Name: ____________________________  Partner: ____________________________

Python Activity 19: Tuples

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

Process:
- Write code that prints a tuple
- Write code that edits a tuple – add, remove, and insert items

Prior Knowledge
- Python concepts from Activities 1-19

Folks, this is a brand new activity. If you encounter any issues/typos, please let Iris know!

Critical Thinking Questions:

FYI: A tuple is an immutable object that stores multiple data items in a contiguous manner. Each value stored in a tuple is called an element.

1. Examine the sample lists below.

<table>
<thead>
<tr>
<th>Sample Tuples in Python</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 &gt;&gt;&gt; mylist = [&quot;pixel&quot;, &quot;tally&quot;, 2]</td>
</tr>
<tr>
<td>1 &gt;&gt;&gt; mytup = (&quot;pixel&quot;, &quot;tally&quot;, 2)</td>
</tr>
<tr>
<td>2 &gt;&gt;&gt; mylist == mytup</td>
</tr>
<tr>
<td>3 False</td>
</tr>
</tbody>
</table>

a. How many elements does the list named mylist contain? ________________
b. How many elements does the tuple named mytup contain? ________________
c. What element is at mytup[1]? __________ And at mytup[2]? __________
d. How do the elements of mylist and mytup differ?

   ______________________________________________________
   e. Why does the comparison on line 2 return False?

   ______________________________________________________
f. How does the syntax for defining a tuple differ from the syntax for a list?

   ______________________________________________________
2. The following code continues from the previous example:

```python
>>> mytup1 = ("pixel", "tally", 2)
>>> mytup2 = "pixel", "tally", 2
>>> mytup1 == mytup2
True
```

a. Write a line of code to access the last element of `mytup1` with indexing: ____________

b. How do the elements of `mytup1` and `mytup2` differ? ______________________

c. How do lines 0 and line 1 differ? _________________________________________

d. Write a line of code to create a new tuple using the parentheses notation style:

```
```

e. Write some code that iterates through two tuples, `t1` and `t2`, and compares values at each index. It prints “Not Equal!” when it encounters two values that are different, and “Equal!” when the 2 values are equivalent:

```
```

3. The following code continues from the previous examples:

```python
>>> mylist.append(42)
>>> mytup.append(42)
AttributeError: 'tuple' object has no attribute 'append'
>>> mytup = "pixel", "tally", 2
>>> mytup += 72,
>>> mytup
('pixel', 'tally', 2, 72)
>>> mytup1 = ("pixel", "tally", 2)
>>> mytup2 = "pixel", "tally", 2
>>> mytup1 == mytup2
True
```

a. What is stored in `mylist` after line 0? _________________________________

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

c. What might the `AttributeError` on line 2 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.

4. Examine the following code:

```python
>>> mytup = "pixel", "tally", 2
>>> mytup += 72,
>>> mytup
('pixel', 'tally', 2, 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. Rewrite 72, in its alternative format:
_____________________

e. Why does line 1 append an item to a tuple, while .append(obj) throws an error?

5. Examine the following code, it is similar to the previous example:

```
0 >>> mytup = "pixel", "tally", 2
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?

________________________________________________________________________

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

______________________________________________________

6. Examine the following code, it is similar to the previous example:

```
0 >>> mytup = ("pixel", "tally", 2)
1 >>> mytup[1] = "wally"
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 “wally”:  
________________________________________________________________________
________________________________________________________________________
________________________________________________________________________
________________________________________________________________________

7. 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?  
________________________
c. Write some code that *iterates* through the list, `mylist`, and adds the items to a new tuple, `mytup`:

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

7. Examine the following code for creating new tuples:

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

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

b. Explain what is occurring on line 0:

   ```python
   a, b, c = 99, 77, 55
   ```

   (The values 99, 77, and 55 are assigned to the variables `a`, `b`, and `c`, respectively.)

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

   ```python
   a, b, c, d, e = some_values
   ```

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.

**Application Questions: Use the Python Interpreter to check your work**

1. Write some lines of code that use tuple assignment to swap two variables’ values:

   ```python
   a, b = 10, 20
   a, b = b, a
   print(a, b)
   ```

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

   ```python
   mytuple = ('first', 'second', 'winner', 'fourth')
   if 'winner' in mytuple:
       print('Tuple contains winner')
   ```
3. Write a python program that finds the repeated values inside of a tuple:

4. Given a list of tuples, write a program that changes the last element of each tuple to “last”: 