Python - Data Structures

```
Module Code: ELEE1147

Module Name: Programming for Engineers

Credits: 15

Module Leader: Seb Blair BEng(H) PGCAP MIET MIHEEM FHEA
```

Download as a PDF

Data Structures in Python

- Essential components of programming
- Help manage and organize data effectively
- Python has built-in and user-defined data structures

Types of Data Structures in Python

1. **Primitive Data Structures**
    - Integers, Floats, Strings, Booleans
2. **Non-Primitive Data Structures**
    - Lists, Tuples, Sets, Dictionaries
    - User-defined structures

Lists

- Ordered, mutable, allows duplicate elements
- Defined with square brackets `[]`
- Common methods: `append()`, `extend()`, `insert()`, `pop()`, `remove()`,

```
# Example
fruits = ["apple", "banana", "cherry"]

fruits.append("orange") # ["apple", "banana", "cherry", "orange"]

fruits.extend(["mango", "grape"]) # ["apple", "banana", "cherry", "orange", "mango", "grape"]

fruits.insert(1, "blueberry") # ["apple", "blueberry" ,"banana", "cherry", "orange", "mango", "grape"]

fruits.pop(2)  # ["apple", "blueberry", "orange", "mango", "grape"]

fruits.remove("banana") # ["apple", "blueberry", "banana", "orange", "mango", "grape"]

fruits.reverse() # ['grape', 'mango', 'orange', 'banana', 'blueberry', 'apple']
```

Tuples

- Ordered, immutable, allows duplicate elements
- Defined with parentheses `()`
- Suitable for fixed data collections

```
# Example
coordinates = (10, 20, 30) # X ,Y, Z

coordinates.index[10] # returns: 1

coordinates[0:2] # sliced: 20, 30

coordinates = coordinates * 3 # (10, 20, 30, 10, 20, 30, 10, 20, 30)

len(coordinates) # 9
```

Sets

- Unordered, mutable, does not allow duplicates
- Defined with curly braces `{}` or `set()`
- Common methods: `add()`, `remove()`, `union()`, `intersection()`

```
# Example
unique_numbers = {1, 2, 3, 4}#

unique_numbers.add(5) # {1, 2, 3, 4, 5}

unique_numbers.update([6, 7, 8]) # {1, 2, 3, 4, 5, 6, 7, 8}

unique_numbers.remove(3) # {1, 2, 4, 5, 6, 7, 8}

unique_numbers.discard(9) # No error if not there unlike .remove

unique_numbers.clear() # set() is now empty.
```

Dictionaries

- Key-value pairs, unordered, mutable
- Defined with curly braces `{key: value}`
- Useful for storing related data

```
# Example
server_info = {
    "hostname": "server01",
    "ip_address": "192.168.1.10",
    "status": "active",
    "cpu_load": 73.5,
    "services": ["httpd", "sshd", "mysql"]
}

server_info["hostname"] # Output: "server01"

server_info.get("uptime", "Not Available")  # Output: "Not Available" (if key not present)

server_info["disk_usage"] = 85.2 # server_info now contains disk_usage

server_info["disk_usage"].pop(0)  # "httpd" removed from "services"

server_info.keys() # Output: dict_keys(["hostname", "status", "cpu_load", "services"])

server_info.values() # Output: dict_values(["server01", "active", 73.5, ["sshd", "mysql"]])
```

Comparison of Data Structures

Feature	List	Tuple	Set	Dictionary
Ordered	Yes	Yes	No	No
Mutable	Mutable	Yes	No	Yes
Duplicate	Duplicate	Allowed	Allowed	Not Allowed
====	====	===	=====	=====
Memory Usage	Medium	Low	Medium	High
Access Time	O(1) for index-based access	O(1) for index-based access	O(1) average for membership check	O(1) average for key access
Insert/Delete Time	O(n) average (insert at end O(1))	Not Applicable O(1)	average for add/remove	O(1) average for add/remove