WriteHTML()

(mPDF >= 1.0)

WriteHTML — Write HTML code to the document

Description

void **WriteHTML** (string *$html* [, int *$mode* [, boolean *$initialise* [, boolean *$close*]]])

Write *html* code to the document.

Note: Prior to mPDF 4.2 a fatal error was caused if *html* was passed as a **NULL** value, **FALSE** or an undefined variable.

Parameters

html

UTF-8 encoded HTML code to write to the document.

mode

Controls what parts of the *html* code is parsed.

Values
0 - Parses a whole *html* document
1 - Parses the *html* as styles and stylesheets only
2 - Parses the *html* as output elements only
3 - (For internal use only - parses the *html* code without writing to document)
4 - (For internal use only - writes the *html* code to a buffer)
DEFAULT: 0

Mode #0 (**DEFAULT**)
Metadata:
- title is read from <title>...</title> tags
- subject, keywords and author are read from <meta ...

Charset:
- if *$allow_charset_conversion* = **TRUE** and a charset= statement is present, mPDF will attempt to convert all the following HTML input from the specified charset to UTF-8

CSS styles:
- any CSS found between <style>...</style> tags
- stylesheets specified by @import url(*.css)
- stylesheets specified by <link rel="stylesheet" href=""
NB Stylesheets with media="all" or media ="screen" will always be parsed.
The variable *$disablePrintCSS* will determine whether stylesheets media="print" are parsed or not.

Anything between <style> tags is then discarded.

If <body> tags are found, all *html* outside these tags are discarded, and the rest is parsed as content for the document.
If no <body> tags are found, all remaining *html* is parsed as content.

Mode #1
The html input is only parsed as CSS style information only.
The code does not have to be surrounded by <style> tags, so you can pass the contents of a stylesheet directly - see Example #1.

Mode #2
If <body> tags are found, all *html* outside these tags are discarded, and the rest is parsed as content for the document.

If no <body> tags are found, all *html* is parsed as content.
Prior to mPDF 4.2 the default CSS was not parsed when using *mode* #2

initialise

Set **TRUE** or **FALSE** to determine whether to initialise all buffers, starting all HTML elements from new. See example 2 for use. You must start with a WriteHTML() that calls *initialise*=**TRUE**
DEFAULT: **TRUE**

close

Set **TRUE** or **FALSE** to specify whether all HTML elements are closed, and buffers cleared. See example 2 for use. You must end with a WriteHTML() that calls *close*=**TRUE**
DEFAULT: **TRUE**

Changelog

Version	Description
2.0	Using WriteHTML without the *mode* parameter no longer clears any CSS styles already imported.
2.1	Parameters *initialise* and *close* introduced.
4.2	Accepts **NULL** string as paramter without error. Parses default CSS when using *mode* as 2

Examples

Example #1

```
<?php
$mpdf=new mPDF();

$stylesheet = file_get_contents('style.css');
$mpdf->WriteHTML($stylesheet,1);
$mpdf->WriteHTML('<p>Hallo World</p>', 2);

$mpdf->Output();
?>
```

Example #2

```
// You can write parts of HTML elements by using the initialise and close parameters:

$mpdf->WriteHTML('<p>This is the beginning...', 2, true, false);
$mpdf->WriteHTML('...this is the middle...', 2, false, false);
$mpdf->WriteHTML('...and this is the end</p>', 2, false, true);
```

See Also

- allow_charset_conversion - attempts to read any charset declaration in the HTML code
- disablePrintCSS - prevents stylesheets set for print media being parsed
- ignore_invalid_utf8 - prevents mPDF from failing if text contains invalid UTF-8 characters
- charset_in - specify the input text character set if not UTF-8
- biDirectional - specify whether mPDF should test for **RTL** text
- allow_html_optional_endtags -specify whether mPDF should try to accommodate optional HTML endtags
- restoreBlockPagebreaks - keep current HTML tags/CSS styles active when forcing a page-break or formfeed

Printed on Wed 05 Aug 2015 12:18:10 GMT +0100 (DST)