

Bash Scripting (Rough notes)

- Plain text editor

- Terminal emulator
- Bash

Diff OS :-

- MacOS :- Bash is pre installed
- Linux distros already has bash
- **B**ourne **A**gain **S**hell

Extension : .sh

Step 1 : create a new empty file.sh

Step 2 : `#!/bin/bash` → Shebang

Step 3 :- `chmod +x`

Step 4 :- `./` → Run

Step 5 :- `#` is used as comments

Try later on :-

```
#!/bin/sh
```

```
echo
```

```
echo "Current working directory: $(pwd)"
```

```
echo
```

```
echo "Today's date & time: $(date)"
```

```
echo
```

➔ Variables :- only letters (A to Z or a to z), numbers (0 to 9) or (`_`)

- Defining Variables

```
variable_xyz=variable_value
```

```
VAR1="Deadpool"
```

```
Var2=111
```

Read-only Variables

- `#!/bin/sh`
-
- `NAME="DEADPOOL"`
- `readonly NAME`
- `NAME="IRONMAN"`

Something that you need to remember :-

- Never put spaces next to the equal sign, why?

Because Bash will that you you want to run a process instead

```
#!/bin/sh
```

```
HELLO="hello variable world"
```

```
echo $HELLO #should print "hello variable world"
```

The syntax of if, elseif, else is:

```
if <test_expression>; then
    <command-to-execute>
elif <test_expression>; then
    <command-to-execute>
else
    <command-to-execute>
fi
```

