PERL LOGICAL OPERATORS EXAMPLE

There are following logical operators supported by Perl language. Assume variable *aholdstrueandvariable*b holds false then:

Operator	Description	Example
and	Called Logical AND operator. If both the operands are true then then condition becomes true.	*$aand$b* is false.
&&	C-style Logical AND operator copies a bit to the result if it exists in both operands.	$a && $b is false.
or	Called Logical OR Operator. If any of the two operands are non zero then then condition becomes true.	*aorb* is true.
||	C-style Logical OR operator copies a bit if it exists in eather operand.	*$a*\|\|*$b* is true.
not	Called Logical NOT Operator. Use to reverses the logical state of its operand. If a condition is true then Logical NOT operator will make false.	not*$aand$b* is false.

Example

Try following example to understand all the logical operators available in Perl. Copy and paste following Perl program in test.pl file and execute this program.

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

$a = true;
$b = false;

print "Value of \$a = $a and value of \$b = $b\n";

$c = ($a and $b);
print "Value of \$a and \$b = $c\n";

$c = ($a  && $b);
print "Value of \$a && \$b = $c\n";

$c = ($a or $b);
print "Value of \$a or \$b = $c\n";

$c = ($a || $b);
print "Value of \$a || \$b = $c\n";

$a = 0;
$c = not($a);
print "Value of not(\$a)= $c\n";
```

When the above code is executed, it produces following result −

```
Value of $a = true and value of $b = false
Value of $a and $b = false
Value of $a && $b = false
Value of $a or $b = true
Value of $a || $b = true
Value of not($a)= 1
```