

Here's a **Shell Scripting Survival Cheat Sheet** 📄 — a quick reference you can keep handy while coding.

Shell Scripting Cheat Sheet

Basics

- **Shebang (defines interpreter):**
 - `#!/bin/bash`
 - **Comments:**
 - `# This is a comment`
-

Variables

- Assign: `VAR=value` (no spaces)
 - Use: `$VAR`
 - Example:
 - `name="Shoubhik"`
 - `echo "Hello $name"`
-

Arithmetic

- Use `$(())` for math:
 - `x=5`
 - `y=3`
 - `echo $((x + y)) # 8`
-

Input & Output

- **Read input:**
 - `read username`
 - `echo "Welcome $username"`
 - **Redirect output:**
 - `echo "Log entry" > file.txt # overwrite`
 - `echo "Another entry" >> file.txt # append`
-

Conditionals

```
if [ $x -gt 10 ]; then
    echo "Greater than 10"
elif [ $x -eq 10 ]; then
    echo "Equal to 10"
else
    echo "Less than 10"
fi
```

Loops

- **For loop:**
 - for i in 1 2 3; do
 - echo "Number \$i"
 - done
- **While loop:**
 - count=1
 - while [\$count -le 3]; do
 - echo "Count \$count"
 - count=\$((count + 1))
 - done

Functions

```
say_hello() {
    echo "Hello $1"
}

say_hello "World"
```

Command Substitution

```
today=$(date)
echo "Today is $today"
```

Exit Codes

- 0 = success, non-zero = error

ls /notexist

```
echo $? # prints exit code
```

Case Statement

```
read choice
```

```
case $choice in
```

```
y|Y) echo "Yes";;
```

```
n|N) echo "No";;
```

```
*) echo "Invalid";;
```

```
esac
```

Useful Shortcuts

- && → run next if success
- || → run next if fail
- ; → run sequentially

Example:

```
mkdir test && cd test || echo "Failed"
```

 This cheat sheet covers the **core syntax** you'll use 90% of the time.