



## ***Rocky Enterprise Linux 9.2 Manual Pages on command 'HTML::Filter.3pm'***

**C:\>man HTML::Filter.3pm**

HTML::Filter(3pm)      User Contributed Perl Documentation      HTML::Filter(3pm)

### NAME

HTML::Filter - Filter HTML text through the parser

### NOTE

This module is deprecated. The "HTML::Parser" now provides the functionality of "HTML::Filter" much more efficiently with the "default" handler.

### SYNOPSIS

```
require HTML::Filter;

$p = HTML::Filter->new->parse_file("index.html");
```

### DESCRIPTION

"HTML::Filter" is an HTML parser that by default prints the original text of each HTML element (a slow version of cat(1) basically). The callback methods may be overridden to modify the filtering for some HTML elements and you can override output() method which is called to print the HTML text.

"HTML::Filter" is a subclass of "HTML::Parser". This means that the document should be given to the parser by calling the \$p->parse() or \$p->parse\_file() methods.

## EXAMPLES

The first example is a filter that will remove all comments from an HTML file.

This is achieved by simply overriding the comment method to do nothing.

```
package CommentStripper;
require HTML::Filter;
@ISA=qw(HTML::Filter);
sub comment { } # ignore comments
```

The second example shows a filter that will remove any <TABLE>s found in the HTML file. We specialize the start() and end() methods to count table tags and then make output not happen when inside a table.

```
package TableStripper;
require HTML::Filter;
@ISA=qw(HTML::Filter);
sub start
{
    my $self = shift;
    $self->{table_seen}++ if $_[0] eq "table";
    $self->SUPER::start(@_);
}

sub end
{
    my $self = shift;
    $self->SUPER::end(@_);
    $self->{table_seen}-- if $_[0] eq "table";
}

sub output
{
    my $self = shift;
```

```
unless ($self->{table_seen}) {  
    $self->SUPER::output(@_);  
}  
}
```

If you want to collect the parsed text internally you might want to do something like this:

```
package FilterIntoString;  
require HTML::Filter;  
@ISA=qw(HTML::Filter);  
sub output { push(@{$_[0]->{fhtml}}, $_[1]) }  
sub filtered_html { join("", @{$_[0]->{fhtml}}) }
```

## SEE ALSO

HTML::Parser

## COPYRIGHT

Copyright 1997-1999 Gisle Aas.

This library is free software; you can redistribute it and/or modify it under the same terms as Perl itself.

perl v5.30.0

2020-02-18

HTML::Filter(3pm)