

Scripting & Computer Environments

Shell Scripting III

IIT-H

Aug 31, 2013

Previously: Shell Scripting II

- Expressions
operators, arithmetics ...
- Decision Making
if...fi, case...esac
- Looping
for, while, until,
break, continue

Today:

- Functions
- Going graphical

- The whats and whys?

Function

- A named section of a code that can be executed.
- Two parts: body + arguments.
- May return a value.
- Usage:

```
function name                # or just name()
{
    statements
    return <value>           # optional
}
```

- The whats and whys?

Function

- A named section of a code that can be executed.
- Two parts: body + arguments.
- May return a value.
- Usage:

```
function name                # or just name()  
{  
    statements  
    return <value>           # optional  
}
```

Example

```
#!/bin/bash

function func1                # or just func1()
{
    echo "Inside func1"
}

i=1
while [ $i -le 10 ]
do
    func1                      # calling...
    i=$((i + 1))
done

echo "Now, outside func1"
```

“Thou shalt validate thy input data!”

- Returning a value from functions

- ➊ Via the default exit status

- ➋ The `return` command

- ➌ Output of the function

- Function Variables

- ➊ Global

- ➋ Local

- Recursive Functions

- Arrays and Functions

(read!)

- Returning a value from functions

- ➊ Via the default exit status

- ➋ The `return` command

- ➌ Output of the function

- Function Variables

- ➊ Global

- ➋ Local

- Recursive Functions

- Arrays and Functions

(read!)

- Returning a value from functions

- ① Via the default exit status

- ② The `return` command

- ③ Output of the function

- Function Variables

- ① Global

- ② Local

- Recursive Functions

- Arrays and Functions

(read!)

- Returning a value from functions

- ① Via the default exit status

- ② The `return` command

- ③ Output of the function

- Function Variables

- ① Global

- ② Local

- Recursive Functions

- Arrays and Functions

(read!)

- The system by which a user interacts with a machine.
- Hardware + software components
- Some UIs:
 - Command-line Interfaces (CLI)
 - Graphical User Interfaces (GUI)
 - Voice User Interfaces
 - Natural-language Interfaces
- Design goal: Produce a *user-friendly, efficient* and *enjoyable* UI.
- The area of *usability* strives to achieve this goal.

- Poor UI design costs a lot.
e.g. computer rage, Internet rage, industrial disaster ...



- Real story:

Confusion over Palm Beach County ballot

Although the Democrats are listed second in the column on the left, they are the third hole on the ballot.

(REPUBLICAN)	(DEMOCRATIC)	(LIBERTARIAN)	(GREEN)	(SOCIALIST WORKERS)	(NATURAL LAW)
GEORGE W. BUSH - PRESIDENT	AL GORE - PRESIDENT	HARRY BROTHME - PRESIDENT	RALPH NADER - PRESIDENT	JAMES HARRIS - PRESIDENT	JOHN HAZELIN - PRESIDENT
DICK CHENEY - VICE PRESIDENT	JOE LIKERMAN - VICE PRESIDENT	ART DELMER - VICE PRESIDENT	WINONA LAURIE - VICE PRESIDENT	MARGARET TROVIE - VICE PRESIDENT	NAT KONDRAUER - VICE PRESIDENT

Punching the second hole casts a vote for the Reform Party.

(REFORM)	(SOCIALIST)	(CONSTITUTION)	(WORKERS WORLD)
PAT OUCHARAN - PRESIDENT	DAVID McREYNOLDS - PRESIDENT	HOWARD PHELPS - PRESIDENT	MONICA MCDONHEAD - PRESIDENT
EDGELA FOSTER - VICE PRESIDENT	MARY CAL VOLLIS - VICE PRESIDENT	J. CURTIS FRAZER - VICE PRESIDENT	GLORIA La REFA - VICE PRESIDENT

WRITE IN CANDIDATE
To vote for a write in candidate, follow the directions on the long stick of your ballot card.

- A step-up in interactive scripts.
- The `dialog` command offers the feel of standard dialog boxes in a text environment.
- Not part of the shell.
- Has various window elements (a.k.a. widgets) to choose from:

message box

menu bar

textbox

input box

password box

yesno ...

Dialog Usage

`dialog [options] --WidgetName Parametres` (note the `--`)

Widget Name	Description
infobox	Displays a message
msgbox	Displays a message (waits for response)
inputbox	Displays a box for text entry
yesno	Displays a box with yes and no buttons
menu	Displays items to select from
textbox	Displays text in a scrollable window
checklist	Displays multiple entries (\neq 1 can be selected at once)
radiolist	Displays multiple entries (only one can be selected)
gauge	Displays a meter showing progress of a task.

- Each widget provides output in 2 forms:

❶ Using exit status

❷ Using STDERR

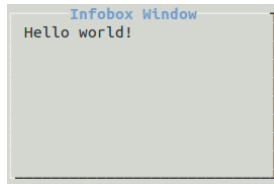
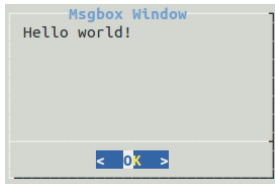
msgbox & infobox

```
dialog --msgbox    <text>    <height>    <width>  
dialog --infobox   <text>    <height>    <width>
```

- To display the message <text> to the user.
- msgbox waits for the user to click 'OK'.
- infobox does not.

```
dialog --title "Msgbox Window" --msgbox 'Hello world!' 10 30
```

```
dialog --title "Infobox Window" --infobox 'Hello world!' 10 30
```

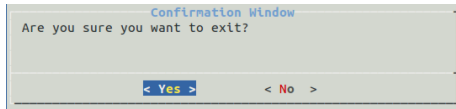


yesno

```
dialog --yesno <text> <height> <width>
```

- To display the user a yes/no question window.
- Exit statuses : 0 (if yes), 1 (if no), 255 (if esc key was pressed)

```
dialog --title "Confirmation Window" \
--yesno "Are you sure you want to exit?" 6 60
```



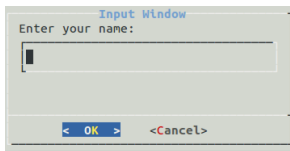
inputbox

```
dialog --inputbox <text> <height> <width> [<init>]
```

- To prompt the user for input data.
- The input is sent to STDERR.
- To retrieve the input value, change the redirection.

```
dialog --title "Input Window" \
      --inputbox "Enter your name:" 10 20 2>name.txt
```

```
cat name.txt
```



passwordbox

```
dialog --passwordbox <text> <height> <width> [<init>]
```

- Just like inputbox, except text is hidden.
- The `--insecure` option displays password as * .
- On exit, input printed on dialog's output.
- Less secure.

```
dialog --title "Password Window" \  
--passwordbox "Enter your password" 10 40 2> secret.txt
```

```
dialog --title "Password Window" \  
--insecure --passwordbox "Enter your password" 10 40 2> secret.txt
```

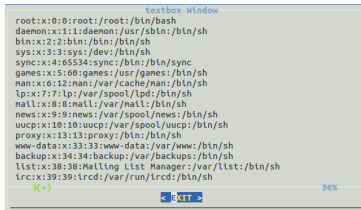
textbox

```
dialog --textbox <file> <height> <width>
```

- To display scrollable window containing contents of <file>.
- Use the arrow keys to scroll.

```
dialog --title "textbox Window" --textbox /etc/passwd 20 70
```

```
ps -au > process.txt ; dialog --textbox process.txt 20 50
```



```
textbox Window
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mail List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
(*)
EXIT
```

menu

```
dialog --menu <text> <height> <width> <menu height> \
    <tag1> <item1> <tag2> <item2> ...
```

- Displays list of choices to the user in the form of a menu.
- Each menu element is made of {tag} and {item}.
- Each {tag} identifies an entry from the other entries.
- {item} is just short description of the choice.
- On exit, the {tag} will be redirected to STDERR. The redirection can be changed to a file.

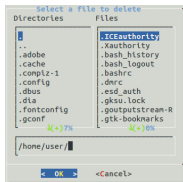
```
dialog --menu "Demo of Menu Widget" 20 30 10 1 "Display users" \
2 "Display Calendar" 3 "Display Disk usage" 4 "Display Time" \
5 "Exit" 2>menu.txt
```

fselect

`dialog --fselect <filepath> <height> <width>`

- Allows the user to browse to the file location and select the file.
- Displays directory listing (left), files in that directory (right) and a textbox containing the currently selected file or directory.

```
FILE=$(dialog --title "Select a file to delete" \
--fselect $HOME/ 15 40 )
echo $FILE
```



Other Widgets

- | | | | |
|------------|------------------------------|--------------------|----------------------------|
| ❶ Checkbox | (<code>--checklist</code>) | ❺ Progress bar | (<code>--gauge</code>) |
| ❷ Radiobox | (<code>--radiolist</code>) | ❻ Directory select | (<code>--dselect</code>) |
| ❸ Calendar | (<code>--calendar</code>) | ❼ Tailbox | (<code>--tailbox</code>) |
| ❹ Timebox | (<code>--timebox</code>) | | |

Read `dialog`'s man page for more!