Hila Levitan

Data Structures

List –[Ordered, mutable collection]

Tuple –(Ordered, immutable collection)

Set –{ Unordered, unique elements }

Dictionary –{ Key: value pairs }

Examples of Data Structures

```python
# List
fruits = ['apple', 'banana', 'cherry']

# Tuple
coordinates = (10, 20)

# Set
unique_numbers = {1, 2, 3, 4}

# Dictionary
person = {'name': 'Hila', 'age': 25}
```

Looping Over Data Structures

Using `for` Loops:

```
14  # List :  ['apple', 'banana', 'cherry']
15  for fruit in fruits:
16      print(fruit)
17
18  # Tuple : (10, 20)
19  for coord in coordinates:
20      print(coord)
21
22  # Set : {1, 2, 3, 4}
23  for number in unique_numbers:
24      print(number)
25
26  # Dictionary : {'name': 'Hila', 'age': 25}
27  for key, value in person.items():
28      print(f"{key}: {value}")
```

Advanced Looping Techniques

Dictionary Keys, Values, and Items:

```
26  # Dictionary : {'name': 'Hila', 'age': 21}
27  for key, value in person.items():
28      print(f"{key}: {value}")
29
30  for key in person.keys():
31      print(key)
32
33  for value in person.values():
34      print(value)
35
```

Accessing Specific Elements

```python
# List
fruits = ['apple', 'banana', 'cherry']
print(fruits[0])  # Output: 'apple'

# Tuple
coordinates = (10, 20)
print(coordinates[1])  # Output: 20

# Set
unique_numbers = {1, 2, 3, 4}
# Sets are unordered; specific element access via indexing is not possible.

# Dictionary
person = {'name': 'Hila', 'age': 21}
print(person['name'])  # Output: 'Hila'
```

The in operator

```python
# tuple
numbers = (10, 20, 30)
if 20 in numbers:
    print("20 is in the tuple!")
# set
unique_numbers = {1, 2, 3}
if 4 in unique_numbers:
    print("4 is in the set!")
else:
    print("4 is not in the set!")
# dictionary
person = {'name': 'Hila', 'age': 21}
if 'name' in person:
    print("The key 'name' exists in the dictionary!")

if 21 in person.values():
    print("The value 21 exists in the dictionary!")
```