Python Tuples

Save As: tuples_example.py

REVIEW: Python Collection Data Types

- **[LIST]** – elements in the **list can change**. The list contains the **same data type**.
- TUPLE
- DICTIONARY

REVIEW: Examples for a [LIST] of values

- **Scores** in all courses taken

  scores = [76, 88, 95]

- Shopping List

  grocery = [“bread”, “carrots”, “cheese”]

Python Collection Data Types

- LIST
- **TUPLE** – elements in the tuple **cannot be changed**. The tuple can **contain mixed data types**.
- DICTIONARY

(Tuples) vs. [Lists]

(Tuples) are **not immutable**.

- **Cannot** Change, Add, or Remove Elements from the tuple.

[Lists] are **mutable**.

- Lists **can change**. Elements added, removed or changed.

Tuple = (uses parenthesis)

```
tupleEmpty = ( )                                #Empty Tuple

tupleNum = (1, 2, 3)                            #Tuple with Integers

tupleString = ("apple", "banana", "cherry")     #Tuple with Strings

tupleMix = (1, "Hello", 3.4)                    #Tuple with Mixed Data Types
```

Tuple Examples

```
vowels = (“a”, “e”, “i”, “o”, “u”)

tupleNum = (1, 2, 3, 4, 5)

tupleLang = (“Python”, “PHP”, “JavaScript”)
```

Tuple – Can have mixed data types

- A tuple can have any number of items and they may be of different types (integer, float, [list], ”string” etc.)

tupleMix = ("mouse", [8, 4, 6], (1, 2, 3))

Access Items in a Tuple – [Index number]

```
fruitTup = ("apple", "banana", "cherry")

print(fruitTup[0])
print(fruitTup[1])
print(fruitTup[2])
```

	0	1	2
fruitTup	fruitTup[0]	fruitTup[1]	fruitTup[2]
	apple	banana	cherry

Working with Tuples

```
tupleNum = (1, 2, 3, 4, 5)
tupleLang = (“Python”, “PHP”, “JavaScript”, “Java”, “Visual Basic”)
```

METHOD 1: Traverse over a Tuple using a for loop

```
fruitTup = ("apple", "banana", "cherry")

for item in fruitTup:
        print(item)
```

item	0	1	2
fruitTup	fruitTup[0]	fruitTup[1]	fruitTup[2]
	apple	banana	cherry

Check if Item Exists in a Tuple – in keyword

```
fruitTup = ("apple", "banana", "cherry")


if "apple" in fruitTup:
  print("Yes, apple is in the fruit tuple")
```

What is OUTPUT?

Slicing a Tuple – accessing certain elements in tuple

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

What is OUTPUT?

Tuple Operations

1. Creating a Tuple
2. Accessing a Tuple Element – use their index number (Negative Indexing)
3. Slicing a Tuple
4. Changing a Tuple Element - **cannot**, tuples are **not immutable (unchangeable).**
5. Adding to a Tuple - **cannot**, tuples are **not immutable (unchangeable).**
6. Deleting a Tuple Element - **cannot**, tuples are **not immutable (unchangeable).**
7. Deleting an entire tuple only.
8. Check if Item **in** Tuple
9. Determine Tuple length