OutputComparisons

Version 1.0.0

Generated by Doxygen 1.8.13

Contents

1 Namespace Index

1.1 Packages

Here are the packages with brief descriptions (if available):

2 Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

object

output_comp.FileCompare 3

3 Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

output_comp.FileCompare
Class Object for Comparing Files 3

4 File Index

4.1 File List

Here is a list of all files with brief descriptions:

output_comp.py 7

5 Namespace Documentation

5.1 output_comp Namespace Reference

Python script to read output files and compare them.

Classes

- class FileCompare

  Class Object for Comparing Files.

5.1.1 Detailed Description

Python script to read output files and compare them.

Script for reading in output produced from an executable and comparing it to another output file. Useful for creating small unit tests for code validations or making evaluations on how changes in parameters change predicted outcomes. Thus, this script will be utilized for the purpose of performing sensitivity analyses.

Author

Austin Ladshaw

Date

05/09/2019

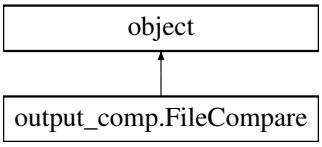
Copyright

6 Class Documentation

6.1 output_comp.FileCompare Class Reference

Class Object for Comparing Files.

Inheritance diagram for output_comp.FileCompare:



Public Member Functions

- def __init__ (self, gold, test)

  Initialization constructor must take in strings for the gold file and test file.
- def __str__ (self)

  Function to display results of the comparison to the console.
- def computeErrors (self)

  Function to read through each file line by line and compare each element and entry in each file with each other.
- def closeFiles (self)

  Function to close the open files (called automatically by computeErrors())

Public Attributes

- num_diff

  Variable to keep track of numerical differences in the files Values closest to zero represent smallest differences.
- str_diff

  Variable to keep track of string differences in the files Values closest to zero represent smallest differences.
- total_num

  Number of total numbers in both files.
- total_word

  Number of total words in both files.
- hasBeenRead
- lnum_gold

  Number of lines in gold file.
- lnum_test

  Number of lines in test file.
- gold_file
- test_file

6.1.1 Detailed Description

Class Object for Comparing Files.

This object will read in a pair of files line by line, parse the line into a set of words or numbers, and compare all items in the files to find differences. This is useful when you are doing unit testing (i.e., want to see if a simulation produced the same output after updating some code) or can be used for a large scale uncertainty or sensitivity analysis between a 'gold' standard simulation and a 'test' case.

Method:

(1) Iterate through each line in each file
(iterate through smallest)

(2) Split each line into individual strings
Parse on spaces

(3) Iterate (nested) over each individual string in a line
(loop over the smallest string list)

(4) Check each string for types
a) compare numbers/floats (try) and developed numeric different (sq. error)
b) compare strings/bools (except) using SequenceMatcher and produce ratio
c) Summate all errors keeping track of error types

(5) Continue through files and report levels of differences/similarities
Store results in the object
Use to determine how much simulation results have changed between runs

Notes:

read() gives all text in file

read(n) gives first n characters in file

readline() gives first line in file (∗∗Each instance reads the next line)

readline(n) gives nth line in the file

readlines() gives list of all lines in file

for line in file: loops over all lines in the file

string.split(ch) produces list of sub-strings parsed by the ch character
leaving ch blank defaults to splitting on blank spaces

use == to compare two strings

```
e.g.,  str1 == str2   -->   True if same, False if different
```

6.1.2 Constructor & Destructor Documentation

6.1.2.1 __init__()

```
def output_comp.FileCompare.__init__ (
              self,
              gold,
              test )
```

Initialization constructor must take in strings for the gold file and test file.

Parameters

gold	name of the file that you are comparing against
test	name of the file being tested for changes against the gold file

6.1.3 Member Function Documentation

6.1.3.1 __str__()

```
def output_comp.FileCompare.__str__ (
              self )
```

Function to display results of the comparison to the console.

6.1.3.2 computeErrors()

```
def output_comp.FileCompare.computeErrors (
              self )
```

Function to read through each file line by line and compare each element and entry in each file with each other.

If the two files are the same, then there will be no differences recorded between the two files.

6.1.3.3 closeFiles()

```
def output_comp.FileCompare.closeFiles (
              self )
```

Function to close the open files (called automatically by computeErrors())

6.1.4 Member Data Documentation

6.1.4.1 num_diff

output_comp.FileCompare.num_diff

Variable to keep track of numerical differences in the files Values closest to zero represent smallest differences.

6.1.4.2 str_diff

output_comp.FileCompare.str_diff

Variable to keep track of string differences in the files Values closest to zero represent smallest differences.

6.1.4.3 total_num

output_comp.FileCompare.total_num

Number of total numbers in both files.

6.1.4.4 total_word

output_comp.FileCompare.total_word

Number of total words in both files.

6.1.4.5 hasBeenRead

output_comp.FileCompare.hasBeenRead

6.1.4.6 lnum_gold

output_comp.FileCompare.lnum_gold

Number of lines in gold file.

6.1.4.7 lnum_test

output_comp.FileCompare.lnum_test

Number of lines in test file.

6.1.4.8 gold_file

`output_comp.FileCompare.gold_file`

6.1.4.9 test_file

`output_comp.FileCompare.test_file`

The documentation for this class was generated from the following file:

- output_comp.py

7 File Documentation

7.1 output_comp.py File Reference

Classes

- class output_comp.FileCompare
  Class Object for Comparing Files.

Namespaces

- output_comp
  Python script to read output files and compare them.

Index