"CHECKPOINT"

Assessment 1 NEEDS REVIEW

1 **Python 1** 3 PTS

12 ATTEMPTS

Complete the function according to its docstring.

PYTHON3.6

```
def max_lists(list1, list2):
    x = []
    for i, j in zip(list1,list2):
        x.append(max(i,j))
    return x
    '''Compares the elements of list1, and list2 (assumed to be the same
    length),and returns the maximum value between the two, for each element.

    Parameters
    ----------
    list1, list2: list, list
        (Of the same length)

    Returns
    -------
    list
    '''
    pass
```

Pending...

HIDE TEST RESULTS

```
.
----------------------------------------------------------------------
Ran 1 test in 0.002s

OK
```

2 **Python 2** 3 PTS

1 ATTEMPT

Complete the function according to its docstring.

PYTHON3.6

```
def get_diagonal(mat):
    '''Given a matrix encoded as a 2 dimensional python list (i.e. a list of
    uniform length lists), returns a list containing all the values along the
    diagonal starting at the index 0, 0. (Assumes that the matrix is nonempty.)

    Parameters
    ----------
    mat: list of lists

    Returns
    -------
    list

    Example
    -------
    >>> mat = [[1, 2], [3, 4], [5, 6]]
    >>> get_diagonal(mat)
    [1, 4]
    '''
    diag_list = []

```

Pending...

HIDE TEST RESULTS

```
F
======================================================================
FAIL: test_get_diagonal (test_methods.TestPython1)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "/usr/src/app/test_methods.py", line 12, in test_get_diagonal
    self.assertEqual(result, [1, 6, 11])
AssertionError: None != [1, 6, 11]

----------------------------------------------------------------------
Ran 1 test in 0.000s

FAILED (failures=1)
```

3 **Python 3** 3 PTS

1 ATTEMPT

Complete the function according to its docstring.

PYTHON3.6

```python
def merge_dictionaries(d1, d2):
    '''Returns a new dictionary containing all the keys from d1 and d2 with
    their associated values. If a key is in both dictionaries, the new value is
    the sum of the two values from d1 and d2.

    Parameters
    ----------
    d1, d2: dictionary, dictionary

    Returns
    -------
    dictionary
    '''
    new_dict = dict()

    for k, v in d1.items():
        new_dict.update({k:sum(v)})

    for k, v in d2.items():
        new_dict.update({k:sum(v)})
```

Pending...

HIDE TEST RESULTS

```
E
======================================================================
ERROR: test_merge_dictionaries (test_methods.TestPython1)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "/usr/src/app/test_methods.py", line 12, in test_merge_dictionaries
    {"b": 2, "c": 5, "d": 10, "f": 6})
  File "/usr/src/app/main.py", line 17, in merge_dictionaries
    new_dict.update({k:sum(v)})
TypeError: 'int' object is not iterable

----------------------------------------------------------------------
Ran 1 test in 0.007s

FAILED (errors=1)
```

4 **Python/Linear Algebra** 3 PTS

1 ATTEMPT

Complete the function according to its docstring.

PYTHON3.6

```python
def matrix_multiplication(A, B):
    '''Return the matrix which is the product of matrix A and matrix B
    where A and B will be (a) integer valued (b) square matrices
    (c) of size n-by-n (d) encoded as lists of lists,  e.g.
    A = [[2, 3, 4], [6, 4, 2], [-1, 2, 0]] corresponds to the matrix
    | 2  3  4 |
    | 6  4  2 |
    |-1  2  0 |

    YOU MAY NOT USE NUMPY. Write your solution in straight python.

    Parameters
    ----------
    a, b: list of lists of integers, list of lists integers
        All inner/outer lists are of length n, so as to construct square
        matrices.

    Returns
    -------
    list of lists of integers
        The product of a matrix multiplication operation.
```

Pending...

HIDE TEST RESULTS

```
E
======================================================================
ERROR: test_matrix_multiplication (test_methods.TestPython1)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "/usr/src/app/test_methods.py", line 16, in test_matrix_multiplication
    self.assertEqual(main.matrix_multiplication(A, B), answer)
  File "/usr/src/app/main.py", line 26, in matrix_multiplication
    dot.append(A*B)
TypeError: can't multiply sequence by non-int of type 'list'

----------------------------------------------------------------------
Ran 1 test in 0.007s

FAILED (errors=1)
```

5 **Pandas 1** 3 PTS

9 ATTEMPTS

Complete the function according to its docstring.

PYTHON3.6

```python
import pandas as pd
import numpy as np
def pandas_add_increase_column(df):
    '''Adds a column to the DataFrame called 'Increase' which contains the
    amount that the median rent increased by from 2011 to 2014.

    Parameters
    ----------
    df: Pandas DataFrame

    Returns
    -------
    None
    '''
    df2 = df['2014'] - df['2011']
    df['Increase'] = df2 - df2.median()
    return df['Increase']
```

Pending...

HIDE TEST RESULTS

```
E
======================================================================
ERROR: test_pandas_add_increase_column (test_methods.TestPython1)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "/usr/local/lib/python3.6/site-packages/pandas/core/indexes/base.py", line 2657, in get_loc
    return self._engine.get_loc(key)
  File "pandas/_libs/index.pyx", line 108, in pandas._libs.index.IndexEngine.get_loc
  File "pandas/_libs/index.pyx", line 132, in pandas._libs.index.IndexEngine.get_loc
  File "pandas/_libs/hashtable_class_helper.pxi", line 1601, in
pandas._libs.hashtable.PyObjectHashTable.get_item
  File "pandas/_libs/hashtable_class_helper.pxi", line 1608, in
pandas._libs.hashtable.PyObjectHashTable.get_item
KeyError: '2014'

During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "/usr/src/app/test_methods.py", line 39, in test_pandas_add_increase_column
    a.pandas_add_increase_column(df)
  File "/usr/src/app/main.py", line 15, in pandas_add_increase_column
    df2 = df['2014'] - df['2011']
  File "/usr/local/lib/python3.6/site-packages/pandas/core/frame.py", line 2927, in __getitem__
    indexer = self.columns.get_loc(key)
  File "/usr/local/lib/python3.6/site-packages/pandas/core/indexes/base.py", line 2659, in get_loc
    return self._engine.get_loc(self._maybe_cast_indexer(key))
  File "pandas/_libs/index.pyx", line 108, in pandas._libs.index.IndexEngine.get_loc
  File "pandas/_libs/index.pyx", line 132, in pandas._libs.index.IndexEngine.get_loc
  File "pandas/_libs/hashtable_class_helper.pxi", line 1601, in
pandas._libs.hashtable.PyObjectHashTable.get_item
  File "pandas/_libs/hashtable_class_helper.pxi", line 1608, in
pandas._libs.hashtable.PyObjectHashTable.get_item
KeyError: '2014'

----------------------------------------------------------------------
Ran 1 test in 0.006s

FAILED (errors=1)
```

6 **Pandas 2** 3 PTS

3 ATTEMPTS

Complete the function according to its docstring.

PYTHON3.6

```python
def pandas_max_rent(df):
    '''Returns a new pandas DataFrame that contains every city and the highest
    median rent from that city for 2011 and 2014.

    Note that city names are not unique so the state is tracked as well.
    For example, Portland, ME and Portland, OR are recognized as different.

    Parameters
    ----------
    df: Pandas DataFrame

    Returns
    -------
    Pandas DataFrame:
        Containing the columns: City, State, med_2011, med_2014


    '''
    df['med_2011'] = df.groupy['State'].median()
    df['med_2014'] = df.groupy['State'].median()
    new_df = df['City','State', 'med_2011', 'med_2014' ]
```

Pending...

HIDE TEST RESULTS

```
E
======================================================================
ERROR: test_pandas_add_increase_column (test_methods.TestPython1)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "/usr/src/app/test_methods.py", line 45, in test_pandas_add_increase_column
    df = a.pandas_max_rent(df_in).reset_index()
  File "/usr/src/app/main.py", line 19, in pandas_max_rent
    df['med_2011'] = df.groupy['State'].median()
  File "/usr/local/lib/python3.6/site-packages/pandas/core/generic.py", line 5067, in __getattr__
    return object.__getattribute__(self, name)
AttributeError: 'DataFrame' object has no attribute 'groupy'

----------------------------------------------------------------------
Ran 1 test in 0.030s

FAILED (errors=1)
```

7 Git 3 PTS

1 ATTEMPT

a) What is Git, and why do software developers use it? (1 pt)

b) What is the 'staging area', or 'index' in git? (1 pt)

c) What does the command 'git commit' do? (1 pt)

a)Git is a version control application that allows developers to keep track of changes and branches of software. Very useful when multiple developers are working on same project as well as keep tracking of changes and reverting back to any version or branch. Good for testing/experimenting without messing with live production software.

b)when you use git add, you are adding a file/folder to a staging area to commit any changes that you have applied while working on the files. This allows for changes to be tracked between commits.

c)git commit saves the changes you've made to the local repo so what when you push to remote repo the changes are uploaded.

Pending...

8 **Python datatypes** 3 PTS

1 ATTEMPT

a) Please give an example of a mutable and immutable datatype. (2 pts)

b) What will be displayed when `b` is printed in the code below? (1 pt)

```
In[1] a = [1, 'a', [2, 3], 4.5]

In[2] b = a
```

show all

a) mutable: list, immutable: tuple or array
b) [-0.1, 'a', [2, 3], 4.5]

Pending...

LESTER RODRIGUEZ Pending

There is no activity on this challenge yet.

Add comment...

SEND