[**python difflib模块讲解示例**](https://blog.csdn.net/Lockey23/article/details/77913855)

标签： [python模块](http://so.csdn.net/so/search/s.do?q=python%E6%A8%A1%E5%9D%97&t=blog) [difflib模块](http://so.csdn.net/so/search/s.do?q=difflib%E6%A8%A1%E5%9D%97&t=blog)

2017年09月09日 11:14:082079人阅读 [评论](javascript:void(0);)(0) [收藏](javascript:void(0);) [举报](javascript:void(0);)

https://csdnimg.cn/release/phoenix/images/category_icon.jpg 分类：

python（107） https://csdnimg.cn/release/phoenix/images/arrow_triangle%20_down.jpg

版权声明：本文为博主原创文章，未经博主允许不得转载。 https://blog.csdn.net/Lockey23/article/details/77913855

difflib模块提供的类和方法用来进行序列的差异化比较，它能够比对文件并生成差异结果文本或者html格式的差异化比较页面，如果需要比较目录的不同，可以使用filecmp模块。

**class difflib.SequenceMatcher**

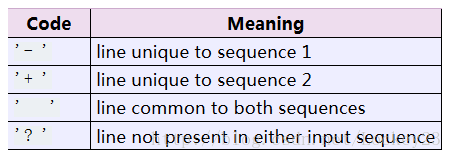
此类提供了比较任意可哈希类型序列对方法。此方法将寻找没有包含‘垃圾’元素的最大连续匹配序列。

通过对算法的复杂度比较，它由于原始的完形匹配算法，在最坏情况下有n的平方次运算，在最好情况下，具有线性的效率。

它具有自动垃圾启发式，可以将重复超过片段1%或者重复200次的字符作为垃圾来处理。可以通过将autojunk设置为false关闭该功能。

**class difflib.Differ**

此类比较的是文本行的差异并且产生适合人类阅读的差异结果或者增量结果，结果中各部分的表示如下：

   
**class difflib.HtmlDiff**

此类可以被用来创建HTML表格 (或者说包含表格的html文件) ，两边对应展示或者行对行的展示比对差异结果。

make\_file(fromlines, tolines [, fromdesc][, todesc][, context][, numlines])

make\_table(fromlines, tolines [, fromdesc][, todesc][, context][, numlines])

以上两个方法都可以用来生成包含一个内容为比对结果的表格的html文件，并且部分内容会高亮显示。

**difflib.context\_diff(a, b[, fromfile][, tofile][, fromfiledate][, tofiledate][, n][, lineterm])**

比较a与b(字符串列表)，并且返回一个差异文本行的生成器

示例：

>>> s1 = ['bacon\n', 'eggs\n', 'ham\n', 'guido\n']

>>> s2 = ['python\n', 'eggy\n', 'hamster\n', 'guido\n']

>>> for line in context\_diff(s1, s2, fromfile='before.py', tofile='after.py'):

... sys.stdout.write(line)

\*\*\* before.py

--- after.py

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\* 1,4 \*\*\*\*

! bacon

! eggs

! ham

guido

--- 1,4 ----

! python

! eggy

! hamster

guido

**difflib.get\_close\_matches(word, possibilities[, n][, cutoff])**

返回最大匹配结果的列表

示例：

>>> get\_close\_matches('appel', ['ape', 'apple', 'peach', 'puppy'])

['apple', 'ape']

>>> import keyword

>>> get\_close\_matches('wheel', keyword.kwlist)

['while']

>>> get\_close\_matches('apple', keyword.kwlist)

[]

>>> get\_close\_matches('accept', keyword.kwlist)

['except']

**difflib.ndiff(a, b[, linejunk][, charjunk])**

比较a与b(字符串列表)，返回一个Differ-style 的差异结果

示例：

>>> diff = ndiff('one\ntwo\nthree\n'.splitlines(1),

... 'ore\ntree\nemu\n'.splitlines(1))

>>> print ''.join(diff),

- one

? ^

+ ore

? ^

- two

- three

? -

+ tree

+ emu

**difflib.restore(sequence, which)**

返回一个由两个比对序列产生的结果

示例

>>> diff = ndiff('one\ntwo\nthree\n'.splitlines(1),

... 'ore\ntree\nemu\n'.splitlines(1))

>>> diff = list(diff) # materialize the generated delta into a list

>>> print ''.join(restore(diff, 1)),

one

two

three

>>> print ''.join(restore(diff, 2)),

ore

tree

emu

**difflib.unified\_diff(a, b[, fromfile][, tofile][, fromfiledate][, tofiledate][, n][, lineterm])**

比较a与b(字符串列表)，返回一个unified diff格式的差异结果.

示例：

>>> s1 = ['bacon\n', 'eggs\n', 'ham\n', 'guido\n']

>>> s2 = ['python\n', 'eggy\n', 'hamster\n', 'guido\n']

>>> for line in unified\_diff(s1, s2, fromfile='before.py', tofile='after.py'):

... sys.stdout.write(line)

--- before.py

+++ after.py

@@ -1,4 +1,4 @@

-bacon

-eggs

-ham

+python

+eggy

+hamster

guido

**实际应用示例**

比对两个文件，然后生成一个展示差异结果的HTML文件

#coding:utf-8

'''

file:difflibeg.py

date:2017/9/9 10:33

author:lockey

email:lockey@123.com

desc:diffle module learning and practising

'''

import difflib

hd = difflib.HtmlDiff()

loads = ''

with open('G:/python/note/day09/0907code/hostinfo/cpu.py','r') as load:

loads = load.readlines()

load.close()

mems = ''

with open('G:/python/note/day09/0907code/hostinfo/mem.py', 'r') as mem:

mems = mem.readlines()

mem.close()

with open('htmlout.html','a+') as fo:

fo.write(hd.make\_file(loads,mems))

fo.close()

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20
* 21
* 22
* 23

运行结果：



生成的html文件比对结果：   
