Shell Script Basics

1. Shebang (`#!/bin/bash`)

The first line in many shell scripts is the shebang. It tells the system which interpreter to use to execute the script.

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

2. Comments

Comments are lines that the shell ignores. They start with `#`.

```
# This is a comment
```

3. Variables

Variables store values and are referenced using the `$` symbol.

```
#!/bin/bash
name="John"
echo "Hello, $name"
```

4. Reading User Input

You can prompt the user for input and store it in a variable.

```
#!/bin/bash
echo "Enter your name:"
read name
echo "Hello, $name"
```

5. Basic Commands

Shell scripts can execute any command that you can run in the terminal.

```
#!/bin/bash
echo "Listing files in the current directory:"
ls
```

6. Conditional Statements

You can use `if`, `else`, and `elif` to execute code based on conditions.

```
#!/bin/bash
echo "Enter a number:"
read number
if [ $number -gt 10 ]; then
  echo "The number is greater than 10."
else
  echo "The number is 10 or less."
fi
```

7. Loops

Loops are used to repeat commands.

`for` Loop

```
#!/bin/bash
for i in 1 2 3 4 5
do
  echo "Number: $i"
done
```

`while` Loop

```
#!/bin/bash
count=1
while [ $count -le 5 ]
do
  echo "Count: $count"
  count=$((count + 1))
done
```

8. Functions

Functions allow you to group commands into reusable blocks.

```
#!/bin/bash
greet() {
  echo "Hello, $1"
}

greet "Alice"
greet "Bob"
```

9. File Operations

You can create, read, write, and delete files within a script.

Create a File

```
#!/bin/bash
echo "This is a sample file." > sample.txt
```

Append to a File

```
#!/bin/bash
echo "This is appended text." >> sample.txt
```

Read from a File

```
#!/bin/bash
while read line
do
  echo $line
done < sample.txt
```

Delete a File

```
#!/bin/bash
rm sample.txt
```

10. Script Execution

Make your script executable with the `chmod` command:

```
chmod +x your_script.sh
```

Run the script:

```
./your_script.sh
```

Example Script

Here's an example script that combines some of these basics:

```
#!/bin/bash
# Greet the user
echo "Enter your name:"
read name
echo "Hello, $name!"

# List files in the current directory
echo "Listing files in the current directory:"
ls

# Loop through numbers and print them
for i in 1 2 3 4 5
do
  echo "Number: $i"
done

# Create and read a file
echo "This is a sample file." > sample.txt
echo "Contents of sample.txt:"
cat sample.txt

# Delete the file
rm sample.txt
echo "sample.txt has been deleted."
```

Some sample Scripts

1. Hello World Script

```
#!/bin/bash
echo "Hello, World!"
```

2. User Input and Greeting Script

```
#!/bin/bash
echo "Enter your name:"
read name
echo "Hello, $name!"
```

3. File Listing Script

```
#!/bin/bash
echo "Listing files in the current directory:"
ls
```

4. Simple Calculation Script

```
#!/bin/bash
echo "Enter two numbers:"
read num1
read num2
sum=$((num1 + num2))
echo "Sum of $num1 and $num2 is: $sum"
```

5. File Backup Script

```
#!/bin/bash
echo "Enter the file to backup:"
read filename
backup_dir=~/backup
cp $filename $backup_dir
echo "Backup of $filename completed in $backup_dir"
```

6. Check if a File Exists

```
#!/bin/bash
echo "Enter a file name:"
read filename
if [ -f "$filename" ]; then
  echo "$filename exists."
else
  echo "$filename does not exist."
fi
```

7. Loop Through Numbers and Print

```
#!/bin/bash
echo "Looping through numbers 1 to 5:"
for i in {1..5}
do
  echo "$i"
done
```

8. Display System Information

```
#!/bin/bash
echo "System Information:"
echo "Hostname: $(hostname)"
echo "CPU: $(grep 'model name' /proc/cpuinfo | head -n 1 | cut -d ':'
-f 2 | sed 's/^[ \t]*//')"
echo "Memory: $(free -h | awk '/^Mem:/ {print $2}')"
echo "Disk Usage: $(df -h / | awk '/\// {print $5}')"
```

9. Simple Menu Selection

```
#!/bin/bash
echo "Menu Selection:"
echo "1. Display Date and Time"
echo "2. Display Calendar"
echo "3. Display Disk Usage"
read choice
case $choice in
  1) echo "Current date and time: $(date)";;
  2) cal;;
  3) df -h;;
  *) echo "Invalid choice";;
esac
```

10. Check Internet Connectivity

```
#!/bin/bash
echo "Checking internet connectivity..."
ping -c 1 google.com > /dev/null
if [ $? -eq 0 ]; then
  echo "Internet is up!"
else
  echo "Internet is down."
fi
```

This should give you a good start on writing and understanding basic shell scripts!