NAME

CGI::Pretty - module to produce nicely formatted HTML code

SYNOPSIS

```
use CGI::Pretty qw( :html3 );


# Print a table with a single data element
print table( TR( td( "foo" ) ) );
```

DESCRIPTION

CGI::Pretty is a module that derives from CGI. It's sole function is to allow users of CGI to output nicely formatted HTML code.

When using the CGI module, the following code: print table(TR(td("foo")));

produces the following output: <TABLE><TR><TD>foo</TD></TR></TABLE>

If a user were to create a table consisting of many rows and many columns, the resultant HTML code would be quite difficult to read since it has no carriage returns or indentation.

CGI::Pretty fixes this problem. What it does is add a carriage return and indentation to the HTML code so that one can easily read it.

```
print table( TR( td( "foo" ) ) );
```

now produces the following output: <TABLE> <TR> <TD>foo</TD> </TR> </TABLE>

Recommendation for when to use CGI::Pretty

CGI::Pretty is far slower than using CGI.pm directly. A benchmark showed that it could be about 10 times slower. Adding newlines and spaces may alter the rendered appearance of HTML. Also, the extra newlines and spaces also make the file size larger, making the files take longer to download.

With all those considerations, it is recommended that CGI::Pretty be used primarily for debugging.

Tags that won't be formatted

The following tags are not formatted: <a>, <pre>, <code>, <script>, <textarea>, and <td>. If these tags were formatted, the user would see the extra indentation on the web browser causing the page to look different than what would be expected. If you wish to add more tags to the list of tags that are not to be touched, push them onto the @AS_IS array:

```
push @CGI::Pretty::AS_IS,qw(XMP);
```

Customizing the Indenting

If you wish to have your own personal style of indenting, you can change the $INDENT variable:

```
$CGI::Pretty::INDENT = "\t\t";
```

would cause the indents to be two tabs.

Similarly, if you wish to have more space between lines, you may change the $LINEBREAK variable:

```
$CGI::Pretty::LINEBREAK = "\n\n";
```

would create two carriage returns between lines.

If you decide you want to use the regular CGI indenting, you can easily do the following:

```
$CGI::Pretty::INDENT = $CGI::Pretty::LINEBREAK = "";
```

AUTHOR

Brian Paulsen <Brian@ThePaulsens.com>, with minor modifications by Lincoln Stein <lstein@cshl.org> for incorporation into the CGI.pm distribution.

Copyright 1999, Brian Paulsen. All rights reserved.

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

Bug reports and comments to Brian@ThePaulsens.com. You can also write to lstein@cshl.org, but this code looks pretty hairy to me and I'm not sure I understand it!

SEE ALSO

CGI