

# UNIX / LINUX - SHELL ARITHMETIC OPERATORS EXAMPLE

<http://www.tutorialspoint.com/unix/unix-arithmetic-operators.htm>

Copyright © tutorialspoint.com

## Advertisements

The following arithmetic operators are supported by Bourne Shell.

Assume variable **a** holds 10 and variable **b** holds 20 then –

| Operator           | Description   | Example   |
|--------------------|---|---|
| &plus; (Addition)  | Adds values on either side of the operator                            | <code>`expr \$a &amp;plus; \$b`</code> will give 30 |
| - (Subtraction)    | Subtracts right hand operand from left hand operand                   | <code>`expr \$a - \$b`</code> will give -10         |
| * (Multiplication) | Multiplies values on either side of the operator                      | <code>`expr \$a \* \$b`</code> will give 200        |
| / (Division)       | Divides left hand operand by right hand operand                       | <code>`expr \$b / \$a`</code> will give 2           |
| % (Modulus)        | Divides left hand operand by right hand operand and returns remainder | <code>`expr \$b % \$a`</code> will give 0           |
| = (Assignment)     | Assigns right operand in left operand                                 | <code>a = \$b</code> would assign value of b into a |
| == (Equality)      | Compares two numbers, if both are same then returns true.             | <code>[ \$a == \$b ]</code> would return false.     |
| != (Not Equality)  | Compares two numbers, if both are different then returns true.        | <code>[ \$a != \$b ]</code> would return true.      |

It is very important to understand that all the conditional expressions should be inside square braces with spaces around them, for example `[ $a == $b ]` is correct whereas, `[$a==$b]` is incorrect.

All the arithmetical calculations are done using long integers.

## Example

Here is an example which uses all the arithmetic operators –

### [Live Demo](#)

```
#!/bin/sh

a=10
b=20

val=`expr $a + $b`
echo "a + b : $val"

val=`expr $a - $b`
echo "a - b : $val"

val=`expr $a \* $b`
echo "a * b : $val"

val=`expr $b / $a`
```

```
echo "b / a : $val"

val=`expr $b % $a`
echo "b % a : $val"

if [ $a == $b ]
then
    echo "a is equal to b"
fi

if [ $a != $b ]
then
    echo "a is not equal to b"
fi
```

The above script will produce the following result –

```
a + b : 30
a - b : -10
a * b : 200
b / a : 2
b % a : 0
a is not equal to b
```

The following points need to be considered when using the Arithmetic Operators –

- There must be spaces between the operators and the expressions. For example,  $2+2$  is not correct; it should be written as  $2 + 2$ .
- Complete expression should be enclosed between ‘ ‘, called the inverted commas.
- You should use \ on the \* symbol for multiplication.
- **if...then...fi** statement is a decision-making statement which has been explained in the next chapter.