IT Scripting and Automation

-Bash Shell Functions-

Lecturer: Art Ó Coileáin

Why to use Functions?

- Write one and use it many times. **REUSABILITY**
- Reduces script length.
- Easier to maintain.
- Do not repeat yourself ! **DRY**

- Functions allow you to write a piece of code and use it many times. Simple call the function that contains the code needed.

Functions

- If you need to perform the same action (task) multiple times in a script that is sign that a function is required.
- A function is a block of reusable code that preforms an action and returns an exit status.
- All functions must be defined before use.
- Data can be pass into a function, the data can be accessed within that function as parameters.

Creating a Function in Shell Script

Syntax:

```
function function-name () {
        # Code goes here.
}

function-name () {
        # Code goes here.
}
```

Creating a Function in Shell Script

Example:

```
#!/bin/bash
function display-hello () {
        echo “Hello World !”
}
display-hello
```

Functions Called by Other Functions

Example :

```
#!/bin/bash
function display-date() {
        echo “Today’s date is $(date +%c)”
}
function display-hello() {
        echo “Hello World !”
        display-date
}
display-hello
```

Functions Called by Other Functions

Scripts are **not** pre-compiled. Functions **must** be defined before they are used.

Example of incorrect usage:

```
#!/bin/bash
function display-hello() {
        echo “Hello World !”
        display-date
}
display-hello # DO NOT DO THIS
function display-date() {
        echo “Today’s date is $(date +%c)”
}
```

Functions Called by Other Functions

Scripts are **not** pre-compiled. Functions must be defined before they are used.

Example, function with loop:

```
#!/bin/bash
function display-hello() {
for NAME in $@
do
        echo “Hello $NAME !”
done
}
display-hello Sean Daniel Mark
```

Global Variables in Functions

```
#!/bin/bash
function variable_function() {
        GLOBAL_VAR=1
}
# GLOBAL_VAR not available yet (empty).
echo $GLOBAL_VAR
variable_function
# GLOBAL_VAR available.
echo $GLOBAL_VAR
```

Local Variables in Functions

- Can only be accessed within the function
- Create using the **local** keyword

```
local LOCAL_VAR=1
```

- Only functions can have local variables

Exist Status (Return Code) in Functions

- Functions have an exit status
- Explicitly

  return <RETURN_CODE>
- Implicitly
  - The exit status of the last command executed in the function
- Remember return codes:
  - **0 = successful**
  - **1-255= unsuccessful**