

# Reading Forms in Perl

*How to use cgi-lib.pl to get input*

---

**Page Resource**

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

---

[Home/CGI/Perl/Input 1](#)

Another way to get input is from the user. To do this, we usually have the user fill out a form and submit the information to us. When doing this, there are characters that need to be unescaped after being run through the server. The easiest way to do that is to use an existing bit of code to do the dirty work. The one we will use is called `cgi-lib.pl`, a well-known Perl library.

## Download/View Instructions

The `cgi-lib.pl` file can be downloaded from:

<http://cgi-lib.berkeley.edu/>

While you are there, read through some of the documentation to see what it can do, and installation recommendations. We will be using it here just to make the form input usable for us, but it has lots of other features.

## Install the File

On my server, I just uploaded the file to the directory I wanted to use it in and set the file permissions (I set to 755, yours may be different, check their instructions). Mine works with the file name `cgi-lib.pl`, some may need to rename it to have the `.cgi` extension.

## Try it Out

Now that we have it, we can use it to read input from a form using either the `get` or `post` method (it will work for both). So, we can create a short form and call the `cgi` we will write to get the data:

```
<HTML>
<BODY>
<FORM method="post" action="http://yoursite.com/cgi-bin/testform.cgi">
Name: <INPUT TYPE="text" name="username" size="30">
<BR>
<INPUT TYPE="submit">
</FORM>
</BODY>
</HTML>
```

This form will allow the user to enter his/her name in the text box, and will send it to a `cgi` program named `testform.cgi`. In `testform.cgi`, we will grab the input and display a new page with the username on it:

```
#!/usr/bin/perl

require "cgi-lib.pl";

&ReadParse(*input);

$username= $input{'username'};

print "Content-type: text/html\n\n";
print "<HTML><BODY>";

print "Hi $username, welcome!";

print "</BODY></HTML>";
```

Notice we have a required file, which is cgi-lib.pl. We call it with the require command:

```
require "cgi-lib.pl";
```

Next, we call a subroutine inside the cgi-lib.pl file called &ReadParse, with the input variable. This is where the input is parsed for us and put in a usable form, which will be an associative array.

The associative array is called %input. To get the value we want, we call it with:

```
$input{'name from form'}
```

The name from form part is the name of the input element inside the form (remember, we named the text box "username" using name="username"). So, we assigned the value of the "username" form element to a variable called \$username:

```
$username= $input{'username'};
```

After that, all we had to do was use the variable in the script. You can have more than one form element of course, they can all be read in and used in the same way. The cgi-lib.pl library should make this bit of work easier, as well as do other things if you like.

Well, that's all for now, let's go on to: [Using Environment Variables](#).

---

## 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.