TutorialKart

Bash Arithmetic Operators

Bash Arithmetic Operators

Bash Arithmetic Operators – There are 11 arithmetic operators supported by Bash Shell.

In this tutorial, we will learn the syntax, description and examples for each of the arithmetic operators.

Addition

Computes addition of two numbers provided as operands.

```
#!/bin/bash

x=$(( 15 + 8 ))
echo $x
```

```
#!/bin/bash

x=`expr 15 + 8`
echo $x
```

Output

```

```

Subtraction

Computes subtraction of second operand from first.

```
#!/bin/bash

x=$(( 15 - 8 ))
echo $x
```

```
#!/bin/bash

x=`expr 15 - 8`
echo $x
```

Output

```
7
```

Multiplication

Computes the product of two operands provided.

```
#!/bin/bash

x=$(( 15 * 8 ))
echo $x
```

```
#!/bin/bash

x=`expr 15 * 8`
echo $x
```

Output

```
120
```

Division

Computes the division of first operand by second operand and returns quotient.

```
#!/bin/bash

x=$(( 15 / 8 ))
echo $x
```

```
#!/bin/bash

x=`expr 15 / 8`
echo $x
```

Output

```
1
```

Exponentiation

Computes the result of second operand raised to the power of first operand

Computes the result of second operand raised to the power of first operand.

```
#!/bin/bash

x=$(( 15 ** 8 ))
echo $x
```

```
#!/bin/bash

x=`expr 15 ** 8`
echo $x
```

Output

```
2562890625
```

Modulo

Computes reminder when the first operand is divided by second operand.

```
#!/bin/bash

x=$(( 15 % 8 ))
echo $x
```

```
#!/bin/bash

x=`expr 15 % 8`
echo $x
```

Output

```
7
```

Increment Variable by Constant

The operator increments the value of first operand by the constant provided.

```
#!/bin/bash

x=2
let "x += 5"
echo $x
```

```
#!/bin/bash
```

```
x=2
(( x += 5 ))
echo $x
```

Output

```

```

Decrement Variable by Constant

The operator decrements the value of first operand by the constant provided.

```
#!/bin/bash

x=2
let "x -= 5"
echo $x
```

```
#!/bin/bash

x=2
(( x -= 5 ))
echo $x
```

Output

```

```

Multiply Variable by Constant

The operator multiplies the value of first operand by the constant provided.

```
#!/bin/bash

x=2
let "x *= 5"
echo $x
```

```
#!/bin/bash

x=2
(( x *= 5 ))
echo $x
```

Output

```

```

Divide Variable by Constant

The operator computes (variable / constant) and stores the result back to variable.

```
#!/bin/bash

x=12
let "x /= 5"
echo $x
```

```
#!/bin/bash

x=12
(( x /= 5 ))
echo $x
```

Output

```
2
```

Remainder of Dividing Variable by Constant

The operator computes (variable % constant) and stores the result back to variable.

```
#!/bin/bash

x=12
let "x %= 5"
echo $x
```

```
#!/bin/bash

x=12
(( x %= 5 ))
echo $x
```

Output

```
2
```

Different ways to compute Arithmetic Operations in Bash

Following are some of the variations you may adopt to perform Arithmetic Operations

Following are some of the variations you may adopt to perform Arithmetic Operations.

Double Parenthesis

Double parenthesis could be used for arithmetic expansion. Following is an example that demonstrates the usage of double parenthesis for arithmetic operations.

```
#!/bin/bash

x=10
y=3

echo $(( x + y ))   # addition
echo $(( $x + $y )) # also valid

echo $(( x - y ))   # subtraction
echo $(( $x - $y )) # also valid

echo $(( x * y ))   # multiplication
echo $(( $x * $y )) # also valid

echo $(( x / y ))   # division
echo $(( $x / $y )) # also valid

echo $(( x ** y ))  # exponentiation
echo $(( $x ** $y ))    # also valid

echo $(( x % y ))   # modular division
echo $(( $x % $y )) # also valid

(( x += 4 ))    # increment variable by constant
echo $x

(( x -= 4 ))    # decrement variable by constant
echo $x

(( x *= 4 ))    # mulitply variable by constant
echo $x

(( x /= 4 ))    # divide variable by constant
echo $x

(( x %= 4 ))    # Remainder of Dividing Variable by Constant
echo $x
```

let construction

let command is used to carry out arithmetic operations. Following is an example :

```
#!/bin/bash

x=10
y=3
```

```
z=0

let "z = $(( x + y ))"  # addition
let z=$((x+y))  # also valid without double quotes when no spaces in expression
echo $z

let "z = $(( x - y ))"  # subtraction
let z=$((x-y))
echo $z

let "z = $(( x * y ))"  # multiplication
let z=$((x*y))
echo $z

let "z = $(( x / y ))"  # division
let z=$((x/y))
echo $z

let "z = $(( x ** y ))" # exponentiation
let z=$((x**y))
echo $z

let "z = $(( x % y ))"  # modular division
let z=$((x%y))
echo $z

let "x += 4"    # increment variable by constant
echo $x

let "x -= 4"    # decrement variable by constant
echo $x

let "x *= 4"    # mulitply variable by constant
echo $x

let "x /= 4"    # divide variable by constant
echo $x

let "x %= 4"    # Remainder of Dividing Variable by Constant
echo $x
```

Backticks

Arithmetic expansion could be done using backticks and expr (all purpose expression evaluator). Following is an example :

```
a=5
b=7
c=`expr $a + $b`
echo $c     # 12
```

Conclusion

In this Bash Tutorial, we have learnt Arithmetic Operators supported by Bash Shell with example for each.

Bash Shell Scripting

- ✦ Bash Tutorial
- ✦ Bash Script Example
- ✦ Bash File Extension
- ✦ Bash Echo
- ✦ Bash Comments
- ✦ Bash Variable
- ✦ Bash Command Line Arguments
- ✦ Bash Read User Input
- ✦ Bash Read Password
- ✦ Bash Date Format
- ✦ Bash Sleep

Operators

- ➪ **Bash Arithmetic Operators**

Conditional Statements

- ✦ Bash If
- ✦ Bash If Else
- ✦ Bash Else If
- ✦ Bash Case

Loops

- ✦ Bash For Loop
- ✦ Bash While Loop
- ✦ Bash Until Loop

Strings

- ✦ Bash String Manipulation Examples
- ✦ Bash String Length
- ✦ Bash If String Equals

- Bash Split String
- Bash SubString
- Bash Concatenate String
- Bash Concatenate Variables to Strings

Functions

- Bash Function
- Bash Override Buitlin Commands

Arrays

- Bash Array

Files

- Bash Write to File
- Bash Read File
- Bash Read File line by line
- Bash If File Exists
- Bash If File is Directory
- Bash If File is Readable

Bash Others

- Bash Check if variable is set