Session - 13

Shell Scripting – Part 2: Variables, Arguments, Loops, and Conditional Statements

This section builds on the basics and covers more advanced features of shell scripting.

🔹 1. Variables in Shell

📌 Declaring and Using

```
name="shining"
echo "Hello, $name"
```

No space around =, and use $ to access the variable.

🔹 2. Positional Parameters (Arguments)

- $0 → Script name
- $1, $2 → 1st, 2nd argument
- $@ → All arguments

🧪 **Example**

```
#!/bin/bash

echo "Script name: $0"

echo "First argument: $1"

echo "Second argument: $2"
```

Run:

```
./script.sh apple orange
```

🔹 **3. If-Else Conditional**

bash

CopyEdit

```
#!/bin/bash

num=10

if [ $num -gt 5 ]; then

  echo "Greater than 5"

else

  echo "Not greater"

fi
```

🧠 **Operators**

Operator	Meaning
-eq	Equal
-ne	Not equal
-gt	Greater than
-lt	Less than
-ge	Greater or equal
-le	Less or equal

🔹 **4. Case Statement**

```
#!/bin/bash
echo "Enter a letter:"
read letter
case $letter in
  a) echo "You entered A";;
  b) echo "You entered B";;
  *) echo "Invalid input";
esac
```

🔹 5. Loops in Shell

🔁 For Loop

```
for i in 1 2 3
do
  echo "Value: $i"
done
```

🔁 While Loop

```
count=1
while [ $count -le 3 ]
do
  echo "Count: $count"
  ((count++))
done
```

🔹 6. Functions in Shell

```
greet() {
  echo "Hello, $1"
}
greet "Linux"
```

- **7. Example Script – Simple Calculator**

```
#!/bin/bash

echo "Enter two numbers:"
read a
read b

echo "Choose operation (+ - * /):"
read op

case $op in
  +) echo "$a + $b = $((a + b))" ;;
  -) echo "$a - $b = $((a - b))" ;;
  \*) echo "$a * $b = $((a * b))" ;;
  /) echo "$a / $b = $((a / b))" ;;
  *) echo "Invalid operator" ;;
esac
```

✅ Summary

Feature	Use
Variables	name="Linux"
Arguments	$1, $2, $@
If-Else	[$a -gt $b]
Loops	for, while
Case	case $var in ... esac
Functions	myFunc() { ... }