Basic Operators in Bash

In Bash scripting, operators are used to perform operations on variables, strings, and files. There are several types of operators in Bash:

1. Arithmetic Operators
2. Relational Operators
3. Logical Operators
4. String Operators
5. File Test Operators

1. Arithmetic Operators

These operators are used to perform arithmetic operations like addition, subtraction, multiplication, and division.

Operator	Description	Example
`+`	Addition	`result=$((a + b))`
`-`	Subtraction	`result=$((a - b))`
`*`	Multiplication	`result=$((a * b))`
`/`	Division	`result=$((a / b))`
`%`	Modulo (remainder)	`result=$((a % b))`
`++`	Increment (by 1)	`((a++))`
`--`	Decrement (by 1)	`((a--))`

Example

```
#!/bin/bash
a=10
b=5
sum=$((a + b))
echo "Sum: $sum"
```

2. Relational Operators

These are used to compare two numbers.

Operator	Description	Example
`-eq`	Equal to	`[ $a -eq $b ]`
`-ne`	Not equal to	`[ $a -ne $b ]`
`-lt`	Less than	`[ $a -lt $b ]`
`-le`	Less than or equal to	`[ $a -le $b ]`
`-gt`	Greater than	`[ $a -gt $b ]`
`-ge`	Greater than or equal to	`[ $a -ge $b ]`

Example

```
#!/bin/bash
a=10
b=5
if [ $a -gt $b ]
then
  echo "$a is greater than $b"
fi
```

3. Logical Operators

Logical operators are used to combine multiple conditions.

Operator	Description	Example
`&&`	AND (both conditions must be true)	`if [ $a -gt 5 ] && [ $b -lt 10 ]`
`		`
`!`	NOT (negates a condition)	`if ! [ -f "$file" ]`

Example

```
#!/bin/bash
a=5
b=8
if [ $a -lt 10 ] && [ $b -gt 5 ]
then
  echo "Both conditions are true"
fi
```

The statement if ! [-f "$file"] is used to check if a file **does not exist** or is **not a regular file** in a Bash script.

Breakdown of the components:

[-f "$file"]:

1. This is a **file test** operator.
2. -f checks if the file specified by $file exists and is a **regular file** (not a directory, symbolic link, or other types of file).
3. If the file exists and is a regular file, the expression returns true (exit status 0).

! **(Logical NOT)**:

1. The ! negates the result of the test.
2. If [-f "$file"] returns true, the ! makes it false. If it returns false (meaning the file doesn't exist or isn't a regular file), the ! makes it true.

How it works:

- **True**: The if block will execute if the file **does not exist** or is **not a regular file**.
- **False**: If the file exists and is a regular file, the if block will be skipped.

Example:

```
#!/bin/bash
file="/path/to/file.txt"

if ! [ -f "$file" ]; then
  echo "File does not exist or is not a regular file."
else
  echo "File exists and is a regular file."
fi
```

Explanation of the Example:

- The script checks if /path/to/file.txt is a regular file.
- If it **does not exist** or is not a regular file (e.g., a directory, symbolic link, etc.), it will output: "File does not exist or is not a regular file."
- If it **exists** and is a regular file, it will output: "File exists and is a regular file."

4. String Operators

String operators are used to perform operations on strings, like checking for equality, length, or pattern matching.

Operator	Description	Example
`=`	Equal to	`[ "$a" = "$b" ]`
`!=`	Not equal to	`[ "$a" != "$b" ]`
`-z`	String is empty	`[ -z "$string" ]`
`-n`	String is not empty	`[ -n "$string" ]`
`>`	Greater than (lexicographically)	`if [[ "$a" > "$b" ]]`
`<`	Less than (lexicographically)	`if [[ "$a" < "$b" ]]`

Example

```
#!/bin/bash
a="hello"
b="world"
if [ "$a" != "$b" ]
then
  echo "Strings are different"
Fi
```

5. File Test Operators

These operators are used to check properties of files (e.g., if a file exists, if it's a directory, etc.).

Operator	Description	Example
`-e`	File exists	`[ -e "$file" ]`
`-f`	File exists and is a regular file	`[ -f "$file" ]`
`-d`	Directory exists	`[ -d "$dir" ]`
`-r`	File is readable	`[ -r "$file" ]`
`-w`	File is writable	`[ -w "$file" ]`
`-x`	File is executable	`[ -x "$file" ]`
`-s`	File is not empty (size > 0)	`[ -s "$file" ]`
`-L`	File is a symbolic link	`[ -L "$file" ]`

Example

```
#!/bin/bash
file="/path/to/file.txt"
if [ -f "$file" ]
then
  echo "File exists and is a regular file."
else
  echo "File does not exist."
Fi
```

Summary

Operator Type	Operator Example	Purpose
Arithmetic	`+`, `-`, `*`, `/`, `%`	Perform arithmetic operations
Relational	`-eq`, `-ne`, `-lt`, `-gt`	Compare numerical values
Logical	`&&`, `	
String	`=`, `!=`, `-z`, `-n`, `<`, `>`	Compare or test properties of strings
File Test	`-e`, `-f`, `-d`, `-r`, `-x`	Test file existence and properties