**File Comparator Using Python**

|  |  |
| --- | --- |
| Prof. Rupali Tornekar, Geetai P. Charde  SY-Electronics and Telecommunication  ***Abstract — Now a day we can see that everyone is interested in coding, computer languages, etc. We write numbers of code and forgot the changes we make in their previous codes. This project presents an idea to compare two text files or code scripts which will help the user or programmer to see the similarities and differences in it.*** ***INTRODUCTION*** A file comparator is a tool which is used to compare two files containing strings and characters simultaneously so that to track the changes present in both files. ‘File comparators using Python’ utility is developed in Python programming language with the use of ‘difflib’ library and display its output in HTML format in required web browser. This type of file comparator is used to maintain and compare code sub-versioning.  **‘difflib’ library:** This module provides classes and functions for comparing sequences. It is useful in comparing files and generates information about differences in contents of file in various formats like HTML file and context, etc. ***Methodology/Experimental*** **Algorithm:**  **Algorithm**: Compare two files.  **Input:** Two files.  **Output**: Comparison between two files in new tab of web browser.  **Step 1**: Start.  **Step 2**: Import necessary library.  **Step 3**: Take a file path from user (as an input).  **Step 4**: Take the numbers of line to be compared as input from the user.  **Step 5:** Extract required lines from text file and store them into a list.  **Step 6:** Convert above list into concatenated text.  **Step 7:** Repeat step 3, 4, 5, 6 and 7 for second file.  **Step 8**: Pass the extracted text from both given file to html\_diff and make\_file function of difflib library and calculate the difference.  **Step 9**: Store the difference in variable and save the difference in html file (storing it to a particular location).  **Step 10**: Automatically display the content of a difference file in a new tab of a web browser using *open\_new\_tab (function)* of webbrowser library.  **Step 11:** Display the task completion message on a window  **Step 12:** Stop. ***Results and Discussions***   **Fig 1. Comparison of sample code** | * *class difflib.HtmlDiff*   This class can be used to create an HTML table (or a complete HTML file containing the table) showing a side by side, line by line comparison of text with inter-line and intra-line change highlights. The table can be generated in either full or contextual difference mode.   * *make\_file(fromlines, tolines, fromdesc='',*   *todesc='', context=False, numlines=5,*  *\*, charset='utf-8')*  Compares ‘fromlines’ and ‘tolines’ (lists of strings) and returns a string which is a complete HTML file containing a table showing line by line differences with inter-line and intra-line changes highlighted.  ‘fromdesc’ and ‘todesc’ are optional keyword arguments to specify from/to file column header strings (both default to an empty string). ‘context’ and ‘numlines’ are both optional keyword arguments. Set ‘context’ to ‘True’ when contextual differences are to be shown, else the default is ‘False’ to show the full files and numlines defaults to 5. When context is True numlines controls the number of context lines which surround the difference highlights. When context is False numlines controls the number of lines which are shown before a difference highlight when using the “next” hyperlinks (setting to zero would cause the “next” hyperlinks to place the next difference highlight at the top of the browser without any leading context).  Here, file1 is considered as the reference to the file 2. Newly added part in file 2 is shown by the green colour, changed part by yellow colour and deleted part by the red colour. Also, there are quick links as first change (f), next change (n) and top of page (t). ***Limitations***  1. It compares only text files. 2. It cannot compare multiple files at a time.  ***Future Scope*** This project can be built in such a way that it can compare table, pdf, images, and multiple files simultaneously. A web site or GUI for the comparison is also possible. ***Conclusion*** This file comparator can be efficiently used for comparison of two different text file. The output of comparator is an HTML file which is stored at particular location and automatically opened in a web-browser displaying similarities and differences. This comparison output can be used to visualise the changes done in the old and new file which is helpful to maintain the versions of a  code file.  ***References***   1. https://docs.python.org/3/library/difflib.html 2. https://docs.python.org/3/ 3. library/webbrowser.html 4. https://docs.python.org/3/library/ 5. https://cloud.smartdraw.com/login/ 6. https://en.wikipedia.org/wiki/File\_comparison 7. https://www.google.com/search?q=difflib   +library+in+python&oq=difflib+lib&aqs= chrome.0.0j69i57j0i10i22i30l3j69i60l3.  14816j0j7&sourceid=chrome&ie=UTF-8 |