

Perl Basics: Output

How to print your output to the web browser

Page Resource

[CGI Resources](#) | [Perl Tutorials](#)

[Home](#)/[CGI](#)/[Perl](#)/[Output](#)

A simple Perl script like the one we just used simply writes output to the browser. To better understand how this is done, we need to look at a few things involved in printing output to the web browser.

The First Line

Remember the very first line of the script we used in the [last section](#)? This is the one you had to look at to be sure it matched the location of Perl on your server:

```
#!/usr/bin/perl
```

This is a line that needs to be in every Perl program you are going to use (at least on the server). It tells the program where to find the Perl interpreter so that the script can run. Without the line, the script will never get started, because it won't have a way to handle the statements. So, be sure to put this line first and make sure it points to the right place.

The print Command

The print command is what tells the program to print out something. This is where it all starts. Here is the format for the print statement:

```
print "Your text and other things go here";
```

This statement prints the content between the quote marks to standard output. If you are running this from the command line, the above statement would simply write:

```
Your text and other things go here
```

If you want to print this to the web browser, there is a catch. You need to place a Content-type header **before** your print statements.

The Content-type Header

If you want the content of your print statements sent to the web browser to be displayed, you need to print this header before the other print statements to be sure the web browser knows to print the correct type of content. Here is the header to use to create a section of HTML, or an entire HTML page using Perl:

```
print "Content-type: text/html\n\n";
```

This tells the browser the content will be text/html and to skip a line. The `\n\n` at the end is necessary, so be sure to include it. Otherwise, the content header won't do its job.

Print Some HTML

After the content header, sending HTML to the browser is not too difficult. Using the print command, you can write your HTML tags inside the quote marks:

```
print "<HTML><HEAD>";
print "<TITLE>CGI Test</TITLE>";
print "</HEAD>";
print "<BODY><H2>I just wrote a web page using Perl!</H2>";
print "</BODY></HTML>";
```

This is fine if you are using simple commands and text like the tags above. However, what do you do if you want to use a link on that page? Remember, a link tag needs quote marks around the link url. With the print statement, we already have quotes, and an extra set inside of them will throw everything off and give you an error. One way to get around this is to escape the quote marks around the url with the `\` escape character. The `\` character allows you to place certain other characters after it so that they are printed rather than interpreted by Perl. So, to escape the quotes, we need to write `\"` for the inside quote marks:

```
print "<A HREF=\\\"http://someplace.com\\\">Click Here</A>";
```

You will also need this if you use other special Perl characters, such as `$`, `%`, `*` and `@`. We'll get into the special characters in more detail later, but the `@` sign is an important one to remember to escape in HTML, like the quote marks. This is because you may want to print your e-mail address on your page, or use a `mailto:` command to make a clickable e-mail link. You'll need to remember to escape the `@` sign, or you are in for more errors:

```
print "<A HREF=\\\"mailto:john\\@pageresource.com\\\">E-mail Me</A>";
```

Even after all that, you may notice that when you view your HTML source on the finished page, it is all run together and just keeps going on the same line until the viewer you are using breaks the line on its own. You can create line breaks where you need them by using another special formatting character, `\n`, which we used in the content type header earlier. The `\n` stands for "new line", and that is what it does. It breaks to the next line, making your HTML more readable if you like it that way. Remember that the `\n` character goes inside the quote marks of your print statement. Below is an example, it uses the new line character so that your title tags are on their own line:

```
print "<HTML><HEAD>\n";
print "<TITLE>CGI Test</TITLE>\n";
print "</HEAD>";
```

So, instead of one long line, you get a break after the `<HEAD>` tag and the `</TITLE>` tag. The code would now look like this when you view the source:

```
<HTML><HEAD>
<TITLE>CGI Test</TITLE>
</HEAD>
```

Now, here is the source code for the program we used in the last section, it should make much more sense now that you have been through the commands involved in its creation:

```
#!/usr/bin/perl

print "Content-type: text/html\n\n";
print "<HTML><HEAD>";
print "<TITLE>CGI Test</TITLE>";
print "</HEAD>";
print "<BODY><H2>I just wrote a web page using Perl!</H2>";
print "</BODY></HTML>";
```

As you can see, we used a few of the things that were introduced in this section. However, there is an easier way to print HTML to the browser, especially if you need to create a long HTML document. We are covering this method in the next section. So, on we go to [Printing HTML](#).

Partners

[CoolHomepages](#) | [Web Design Library](#) | [Website Content](#)

The tutorials and articles on these pages are © 1997-2005 by John Pollock and may not be reposted without written permission from the author, and may not be reprinted for profit.



[Previous](#)

By: [John Pollock](#)



[Next](#)

[Main Page](#) | [HTML](#) | [JavaScript](#) | [Graphics](#) | [DHTML/Style Sheets](#) | [ASP/PHP](#)
[PutWeb/FTP](#) | [CGI/Perl](#) | [Promotion](#) | [Java](#) | [Design Articles](#)
[Support Forums](#) | [Site Search](#) | [FAQs](#) | [Privacy](#) | [Contact](#)

Copyright © 1997-2005 [The Web Design Resource](#). All rights reserved.