CompSci 101 Lab 7 Part A

Tuples

Tuples

- A tuple is a comma-separated sequence of values and can be indexed just like lists.

Example:

```
contact_details_tuple = ("Ann", 413, "923-4947")
name = contact_details_tuple[0]
room = contact_details_tuple[1]
phone = contact_detals_tuple[2]
```

- However, unlike lists, tuples are **immutable**. You cannot change the elements in a tuple.

Example of what you can't do:

~~contact_details_tuple[1] = 415~~ NO

Single Element Tuples

- To create a tuple with a single element, you have to include a final comma.
- A value in parentheses without a comma is not a tuple.

Examples:

- `my_tuple = (56, )`
- `my_tuple = 56,`
- But NOT `my_tuple = (56)`

tuple() function

- Another way to create a tuple is with the built-in function tuple().
- If the argument is a sequence (e.g. string, list or tuple), the result is a tuple with the elements of the sequence.

Examples:

- letters_tuple = tuple("nice")
- print(letters_tuple)
  - would print: ('n', 'i', 'c', 'e')

- names_list = ['Ann', 'Joy', 'Fred', 'Alex']
- names_tuple = tuple(names_list)
- print(names_tuple)
  - would print: ('Ann', 'Joy', 'Fred', 'Alex')

Accessing the elements in a tuple

- Each element in a tuple can be accessed using the index value inside a set of square brackets.

Examples:

- colours_tuple = ("white", "yellow ", "pink")

- print(colours_tuple[2])
  - Would print: pink.

- print(colours_tuple[-1])
  - Would also print: pink.

Iterating through a tuple using a for … in … loop

Example 1:

```
numbers_tuple = (6, 3, 0, 8)
for number in numbers_tuple:
    print(number)
```

would print the following:

```
6
3
0
8
```

Some inbuilt functions that can be used with tuples

- len(a_tuple)
  - returns the number of elements in a tuple
- min(a_tuple)
  - returns the minimum element in a tuple
- max(a_tuple)
  - returns the maximum element in a tuple
- sum(a_tuple)
  - returns the sum of the elements in the tuple (only for numbers).

The index() Method

- index(value)
  - returns the index of the first element in the tuple with the value specified in the parameter.
  - It throws an error if the value does not exist so a call to the index() method should always be preceded by a check to see if the value is in the tuple.

Example:

```
if "red" in colour_tuple:
    print("red is in Position", colour_tuple.index("red"))
```

Operators that can be used with tuples

+ (Concatenate)
* (Repeat)
in (Membership)

Examples:

```
numbers_tuple = (4, 7, 2)
bigger_tuple = numbers_tuple + (4, 7, 9)
numbers_tuple_repeated = numbers_tuple * 2
if 5 not in bigger_tuple:
    bigger_tuple += (5,)
```

Slicing Tuples

- Tuples can be sliced in the same way as strings and lists.

Example:

```
numbers_tuple = (4, 7, 2, 3, 9, 1, 0)
slice = numbers_tuple[1:6:2]
print(slice)
```

Would print:

```
(7, 3, 1)
```

Converting a Tuple into a List

- Tuples can be converted into a list using the list() function.

Example:

```
letters_tuple = ("A", "B", "C")
letters_list = list(letters_tuple)
```

Sorting a Tuple

- To sort a tuple you first need to convert the tuple into a list, sort the list, then convert the list back into a tuple afterwards.
- This is because tuples are **immutable**.

Example:

```
numbers_tuple = (21, 32, 2, 9, 20)
numbers_list = list(numbers_tuple)
numbers_list.sort()
numbers_tuple = tuple(numbers_list)
```

Multiple Assignment (Tuple Assignment)

- Assignment to more than one variable can be done on one line using ‘tuple assignment’.
- The left side is a tuple of variables, and the right side is a tuple of expressions.
- Each value is assigned to its respective variable.

Example:

```
name, age, phone = "Joe",  25, "028-6940-2768"
```

Returning more than one value from a function

- In theory, functions can only return one value.
- However, if the value returned is a tuple, the effect is the same as returning multiple values.

Example:

```
def main():
      name, age, upi = get_student_details()

def get_student_details()
    name = input("Enter name: ")
    age = int(input("Enter age: "))
    upi = input("Enter upi: ")
    return name, age, upi
```

Part B – File Processing

That completes the video on tuples

Now please watch Part B – File Processing