Day 14- Bash Scripting Part 1

A shell script is a text file containing a series of commands that the shell executes in order.

- Shell scripts often have a `.sh` extension, but it is not required.
- Best practice: name scripts descriptively without relying on `.sh`.
- Before running a script, you must make it executable:
  - `chmod +x script_name`
- Running a script:
- Directly by path:
  - `./script_name`
- If you want to run it from anywhere, add the script directory to PATH:
  - `export PATH=$PATH:/path/to/script`

Variables

- A variable is a placeholder for a value that can change.
- Variables start with a `$` when referenced.
- They must contain only alphanumeric characters and underscores.
- Variables are **case sensitive**.
- Defining and using a variable:
  - `name="DevOps"`
  - `echo $name`
- Example:

```
#!/bin/bash
# A simple greeting script
name="Fatima"
echo "Hello, $name! Welcome to my DevOps journey."
```

Command Line Arguments

When passing arguments to a script, they are automatically assigned to special variables.

- `$0` : Name of the script
- `$1` : First argument
- `$2` : Second argument
- `$3` : Third argument
- `"$@"` : All arguments as separate quoted words
- `"$*"` : All arguments as a single word
- Format:

```
#!/bin/bash
echo "Script name: $0"
echo "First argument: $1"
echo "Second argument: $2"
```

Reading Inputs

- The `read` statement takes input from the user during script execution.
- Example:
  - `read num1`
- To prompt the user:

  ```
  read -p "Enter a number: " num
  ```

  ```
  echo "You entered: $num1"
  ```

- Command line arguments are preferred when scripts are called by other scripts.
- `read` is better for manual interactive scripts.

Operators

Arithmetic Operators:

- `+` : Addition
- `-` : Subtraction
- `*` : Multiplication
- `/` : Division
- `%` : Modulus
- Example:

  ```
  a=10
  b=5
  echo $((a+b))
  ```

- Calculator Script:

  ```
  #!/bin/bash
  # Basic calculator
  read -p "Enter first number: " a
  read -p "Enter second number: " b
  echo "Addition: $((a+b))"
  echo "Subtraction: $((a-b))"
  echo "Multiplication: $((a*b))"
  echo "Division: $((a/b))"
  ```

- Comparison Operators (Integers):
  - `-eq` : Equal
  - `-ne` : Not equal
  - `-lt` : Less than
  - `-le` : Less than or equal to
  - `-gt` : Greater than
  - `-ge` : Greater than or equal to
- Logical Operators:
  - `&&` : Logical AND
  - `||` : Logical OR
  - `!` : Logical NOT
- String Operators:
  - `=` : Equal
  - `!=` : Not equal
  - `<` : Less than (lexicographically)

    - `>` : Greater than (lexicographically)
    - `-z` : String is null
    - `-n` : String is not null
- File Test Operators:
    - `-e file` : Check if file exists
    - `-d file` : Check if directory exists
    - `-f file` : Check if regular file
    - `-r file` : Check if readable
    - `-w file` : Check if writable
    - `-x file` : Check if executable