terminal Python Console
Event Log
PEP 8: blank line at end of file
6:1 LF UTF-8 4 spaces Python 3.8 (CSCI3328) (2)

```python
x = ["alpha", "bravo", "charlie", "delta"]
print(x)
print(x[-1])
print(x[-2])

```

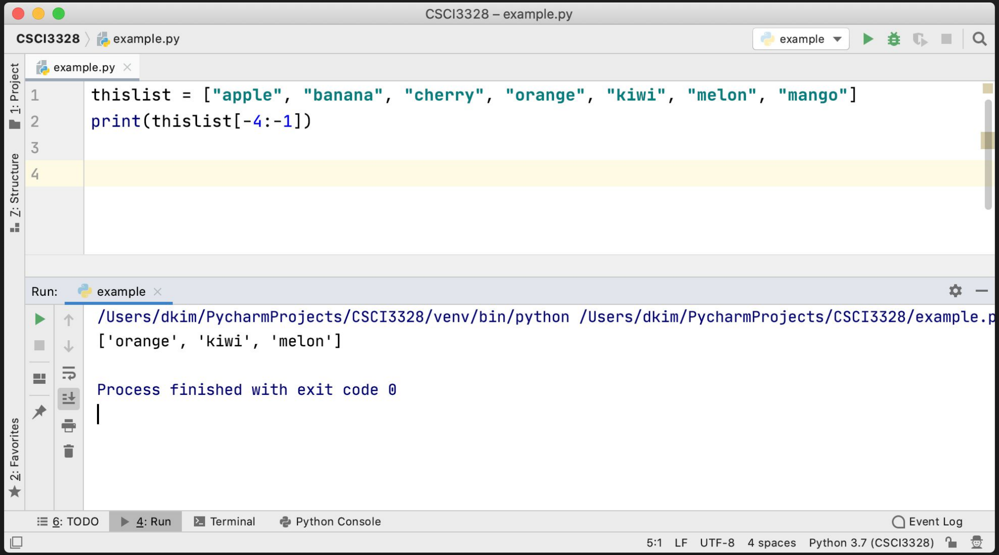
```
/home/dkim/PycharmProjects/CSCI3328/bin/python /home/dkim/PycharmProjects/
['alpha', 'bravo', 'charlie', 'delta']
delta
charlie

Process finished with exit code 0
```

Range of Negative Indexes

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

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



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

```
/Users/dkim/PycharmProjects/CSCI3328/venv/bin/python /Users/dkim/PycharmProjects/CSCI3328/example.p
['orange', 'kiwi', 'melon']

Process finished with exit code 0
```

`len()`

To find the number of elements in a Python list, you can use the `len()` function. This function returns the total count of elements in the list.

For example:

```
fruits = ["mango", "apple", "grapes"]

print (len(fruits))
```

More useful functions

`len(list)`: Returns the number of items in the list.

`max(list)`: Returns the item with the highest value in the list.

`min(list)`: Returns the item with the lowest value in the list.

`sorted(list)`: Returns a new sorted list from the elements of the original list.

`sum(list)`: Calculates the sum of the elements in the list (numeric types only).

List Methods

`list.append(element):` Adds a single element to the end of the list.

`list.extend(iterable)`: Adds all elements of an iterable (e.g., another list) to the end of the list.

`list.insert(index, element)`: Inserts an element at the specified index.

`list.remove(element)`: Removes the first occurrence of the element from the list.

`list.pop([index])`: Removes and returns the element at the given index (or the last item if the index is not specified).

`list.clear()`: Removes all items from the list.

`list.count(element)`: Returns the number of occurrences of the element in the list.

`list.sort()`: Sorts the items

Lab 7

1. **Create a List:**
   Write a Python program to create a list named it_companies.
   This list should contain the names of five different IT companies. For example, ["Apple", "Google", "Microsoft", "Amazon", "Facebook"].

2. **Slicing the List:**
   Using slicing, print the third and fourth company names from the list. Remember, list indexing in Python starts at 0.

3. **Negative Indexing:**
   Use negative indexing to print the last two company names in the list.

4. **List Length:**
   Use the len() function to print the length of the list it_companies.
   deadline.

Submission:

Run your Python script and capture a screenshot of the code along with its output.

Save your Python script as a .py file.

Upload both the screenshot and the .py file to Blackboard for evaluation.

Lab 7

List Creation (20 Points):
Successfully creating a list named it_companies with five different IT company names (20 Points)
Incomplete or incorrect list creation (0 Points)

Slicing the List (20 Points):
Correctly using slicing to print the third and fourth company names from the list (20 Points)
Incorrect slicing or output (0 Points)

Negative Indexing (20 Points):
Accurately using negative indexing to print the last two company names in the list (20 Points)
Incorrect negative indexing or output (0 Points)

List Length (20 Points):
Correctly using the len() function to print the length of the it_companies list (20 Points)
Incorrect usage of len() function or output (0 Points)

Submission (20 Points):
Submitting a clear screenshot showing both the code and its output, and the Python script saved as a .py file (20 Points)
Incomplete or unclear screenshot and/or missing .py file (0 Points)

Total Points: 100

Tuple

Tuples, similar to lists, are a type of sequence in Python.

However, there are key differences between the two. Unlike lists, tuples are immutable, meaning they cannot be altered once created. Additionally, tuples are defined using parentheses () instead of the square brackets [] used for lists.

To create a tuple, simply separate values with commas. While enclosing these values in parentheses is optional, it is a common practice for clarity.

For example,

```
tup1 = ('physics', 'chemistry', 1997, 2000);

tup2 = (1, 2, 3, 4, 5 );

tup3 = "a", "b", "c", "d";
```

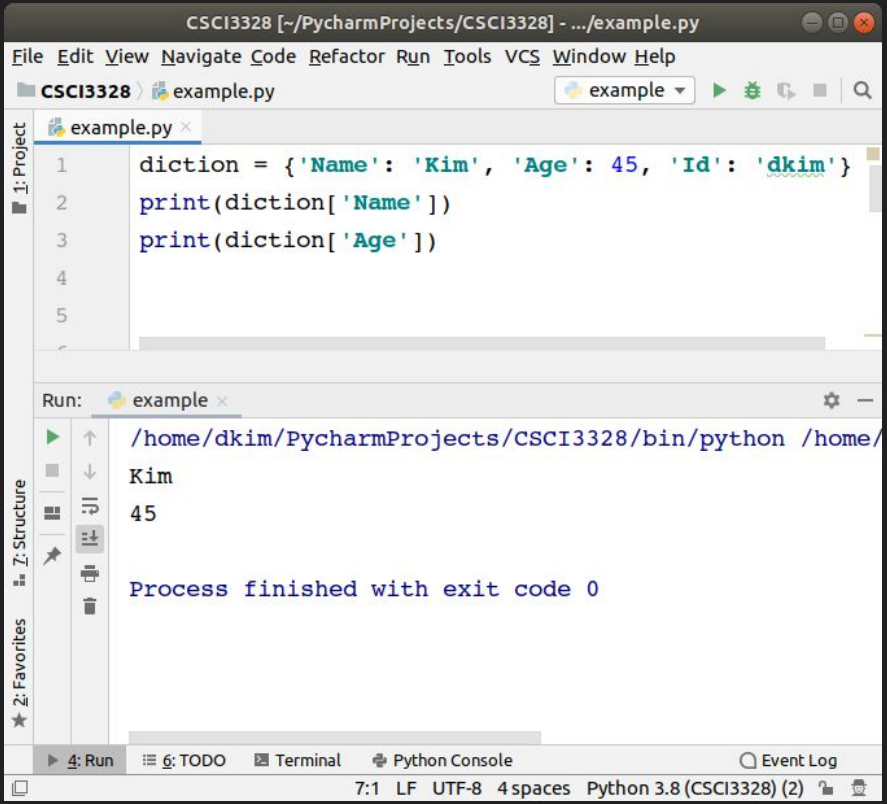
Dictionary

In a dictionary, each key-value pair is distinctly separated by a colon (:), and individual pairs are separated by commas.

The entire collection of pairs is enclosed within curly braces {}.

An empty dictionary is represented by two curly braces without any content: {}.

It's important to note that while dictionary values can be of any data type and may be duplicated, **keys must be unique and of an immutable type**, such as strings, numbers, or tuples.


CSCI3328 [~/PycharmProjects/CSCI3328] - .../example.py
File Edit View Navigate Code Refactor Run Tools VCS Window Help
CSCI3328 example.py example
example.py
Run: example
4: Run 6: TODO Terminal Python Console Event Log
7:1 LF UTF-8 4 spaces Python 3.8 (CSCI3328) (2)

```
diction = {'Name': 'Kim', 'Age': 45, 'Id': 'dkim'}
print(diction['Name'])
print(diction['Age'])
```

```
/home/dkim/PycharmProjects/CSCI3328/bin/python /home/
Kim
45

Process finished with exit code 0
```