**Shell Scripting Basics**

Shell scripting allows you to automate tasks, execute multiple commands sequentially, and interact with your system via scripts. Here’s a concise introduction to the basics of shell scripting.

https://chatgpt.com/share/67211b9d-b340-800c-8ab9-2a402cb20a44

**1. What is Shell Scripting?**

A **shell script** is a program written for a command-line interpreter or "shell" (e.g., Bash, Zsh). It contains a series of commands to be executed by the shell, automating repetitive tasks.

**2. Writing a Shell Script**

1. **Create a file**:

bash

Copy code

touch myscript.sh

1. **Make it executable**:

bash

Copy code

chmod +x myscript.sh

1. **Add the Shebang (#!)**  
   The **shebang** tells the system which interpreter to use (usually Bash).

bash

Copy code

#!/bin/bash

1. **Write the script** (inside myscript.sh):

bash

Copy code

#!/bin/bash

echo "Hello, World!"

1. **Run the script**:

bash

Copy code

./myscript.sh

**3. Basic Commands**

* **Printing to the console**:

bash

Copy code

echo "Hello, Niraj!"

* **Variables**:

bash

Copy code

name="Niraj"

echo "Your name is $name"

* **Reading user input**:

bash

Copy code

echo "Enter your age: "

read age

echo "You are $age years old"

**4. Conditional Statements**

* **If-else**:

bash

Copy code

#!/bin/bash

echo "Enter a number: "

read num

if [ $num -gt 10 ]; then

echo "Number is greater than 10"

else

echo "Number is 10 or less"

fi

**1. Arithmetic Operators**

Used for numerical operations.  
These operations are typically performed with expr or $(()).

| **Operator** | **Meaning** | **Example** | **Output** |
| --- | --- | --- | --- |
| + | Addition | expr 5 + 3 or $((5 + 3)) | 8 |
| - | Subtraction | expr 5 - 3 or $((5 - 3)) | 2 |
| \* | Multiplication | expr 5 \\* 3 or $((5 \* 3)) | 15 |
| / | Division | expr 6 / 3 or $((6 / 3)) | 2 |
| % | Modulus (remainder) | expr 5 % 2 or $((5 % 2)) | 1 |

**2. Relational (Comparison) Operators**

Used to compare numbers. These are often used with **if** statements.

| **Operator** | **Meaning** | **Example** |
| --- | --- | --- |
| -eq | Equal to | [ 5 -eq 5 ] |
| -ne | Not equal to | [ 5 -ne 3 ] |
| -lt | Less than | [ 3 -lt 5 ] |
| -le | Less than or equal to | [ 3 -le 5 ] |
| -gt | Greater than | [ 5 -gt 3 ] |
| -ge | Greater than or equal to | [ 5 -ge 5 ] |

Example:

bash

Copy code

num=10

if [ $num -gt 5 ]; then

echo "Greater than 5"

else

echo "Less than or equal to 5"

fi

**3. Logical Operators**

Used to combine multiple conditions.

| **Operator** | **Meaning** | **Example** |
| --- | --- | --- |
| && | Logical AND (both true) | [ $a -gt 5 ] && [ $b -lt 10 ] |
| ` |  | ` |
| ! | Logical NOT (negation) | [ ! $a -eq 5 ] |

Example:

bash

Copy code

a=8

b=3

if [ $a -gt 5 ] && [ $b -lt 10 ]; then

echo "Both conditions are true"

fi

**4. String Operators**

Used to manipulate or compare strings.

| **Operator** | **Meaning** | **Example** |
| --- | --- | --- |
| = | Equal to | [ "$str1" = "$str2" ] |
| != | Not equal to | [ "$str1" != "$str2" ] |
| -z | String is empty | [ -z "$str" ] |
| -n | String is not empty | [ -n "$str" ] |

Example:

bash

Copy code

str="Hello"

if [ -n "$str" ]; then

echo "String is not empty"

fi

**5. File Test Operators**

Used to check for file properties (like existence, permissions, etc.).

| **Operator** | **Meaning** | **Example** |
| --- | --- | --- |
| -e | File exists | [ -e myfile.txt ] |
| -f | Exists and is a regular file | [ -f myfile.txt ] |
| -d | Directory exists | [ -d mydir ] |
| -r | File is readable | [ -r myfile.txt ] |
| -w | File is writable | [ -w myfile.txt ] |
| -x | File is executable | [ -x myfile.txt ] |
| -s | File is not empty | [ -s myfile.txt ] |

Example:

bash

Copy code

if [ -f "myscript.sh" ]; then

echo "File exists and is a regular file"

fi

**6. Assignment Operators**

Used to assign values to variables.

| **Operator** | **Meaning** | **Example** |
| --- | --- | --- |
| = | Assign | a=10 |
| += | Add and assign | a=$((a + 5)) |
| -= | Subtract and assign | a=$((a - 3)) |

Example:

bash

Copy code

a=5

a=$((a + 5))

echo "Value of a: $a" # Output: 10

**7. Escape Sequences**

Used to handle special characters in strings.

| **Escape Sequence** | **Meaning** |
| --- | --- |
| \n | Newline |
| \t | Tab |
| \\ | Backslash |
| \" | Double quote |

Example:

bash

Copy code

echo -e "Hello\nWorld" # -e enables escape sequences

**Summary of Common Usage in Scripts**

bash

Copy code

#!/bin/bash

a=10

b=20

if [ $a -lt $b ]; then

echo "$a is less than $b"

fi

# Arithmetic operation

sum=$((a + b))

echo "Sum: $sum"

# Logical operation

if [ $a -gt 5 ] && [ $b -gt 15 ]; then

echo "Both conditions are true"

fi

# String comparison

str1="Niraj"

str2="Niraj"

if [ "$str1" = "$str2" ]; then

echo "Strings are equal"

fi

# File test

if [ -f "./myscript.sh" ]; then

echo "myscript.sh is a regular file"

fi

**5. Loops**

* **For loop**:

bash

Copy code

for i in {1..5}; do

echo "Number: $i"

done

* **While loop**:

bash

Copy code

counter=1

while [ $counter -le 5 ]; do

echo "Counter: $counter"

((counter++))

done

**6. Functions**

bash

Copy code

#!/bin/bash

greet() {

echo "Hello, $1!"

}

greet "Niraj"

**7. Useful Commands in Shell Scripts**

* **date**: Display the current date and time.

bash

Copy code

echo "Current date: $(date)"

* **ls**: List directory contents.

bash

Copy code

ls -l

* **mkdir**: Create a new directory.

bash

Copy code

mkdir new\_folder

* **grep**: Search for patterns in a file.

bash

Copy code

grep "pattern" filename.txt

**8. Handling Arguments**

bash

Copy code

#!/bin/bash

echo "Script name: $0"

echo "First argument: $1"

echo "Second argument: $2"

Run it with:

bash

Copy code

./myscript.sh arg1 arg2

**9. Error Handling**

Use **set -e** to exit the script if any command fails:

bash

Copy code

#!/bin/bash

set -e

mkdir test\_folder

cd test\_folder || exit

**10. Scheduling Shell Scripts with Cron**

You can schedule shell scripts to run periodically using **cron**.  
Edit the crontab:

bash

crontab -e

Add a cron job (e.g., run script every day at 5 PM):

bash

Copy code

0 17 \* \* \* /path/to/myscript.sh