链滴

HTMLTestRunner 美化附源码。

作者：a929569603

原文链接：https://ld246.com/article/1576587485095

来源网站：链滴

某日，项目间隙，搞一下unittest框架，发现HTMLTestRunner报告模块，太丑，优化。
在网上COPY一个被优化过的版本感觉挺不错。不过没有我要的附截图功能，能忍吗？不能啊。
一个下午读源码，写逻辑修改。

使用必读说明：

1.支持python3+

########以下为本次更新，简书by.走在人生的路上z ###########

本次修复：通过按钮背景颜色BUG。

增加：增加详细信息内展示出错图片功能。

注意每条CASE必须在断言处加入异常处理后截图，截图返回照片地址。方可使用 以下示例。。

```
        count=0
        try:
            assert count != 0
        except:
            a= '\n' +'2001.png' #调用自己封装好的截图函数，返回图片地址名称，注意必须为小写png结尾
            self.fail(a)  #输出错误，错误信息为截图的地址。
```

效果展示

![收起效果]

![展开效果]

原文链接：HTMLTestRunner 美化附源码。

源码

```
#coding=utf-8
import datetime
import io
import sys
import time
import unittest
from xml.sax import saxutils
import re
import sys
"""
A TestRunner for use with the Python unit testing framework. It
generates a HTML report to show the result at a glance.
The simplest way to use this is to invoke its main method. E.g.
    import unittest
    import HTMLTestRunner
    ... define your tests ...
    if __name__ == '__main__':
        HTMLTestRunner.main()
For more customization options, instantiates a HTMLTestRunner object.
HTMLTestRunner is a counterpart to unittest's TextTestRunner. E.g.
    # output to a file
    fp = file('my_report.html', 'wb')
    runner = HTMLTestRunner.HTMLTestRunner(
                stream=fp,
                title='My unit test',
                description='This demonstrates the report output by HTMLTestRunner.'
                )
    # Use an external stylesheet.
    # See the Template_mixin class for more customizable options
```

原文链接：HTMLTestRunner 美化附源码。

```
    runner.STYLESHEET_TMPL = '<link rel="stylesheet" href="my_stylesheet.css" type="text/css
>'
    # run the test
    runner.run(my_test_suite)
----------------------------------------------------------------------
Copyright (c) 2004-2007, Wai Yip Tung
All rights reserved.
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:
* Redistributions of source code must retain the above copyright notice,
  this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright
  notice, this list of conditions and the following disclaimer in the
  documentation and/or other materials provided with the distribution.
* Neither the name Wai Yip Tung nor the names of its contributors may be
  used to endorse or promote products derived from this software without
  specific prior written permission.
THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS
IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED
TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER
OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
"""

# URL: http://tungwaiyip.info/software/HTMLTestRunner.html

__author__ = "Wai Yip Tung,  Findyou"
__version__ = "0.8.2.2"


"""
Change History
Version 0.8.2.1 -Findyou
* 改为支持python3
Version 0.8.2.1 -Findyou
* 支持中文，汉化
* 调整样式，美化（需要连入网络，使用的百度的Bootstrap.js）
* 增加 通过分类显示、测试人员、通过率的展示
* 优化“详细”与“收起”状态的变换
* 增加返回顶部的锚点
Version 0.8.2
* Show output inline instead of popup window (Viorel Lupu).
Version in 0.8.1
* Validated XHTML (Wolfgang Borgert).
* Added description of test classes and test cases.
Version in 0.8.0
* Define Template_mixin class for customization.
```

```
* Workaround a IE 6 bug that it does not treat <script> block as CDATA.
Version in 0.7.1
* Back port to Python 2.3 (Frank Horowitz).
* Fix missing scroll bars in detail log (Podi).
#########以上为网友#################
########以下为本次更新，简书by.走在人生的路上z  ############
本次修复：通过按钮背景颜色BUG。
增加：增加详细信息内展示出错图片功能。
注意每条CASE必须在断言处加入异常处理后截图，截图返回照片地址。方可使用 以下示例。。
        # try:
        #     assert count != 0
        # except:
        #     a='\n2001.png' #调用自己封装好的截图函数，返回图片地址名称，注意必须为小写png
尾
        #     self.fail(a)  #输出错误，错误信息为截图的地址。

"""

# TODO: color stderr
# TODO: simplify javascript using ,ore than 1 class in the class attribute?



# ------------------------------------------------------------------------
# The redirectors below are used to capture output during testing. Output
# sent to sys.stdout and sys.stderr are automatically captured. However
# in some cases sys.stdout is already cached before HTMLTestRunner is
# invoked (e.g. calling logging.basicConfig). In order to capture those
# output, use the redirectors for the cached stream.
#
# e.g.
#   >>> logging.basicConfig(stream=HTMLTestRunner.stdout_redirector)
#   >>>

class OutputRedirector(object):
    """ Wrapper to redirect stdout or stderr """
    def __init__(self, fp):
        self.fp = fp

    def write(self, s):
        self.fp.write(s)

    def writelines(self, lines):
        self.fp.writelines(lines)

    def flush(self):
        self.fp.flush()

stdout_redirector = OutputRedirector(sys.stdout)
stderr_redirector = OutputRedirector(sys.stderr)

# ----------------------------------------------------------------------
# Template
```

```
class Template_mixin(object):
    """
    Define a HTML template for report customerization and generation.
    Overall structure of an HTML report
    HTML
    +------------------------+
    |<html>                  |
    |  <head>                |
    |                        |
    |   STYLESHEET           |
    |   +----------------+   |
    |   |                |   |
    |   +----------------+   |
    |                        |
    |  </head>               |
    |                        |
    |  <body>                |
    |                        |
    |   HEADING              |
    |   +----------------+   |
    |   |                |   |
    |   +----------------+   |
    |                        |
    |   REPORT               |
    |   +----------------+   |
    |   |                |   |
    |   +----------------+   |
    |                        |
    |   ENDING               |
    |   +----------------+   |
    |   |                |   |
    |   +----------------+   |
    |                        |
    |  </body>               |
    |</html>                 |
    +------------------------+
    """

    STATUS = {
    0: '通过',
    1: '失败',
    2: '错误',
    }

    DEFAULT_TITLE = '单元测试报告'
    DEFAULT_DESCRIPTION = ''
    DEFAULT_TESTER='简书by:走在人生的路上z'

    # ------------------------------------------------------------------------
    # HTML Template

    HTML_TMPL = r"""<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml
/DTD/xhtml1-strict.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>%(title)s</title>
    <meta name="generator" content="%(generator)s"/>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
    <link href="http://libs.baidu.com/bootstrap/3.0.3/css/bootstrap.min.css" rel="stylesheet">
    <script src="http://libs.baidu.com/jquery/2.0.0/jquery.min.js"></script>
    <script src="http://libs.baidu.com/bootstrap/3.0.3/js/bootstrap.min.js"></script>
    %(stylesheet)s
</head>
<body >
<script language="javascript" type="text/javascript">
output_list = Array();
/*level 调整增加只显示通过用例的分类 --Findyou
0:Summary //all hiddenRow
1:Failed  //pt hiddenRow, ft none
2:Pass    //pt none, ft hiddenRow
3:All     //pt none, ft none
*/
function showCase(level) {
    trs = document.getElementsByTagName("tr");
    for (var i = 0; i < trs.length; i++) {
        tr = trs[i];
        id = tr.id;
        if (id.substr(0,2) == 'ft') {
            if (level == 2 || level == 0 ) {
                tr.className = 'hiddenRow';
            }
            else {
                tr.className = '';
            }
        }
        if (id.substr(0,2) == 'pt') {
            if (level < 2) {
                tr.className = 'hiddenRow';
            }
            else {
                tr.className = '';
            }
        }
    }
    //加入【详细】切换文字变化 --Findyou
    detail_class=document.getElementsByClassName('detail');
    //console.log(detail_class.length)
    if (level == 3) {
        for (var i = 0; i < detail_class.length; i++){
            detail_class[i].innerHTML="收起"
        }
    }
    else{
            for (var i = 0; i < detail_class.length; i++){
            detail_class[i].innerHTML="详情"
        }
    }
```

```
}
function showClassDetail(cid, count) {
    var id_list = Array(count);
    var toHide = 1;
    for (var i = 0; i < count; i++) {
        //ID修改 点 为 下划线 -Findyou
        tid0 = 't' + cid.substr(1) + '_' + (i+1);
        tid = 'f' + tid0;
        tr = document.getElementById(tid);
        if (!tr) {
            tid = 'p' + tid0;
            tr = document.getElementById(tid);
        }
        id_list[i] = tid;
        if (tr.className) {
            toHide = 0;
        }
    }
    for (var i = 0; i < count; i++) {
        tid = id_list[i];
        //修改点击无法收起的BUG，加入【详细】切换文字变化 --Findyou
        if (toHide) {
            document.getElementById(tid).className = 'hiddenRow';
            document.getElementById(cid).innerText = "详情"
        }
        else {
            document.getElementById(tid).className = '';
            document.getElementById(cid).innerText = "收起"
        }
    }
}
function html_escape(s) {
    s = s.replace(/&/g,'&');
    s = s.replace(/</g,'<');
    s = s.replace(/>/g,'>');
    return s;
}
</script>
%(heading)s
%(report)s
%(ending)s
</body>
</html>
"""
    # variables: (title, generator, stylesheet, heading, report, ending)


    # ------------------------------------------------------------------------
    # Stylesheet
    #
    # alternatively use a <link> for external style sheet, e.g.
    #   <link rel="stylesheet" href="$url" type="text/css">

    STYLESHEET_TMPL = """
```

```
<style type="text/css" media="screen">
body        { font-family: Microsoft YaHei,Tahoma,arial,helvetica,sans-serif;padding: 20px; font-s
ze: 80%; }
table       { font-size: 100%; }
/* -- heading ------------------------------------------------------------------ */
.heading {
    margin-top: 0ex;
    margin-bottom: 1ex;
}
.heading .description {
    margin-top: 4ex;
    margin-bottom: 6ex;
}
/* -- report ------------------------------------------------------------------- */
#total_row  { font-weight: bold; }
.passCase   { color: #5cb85c; }
.failCase   { color: #d9534f; font-weight: bold; }
.errorCase  { color: #f0ad4e; font-weight: bold; }
.hiddenRow  { display: none; }
.testcase   { margin-left: 2em; }
.btn-danger0 {
    color: #fff;
    background-color: #5cb85c;
    border-color: #5cb85c;
    }
.btn-danger1 {
    color: #fff;
    background-color: #d9534f;
    border-color: #d9534f;
    }
</style>
"""
##d9534f
    # ------------------------------------------------------------------------
    # Heading
    #

    HEADING_TMPL = """<div class='heading'>
<h1 style="font-family: Microsoft YaHei">%(title)s</h1>
%(parameters)s
<p class='description'>%(description)s</p>
</div>
""" # variables: (title, parameters, description)

    HEADING_ATTRIBUTE_TMPL = """<p class='attribute'><strong>%(name)s : </strong> %(va
ue)s</p>
""" # variables: (name, value)



    # ------------------------------------------------------------------------
    # Report
    #
    # 汉化,加美化效果 --Findyou
```

原文链接：HTMLTestRunner 美化附源码。

```
    REPORT_TMPL = """
<p id='show_detail_line'>
<a class="btn btn-primary" href='javascript:showCase(0)'>概要 (%(passrate)s )</a>
<a class="btn btn-danger" href='javascript:showCase(1)'>失败(%(fail)s )</a>
<a class="btn btn-success" href='javascript:showCase(2)'>通过(%(Pass)s )</a>
<a class="btn btn-info" href='javascript:showCase(3)'>所有(%(count)s )</a>
</p>
<table id='result_table' class="table table-condensed table-bordered table-hover">
<colgroup>
<col align='left' />
<col align='right' />
<col align='right' />
<col align='right' />
<col align='right' />
<col align='right' />
</colgroup>
<tr id='header_row' class="text-center success" style="font-weight: bold;font-size: 14px;">
    <td>用例集/测试用例</td>
    <td>总计</td>
    <td>通过</td>
    <td>失败</td>
    <td>错误</td>
    <td>详细</td>
    <td>截图</td>
</tr>
%(test_list)s
<tr id='total_row' class="text-center active">
    <td>总计</td>
    <td>%(count)s</td>
    <td>%(Pass)s</td>
    <td>%(fail)s</td>
    <td>%(error)s</td>
    <td>通过率：%(passrate)s</td>
    <td id='ztc'>测试报告</td>
</tr>
</table>
""" # variables: (test_list, count, Pass, fail, error ,passrate)

    REPORT_CLASS_TMPL = r"""
<tr class='%(style)s warning'>
    <td>%(desc)s</td>
    <td class="text-center">%(count)s</td>
    <td class="text-center">%(Pass)s</td>
    <td class="text-center">%(fail)s</td>
    <td class="text-center">%(error)s</td>
    <td class="text-center"><a href="javascript:showClassDetail('%(cid)s',%(count)s)" class="d
tail" id='%(cid)s'>详情</a></td>
    <td class="text-center"><a href="javascript:showClassDetail('%(cid)s',%(count)s)" class="d
tail" id='%(cid)s'>截图详情</a></td>
</tr>
""" # variables: (style, desc, count, Pass, fail, error, cid)

    #失败 的样式，去掉原来JS效果，美化展示效果  -Findyou
    REPORT_TEST_WITH_OUTPUT_TMPL = r"""
```

原文链接：HTMLTestRunner 美化附源码。

```
<tr id='%(tid)s' class='%(Class)s'>
    <td class='%(style)s'><div class='testcase'>%(desc)s</div></td>
    <td colspan='5' align='center'>
    <!--默认收起错误信息 -Findyou -->
    <button id='btn_%(tid)s' type="button" name='%(status)s' class="btn btn-danger%(cgsb)s
btn-xs collapsed" data-toggle="collapse" data-target='#div_%(tid)s'>%(status)s</button>
    <div id='div_%(tid)s' class="collapse">
    <!-- 默认展开错误信息 -Findyou
    <button id='btn_%(tid)s' type="button"  class="btn btn-danger btn-xs" data-toggle="coll
pse" data-target='#div_%(tid)s'>%(status)s</button>
    <div id='div_%(tid)s' class="collapse in"> -->
    <pre>
    %(script)s
    <img src='%(jietu)s'  alt="此CASE已通过，或未通过且未截图。"height=251 width=400 />
    </pre>
    </div>
    </td>
    <td><a/ href='%(jietu)s'>%(jietu)s</a></td>
</tr>
""" # variables: (tid, Class, style, desc, status)

    # 通过 的样式，加标签效果  -Findyou
    REPORT_TEST_NO_OUTPUT_TMPL = r"""
<tr id='%(tid)s' class='%(Class)s'>
    <td class='%(style)s'><div class='testcase'>%(desc)s</div></td>
    <td colspan='5' align='center'><span class="label label-success success">%(status)s</spa
></td>
</tr>
""" # variables: (tid, Class, style, desc, status)

    REPORT_TEST_OUTPUT_TMPL = r"""
%(id)s: %(output)s
""" # variables: (id, output)

    # ------------------------------------------------------------------------
    # ENDING
    #
    # 增加返回顶部按钮  --Findyou
    ENDING_TMPL = """<div id='ending'> </div>
    <div style=" position:fixed;right:50px; bottom:30px; width:20px; height:20px;cursor:pointer
>
    <a href="#"><span class="glyphicon glyphicon-eject" style = "font-size:30px;" aria-hidde
="true">
    </span></a></div>
    """

# -------------------- The end of the Template class -------------------


TestResult = unittest.TestResult

class _TestResult(TestResult):
    # note: _TestResult is a pure representation of results.
    # It lacks the output and reporting ability compares to unittest._TextTestResult.
```

```
def __init__(self, verbosity=1):
    TestResult.__init__(self)
    self.stdout0 = None
    self.stderr0 = None
    self.success_count = 0
    self.failure_count = 0
    self.error_count = 0
    self.verbosity = verbosity

    # result is a list of result in 4 tuple
    # (
    #   result code (0: success; 1: fail; 2: error),
    #   TestCase object,
    #   Test output (byte string),
    #   stack trace,
    # )
    self.result = []
    #增加一个测试通过率 --Findyou
    self.passrate=float(0)


def startTest(self, test):
    TestResult.startTest(self, test)
    # just one buffer for both stdout and stderr
    self.outputBuffer = io.StringIO()
    stdout_redirector.fp = self.outputBuffer
    stderr_redirector.fp = self.outputBuffer
    self.stdout0 = sys.stdout
    self.stderr0 = sys.stderr
    sys.stdout = stdout_redirector
    sys.stderr = stderr_redirector


def complete_output(self):
    """
    Disconnect output redirection and return buffer.
    Safe to call multiple times.
    """
    if self.stdout0:
        sys.stdout = self.stdout0
        sys.stderr = self.stderr0
        self.stdout0 = None
        self.stderr0 = None
    return self.outputBuffer.getvalue()


def stopTest(self, test):
    # Usually one of addSuccess, addError or addFailure would have been called.
    # But there are some path in unittest that would bypass this.
    # We must disconnect stdout in stopTest(), which is guaranteed to be called.
    self.complete_output()
```

原文链接：HTMLTestRunner 美化附源码。

```
    def addSuccess(self, test):
        self.success_count += 1
        TestResult.addSuccess(self, test)
        output = self.complete_output()
        self.result.append((0, test, output, ''))
        if self.verbosity > 1:
            sys.stderr.write('ok ')
            sys.stderr.write(str(test))
            sys.stderr.write('\n')
        else:
            sys.stderr.write('.')

    def addError(self, test, err):
        self.error_count += 1
        TestResult.addError(self, test, err)
        _, _exc_str = self.errors[-1]
        output = self.complete_output()
        self.result.append((2, test, output, _exc_str))
        if self.verbosity > 1:
            sys.stderr.write('E  ')
            sys.stderr.write(str(test))
            sys.stderr.write('\n')
        else:
            sys.stderr.write('E')

    def addFailure(self, test, err):
        self.failure_count += 1
        TestResult.addFailure(self, test, err)
        _, _exc_str = self.failures[-1]
        output = self.complete_output()
        self.result.append((1, test, output, _exc_str))
        if self.verbosity > 1:
            sys.stderr.write('F  ')
            sys.stderr.write(str(test))
            sys.stderr.write('\n')
        else:
            sys.stderr.write('F')


class HTMLTestRunner(Template_mixin):
    """
    """
    def __init__(self, stream=sys.stdout, verbosity=1,title=None,description=None,tester=None)

        self.stream = stream
        self.verbosity = verbosity
        if title is None:
            self.title = self.DEFAULT_TITLE
        else:
            self.title = title
        if description is None:
            self.description = self.DEFAULT_DESCRIPTION
        else:
            self.description = description
```

原文链接：HTMLTestRunner 美化附源码。

```
        if tester is None:
            self.tester = self.DEFAULT_TESTER
        else:
            self.tester = tester

        self.startTime = datetime.datetime.now()


    def run(self, test):
        "Run the given test case or test suite."
        result = _TestResult(self.verbosity)
        test(result)
        self.stopTime = datetime.datetime.now()
        self.generateReport(test, result)
        print('\nTime Elapsed: %s' % (self.stopTime-self.startTime))
        # file=sys.stderr
        return result


    def sortResult(self, result_list):
        # unittest does not seems to run in any particular order.
        # Here at least we want to group them together by class.
        rmap = {}
        classes = []
        for n,t,o,e in result_list:
            cls = t.__class__
            if cls not in rmap:
                rmap[cls] = []
                classes.append(cls)
            rmap[cls].append((n,t,o,e))
        r = [(cls, rmap[cls]) for cls in classes]
        return r

    #替换测试结果status为通过率 --Findyou
    def getReportAttributes(self, result):
        """
        Return report attributes as a list of (name, value).
        Override this to add custom attributes.
        """
        startTime = str(self.startTime)[:19]
        duration = str(self.stopTime - self.startTime)
        status = []
        status.append('共 %s' % (result.success_count + result.failure_count + result.error_count))
        if result.success_count: status.append('通过 %s'    % result.success_count)
        if result.failure_count: status.append('失败 %s' % result.failure_count)
        if result.error_count:   status.append('错误 %s'   % result.error_count  )
        if status:
            status = ',  '.join(status)
            self.passrate = str("%.2f%%" % (float(result.success_count) / float(result.success_count
+ result.failure_count + result.error_count) * 100))
        else:
            status = 'none'
        return [
            ('测试人员', self.tester),
```

```
        ('开始时间',startTime),
        ('合计耗时',duration),
        ('测试结果',status + "，通过率= "+self.passrate),
    ]


def generateReport(self, test, result):
    report_attrs = self.getReportAttributes(result)#返回通过 开始时间 合计耗时和通过 率。
    generator = 'HTMLTestRunner %s' % __version__
    stylesheet = self._generate_stylesheet()
    heading = self._generate_heading(report_attrs)
    # print("heading")
    # print(heading)
    report = self._generate_report(result)
    # print('report')
    # print(report)
    ending = self._generate_ending()
    output = self.HTML_TMPL % dict(
        title = saxutils.escape(self.title),
        generator = generator,
        stylesheet = stylesheet,
        heading = heading,
        report = report,
        ending = ending,
    )
    # print(output.encode('utf8'))
    self.stream.write(output.encode('utf8'))


def _generate_stylesheet(self):
    return self.STYLESHEET_TMPL

#增加Tester显示 -Findyou
def _generate_heading(self, report_attrs):
    a_lines = []
    for name, value in report_attrs:
        line = self.HEADING_ATTRIBUTE_TMPL % dict(
                name = saxutils.escape(name),
                value = saxutils.escape(value),
            )
        a_lines.append(line)
    heading = self.HEADING_TMPL % dict(
        title = saxutils.escape(self.title),
        parameters = ''.join(a_lines),
        description = saxutils.escape(self.description),
        tester= saxutils.escape(self.tester),
    )
    return heading

#生成报告  --Findyou添加注释
def _generate_report(self, result):
    rows = []
    sortedResult = self.sortResult(result.result)
    for cid, (cls, cls_results) in enumerate(sortedResult):
```

```
            # subtotal for a class
            np = nf = ne = 0
            for n,t,o,e in cls_results:
                if n == 0: np += 1
                elif n == 1: nf += 1
                else: ne += 1

            # format class description
            if cls.__module__ == "__main__":
                name = cls.__name__
            else:
                name = "%s.%s" % (cls.__module__, cls.__name__)
            doc = cls.__doc__ and cls.__doc__.split("\n")[0] or ""
            desc = doc and '%s: %s' % (name, doc) or name

            row = self.REPORT_CLASS_TMPL % dict(
                style = ne > 0 and 'errorClass' or nf > 0 and 'failClass' or 'passClass',
                desc = desc,
                count = np+nf+ne,
                Pass = np,
                fail = nf,
                error = ne,
                cid = 'c%s' % (cid+1),
            )
            rows.append(row)

            for tid, (n,t,o,e) in enumerate(cls_results):
                self._generate_report_test(rows, cid, tid, n, t, o, e)

        report = self.REPORT_TMPL % dict(
            test_list = ''.join(rows),
            count = str(result.success_count+result.failure_count+result.error_count),
            Pass = str(result.success_count),
            fail = str(result.failure_count),
            error = str(result.error_count),
            passrate =self.passrate,
        )

        return report


    def _generate_report_test(self, rows, cid, tid, n, t, o, e):
        # e.g. 'pt1.1', 'ft1.1', etc
        has_output = bool(o or e)
        # ID修改点为下划线,支持Bootstrap折叠展开特效 - Findyou
        tid = (n == 0 and 'p' or 'f') + 't%s_%s' % (cid+1,tid+1)
        name = t.id().split('.')[-1]
        doc = t.shortDescription() or ""
        desc = doc and ('%s: %s' % (name, doc)) or name
        tmpl = has_output and self.REPORT_TEST_WITH_OUTPUT_TMPL or self.REPORT_TEST_N
_OUTPUT_TMPL


        # utf-8 支持中文 - Findyou
```

```
            # o and e should be byte string because they are collected from stdout and stderr?
            if isinstance(o, str):
                # TODO: some problem with 'string_escape': it escape \n and mess up formating
                # uo = unicode(o.encode('string_escape'))
                # uo = o.decode('latin-1')
                uo = o
            else:
                uo = o
            if isinstance(e, str):
                # TODO: some problem with 'string_escape': it escape \n and mess up formating
                # ue = unicode(e.encode('string_escape'))
                # ue = e.decode('latin-1')
                ue = e
            else:
                ue = e

            script = self.REPORT_TEST_OUTPUT_TMPL % dict(
                id = tid,
                output = saxutils.escape(uo+ue),
            )
            list1=[]
            for a in re.finditer('\n',script):
                list1.append(a.span())#添加\n位置
        try:
                num=list1[-3]
        except:
            num=list1[-2]
            num=num[0]
            list1=[]
            if script[num:][-5:-2]=="png":
                jietu=script[num:]
            else:
                jietu=''
            row = tmpl % dict(
                tid = tid,
                Class = (n == 0 and 'hiddenRow' or 'none'),
                style = n == 2 and 'errorCase' or (n == 1 and 'failCase' or 'passCase'),
                desc = desc,
                script = script,
                status = self.STATUS[n],
                jietu =jietu,
                cgsb=n,
            )

            rows.append(row)
            if not has_output:
                return

        def _generate_ending(self):
            return self.ENDING_TMPL


##############################################################################
#######
```

原文链接：HTMLTestRunner 美化附源码。

```
# Facilities for running tests from the command line
##############################################################################

# Note: Reuse unittest.TestProgram to launch test. In the future we may
# build our own launcher to support more specific command line
# parameters like test title, CSS, etc.
class TestProgram(unittest.TestProgram):
    """
    A variation of the unittest.TestProgram. Please refer to the base
    class for command line parameters.
    """
    def runTests(self):
        # Pick HTMLTestRunner as the default test runner.
        # base class's testRunner parameter is not useful because it means
        # we have to instantiate HTMLTestRunner before we know self.verbosity.
        if self.testRunner is None:
            self.testRunner = HTMLTestRunner(verbosity=self.verbosity)
        unittest.TestProgram.runTests(self)

main = TestProgram

##############################################################################
# Executing this module from the command line
##############################################################################

if __name__ == "__main__":
    main(module=None)
```