SelfAssessment81

October 25, 2015

1 Exercise 8.1

Write a function to concatenate two tuples and return the concatenation as a new tuple. However, if either tuple is empty, return a tuple containing nothing but a single 0 (a zero inside a tuple).

Answer appears after one blank page (so you don't peek).

Are you sure you're ready to peek?

2 Possible Solutions

The code will: 1. Take in two variables x and y 2. Make sure they're both tuples. 3. If they are, make sure they both have lengths > 0 4. If 2 but not 3 then `return` 0 5. If 2 & 3 are okay then concatenate them using the `+` operator. 6. If 2 isn't satisfied then `raise` a TypeError telling the user that the objects must be tuples.

```
In [2]: def tuple_cat(x,y):
            if isinstance(x,tuple) & isinstance(y,tuple):
                if len(x) == 0 or len(y) == 0:
                    return (0)
                else:
                    return x + y
            else:
                raise TypeError('Objects must be tuples.')

        # Tests:
        print tuple_cat((1,), (1,2,3))
        print tuple_cat((), (1,2,3))
        print tuple_cat('Wurlitzer', (1,2,3))
```

```
(1, 1, 2, 3)
0
```

```
        ---------------------------------------------------------------------------

        TypeError                                 Traceback (most recent call last)

        <ipython-input-2-f03f82f3ad92> in <module>()
         11 print tuple_cat((1,), (1,2,3))
         12 print tuple_cat((), (1,2,3))
    ---> 13 print tuple_cat('Wurlitzer', (1,2,3))


        <ipython-input-2-f03f82f3ad92> in tuple_cat(x, y)
          6             return x + y
          7     else:
    ----> 8         raise TypeError('Objects must be tuples.')
          9
         10 # Tests:


        TypeError: Objects must be tuples.
```

This all seems to work as expected. I added in the TypeError. We didn't ask for it in the question, but I wanted to again show you how to test for certain data types (`isinstance`) and show you an example of how to raise errors.