



SECTION 4

REQUESTING USER INPUT

SECTION CHEAT SHEET

POSITIONAL PARAMETERS

The shell assigns numbers called positional parameters to each command-line argument that is entered. (\$1, \$2, \$3...)

Example Script:

```
myscript Tom /home/Tom red
```

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

```
echo "My Name is $1"
```

```
echo "My home directory is $2"
```

```
echo "My favourite colour is $3"
```

SPECIAL PARAMETERS

Special parameters are like regular parameters, but are created for us by the shell, and are **unmodifiable**

Special Parameter	Description	Command Line	Parameter Value for Command Line
<code>\$#</code>	Stores the number of command line arguments provided to the script	<code>example.sh 1 2 3</code>	3
<code>\$0</code>	Stores the script name	<code>test.sh 1 2 3</code>	<code>test.sh</code>
<code>\$@</code>	Expands to each positional parameter as its own word with subsequent word splitting	<code>example.sh "daily reports"</code>	<u>daily</u> <u>reports</u> (2 words)
<code>"\$@"</code>	Expands to each positional parameter as its own word without subsequent word splitting	<code>example.sh "daily reports"</code>	<u>daily_reports</u> (1 word)
<code>\$*</code>	Exactly the same as <code>\$@</code>	<code>example.sh "daily reports"</code>	<u>daily</u> <u>reports</u> (2 words)
<code>"\$*"</code>	Expands to all positional parameter as one word separated by the first letter of the IFS variable without subsequent word splitting	IFS=, <code>example.sh "daily reports"</code>	<u>daily,reports</u> (1 word)
<code>\$?</code>	Gives the exit code returned by the most recent command	<code>echo "hello"</code> <code>echo \$?</code>	0 (because <code>echo "hello"</code> was successful)

THE READ COMMAND:

The read command asks for input from the user and saves this input into a variable

Syntax for the read command:

```
read variable
```

OPTIONS

`-p "prompt"`

Displays a **prompt** to user about what information they must enter

`-t time`

Timeout if the user does not enter any value within **time** seconds.

`-s`

Prevent the input that the user enters from being shown in the terminal. The "secret" option.

`-N chars`

Limit the users response to exactly **chars** characters

`-n chars`

Limit the users response to a maximum of **chars** characters

Example Script:

```
#!/bin/bash

read -t 5 -p "Input your first name within 5 seconds: " name
read -n 2 -p "Input your age (max 2 digits): " age
read -s -N 5 -p "Enter your zip code (exactly 5 digits): " zipcode
echo "$name, $age, $zipcode" >> data.csv
```

THE SELECT COMMAND

The select command provides the user with a dropdown menu to select from. The user may select an option from a list of **options**.

It is also possible to provide a prompt to a user using the **PS3** shell variable.

Syntax for the select command:

```
PS3="Please select an option below: "
select variable in options; do
    commands...
    break
done
```

Example Script:

```
#!/bin/bash
PS3="What is the day of the week?: "
select day in mon tue wed thu fri sat sun; do
    echo "The day of the week is $day"
    break
done
```
