

# Perl Operators: Part 2

## *How to use (more) operators in Perl*

**Page Resource**

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

[Home](#)/[CGI](#)/[Perl](#)/[Operators 2](#)

After all of that information in the last section, you might think there couldn't be any more. Well, there is more-- but these are a little more interesting.

### Numeric Comparison

These operators are used to compare two numbers, but not to compare strings. We'll get to those next. These operators are typically used in some type of [conditional statement](#) that executes a block of code or initiates a loop. We'll get to the conditional statements in the next section, but to introduce this we will use the beginning of an if () condition. Before that, let's look at the list:

Operator	Function
==	Equal to
!=	Not Equal to
>	Greater than
<	Less than
>=	Greater than or Equal to
<=	Less than or Equal to

So, suppose you want to execute some code only if one number is equal to another. You would use the if () condition with the == operator above:

```
$money=5;

if ($money==5)
{
    ....more code....
}
```

Since money is equal to 5, it would execute the code you place between the curly brackets. It works the same way if you use one of the other operators:

```
$money=5;

if ($money<3)
{
    ....more code....
}
```

This time it would not go through, as 5 is not less than 3-- the value of \$money.

## String Comparison

These are similar to the numerical comparisons, but they work with strings. We will note a few differences in the way these work after the list below:

Operator	Function
eq	Equal to
ne	Not Equal to
gt	Greater than
lt	Less than
ge	Greater than or Equal to
le	Less than or Equal to

Strings are equal if they are exactly the same. So, "cool" and "cool" are equal, but "cool" and "coolz" are not. Here is a string equality example:

```
$i_am="cool";

if ($i_am eq "cool")
{
    ....more cool code....
}
```

The greater-than and less-than operators compare strings using alphabetical order. Something that starts with "a" is greater than something that starts with "c". Also, small letters are greater than capital letters. Thus, "hello" is greater than "Hello". So, that is how it will compare it. Here is a sample:

```

$i_am="delightful";

if ($i_am lt "cool")
{
    print "You are not very cool, dude.";
}

```

## Logical Operators

These are often used when you need to check more than one condition. Here they are:

Operator	Function
&&	AND
	OR
!	NOT

So, if you want to see if a number is less than or equal to 10, and also greater than zero:

```

$number=5;

if (($number <= 10) && ($number > 0))
{
    ...code....
}

```

Notice the nested parentheses. Since we are checking for two conditions, we want to be sure the comparison is done first. Thus, they have their own sets of parentheses within the parentheses for the if () condition. Afterward, it checks to see if both conditions are true. Again, more on the conditionals in the next section.

Well, that's it for the operators for now. In the next section, we'll pick up with conditional statements. Let's go on to [Conditional Statements](#)

---

## 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.