

Name: _____

Partner: _____

Python Activity 40: Tuples

Sometimes you need an immutable list!

Learning Objectives

Students will be able to:

Content:

- Define a **tuple**
- Identify **elements** of a tuple
- Explain how to access individual elements of a tuple
- Explain how to replace an item in a tuple

Process:

- Write code that accesses the elements of a tuple
- Write code to convert sequences into a tuple
- Write code that edits a tuple – add, remove, and insert items

Prior Knowledge

- Python concepts: +=, append, slicing, mutability, list comprehensions, range

Critical Thinking Questions:

1. Examine the sample tuples below.

Sample Tuples in Python

```
0 >>> mylist = ["pixel", "jerry", 4]
1 >>> mytup = ("pixel", "jerry", 4)
2 >>> mylist == mytup
3 False
```

a. How many elements does the list named `mylist` contain? _____

b. How many elements might the **tuple** named `mytuple` contain? _____



c. How does the syntax for defining a tuple differ from the syntax for a list? _____

d. What element is at `mytuple[1]`? _____ And at `mytuple[2]`? _____

e. How do the elements of `mylist` and `mytuple` differ? _____



f. Why might the comparison on line 3 return `False`? _____

2. The following code continues from the previous examples:

```
4 >>> mylist.append(42)
5 >>> mytup.append(42)
6 AttributeError: 'tuple' object has no attribute 'append'
```

a. What is stored in `mylist` after line 4? _____

b. What is stored in `mytup` after line 5? _____



- c. What might the `AttributeError` on line 6 mean?

FYI: Tuples are **immutable**, and so they do not have any methods to modify the tuple itself. You'll need to construct a new tuple in order to change a tuple.

3. Examine the following code working with tuples:

```
0 >>> mytup = ("pixel", "jerry", 4)
1 >>> mytup += 72
2 TypeError: can only concatenate tuple (not 'int') to tuple
```

- a. What type of object is `mytup`? _____
- b. What type of object is `72`? _____
- c. How should we modify line 1 to append `72` to our tuple?

4. Examine the following code:

```
0 >>> mytup = ("pixel", "jerry", 4)
1 >>> mytup += (72,)
2 >>> mytup
3 ('pixel', 'sally', 4, 72)
```

- a. How does what is stored in `mytup` at line 2 differ from what it contains at line 0?

- b. What type of object is `mytup`? _____



- c. What type of object is `(72,)` ? _____



- d. Why might line 1 append an item to a *new* tuple, while `.append()` throws an error?



- e. Write a line of code to append the string “second” to the tuple, `mytup`:

5. Examine the following code working with tuples in interactive python:

```
0 >>> mytup = ("pixel", "jerry", 4)
1 >>> mytup[1] = "artie"
2 TypeError: 'tuple' object does not support item assignment
```

- a. What is the programmer trying to do on line 1?



- b. Write some lines of code to replace the second element of `mytup` with “artie”:

6. Examine the following code for creating new tuples:

```
0 >>> etup = ()
1 >>> len(etup)
```



a. How do we know that `etup` is a tuple?



b. What will the output of line 1 be?

7. Examine the following code with built-in functions:

```
0 >>> word = "Computerz"
1 >>> ch_mut = list(word)
2 >>> ch_mut
3 ['C', 'o', 'm', 'p', 'u', 't', 'e', 'r', 'z']
4 >>> ch_imm = tuple(ch_mut)
5 ('C', 'o', 'm', 'p', 'u', 't', 'e', 'r', 'z')
6 >>> list((1, 2, 3, "go", '!'))
```

a. What *type* of object is stored in `ch_mut`? How do you know?

b. What *type* of object is stored in `ch_imm`? How do you know?

c. What *type* of object is converted to a list on line 6? How do you know?



d. Explain what the ***tuple(..)*** built-in function does:



8. If you had to guess, what do you think each of these tuple operators do? (i.e., what might the code in the left do?)

Operator / Function	What the function/operator does
<code>single_tup = (5,)</code>	
<code>empty_tup = ()</code>	
<code>mytup[2]</code>	
<code>len(mytup)</code>	
<code>mytup += (5,)</code>	
<code>mytup[1:4:2]</code>	
<code>4 in mytup</code>	
<code>5 not in mytup</code>	
<code>[ele for ele in mytup]</code>	

Write some code that *iterates* through the list, `mylist`, and adds the items to a new tuple, `mytuple`:

```
mylist = range(0,100)
```

9. Examine the following code in interactive Python:

```
0 >>> a, b, c = 99, 77, 55
1 >>> a
2 99
3 >>> c
4 55
```

- a. What value is stored in the variable, `b`?



- b. Explain what is occurring on line 0:

- c. Write *one* line of code to assign 5 different values to 5 different variables:

FYI: Tuple assignment allows for a tuple of variables on the *lefthand* side of an assignment operator to be assigned the values of a tuple on the *righthand* side.

10. Examine the following code in interactive Python:

```
0 >>> pup_info = ["Pixel Puppy", 4, True]
1 >>> name, age, is_good_girl = pup_info
2 >>> age
3 4
```

- a. What is stored in `pup_info[1]`? How does it differ from the value of `age`?

- b. What is stored in `is_good_girl` after line 1? _____



- c. Explain what is occurring on line 1:

11. Examine the following code in interactive Python:

```
0 >>> a, b = 4, 7
1 >>> b, a = a, b
2 >>> a
3 7
```

a. What is stored in `b` after line 0?

b. What is stored in `b` after line 1?



c. Explain what is occurring on line 1:

Application Questions: Use the Python Interpreter to check your work

1. Write a python program that checks whether a tuple contains the value “winner” by iterating through the items of the tuple:

2. Write a python program that finds the repeated values inside of a tuple:

3. Given a **list of tuples**, write a program that creates a new list of tuples where each tuple contains all of the same values, except the last element of each tuple is replaced by the string “last”: