

# Using Variables

## *How to use variables in Perl*

---

**Page Resource**

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

---

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

Using a variable in Perl isn't as bad as it seems. Basically, you just define the variable and then use it. However, there are some things you need to know.

First, a regular variable in Perl is always preceded by the \$ sign. So, if you want a variable named adrevenue, you would have to write it as:

```
$adrevenue
```

### Number Values

Now, we need to give this variable a value. This is often done when the variable is defined. To start, lets give it a numerical value. To do this, we need to set the variable equal to a number using the = sign, and end the statement with a semicolon:

```
$adrevenue=100;
```

You can now use the variable inside your print statement to output the value of the \$adrevenue variable:

```
$adrevenue=100;  
print "The ad revenue for my site today is $adrevenue dollars.";
```

This just prints the sentence:

```
The ad revenue for my site today is 100 dollars.
```

### String Values

To give a variable a string value (text, text with numbers), we need to surround the value using single quotes or double quotes. There is a difference in using the single or double quotes though. If you use single quotes, your string is taken as-is:

```
$my_stomach='full';  
print "My stomach feels $my_stomach.";
```

This gives `$my_stomach` a value of `full`, and that is what is printed:

```
My stomach feels full.;"
```

The use of double quotes allows you to use other variables as part of the definition of a variable. It would now recognize the `$` sign as setting off another variable and not as part of the string. So, now you could use something like this:

```
$my_stomach='full';
$full_sentence="My stomach feels $my_stomach.";
print "$full_sentence";
```

Now, the value of `$my_stomach` is used as part of the `$full_sentence` variable. This can be very handy at times. Now, it would print the following sentence (again):

```
My stomach feels full.;"
```

## Use Them

Now, your Perl script can include some variables (though they don't serve much purpose just yet). Here is a sample script using what we have so far:

```
#!/usr/bin/perl

$adrevenue=100;
$my_stomach='full';
$full_sentence="My stomach feels $my_stomach.";

print "Content-type: text/html\n\n";
print <<ENDHTML;
<HTML>
<HEAD>
<TITLE>Perl Variable Test</TITLE>
</HEAD>
<BODY>
I wish I could make $adrevenue dollars a day from my web site!
<P>
$full_sentence
</BODY>
</HTML>

ENDHTML
```

Now, this prints a wonderful HTML document that says:

I wish I could make 100 dollars a day from my web site!

My stomach feels full.

You can see the script display the HTML example above with the example link below:

[Variables Example](#)

Well, that's it for variables for now. Let's go on to [Perl Operators](#)

---

## 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.