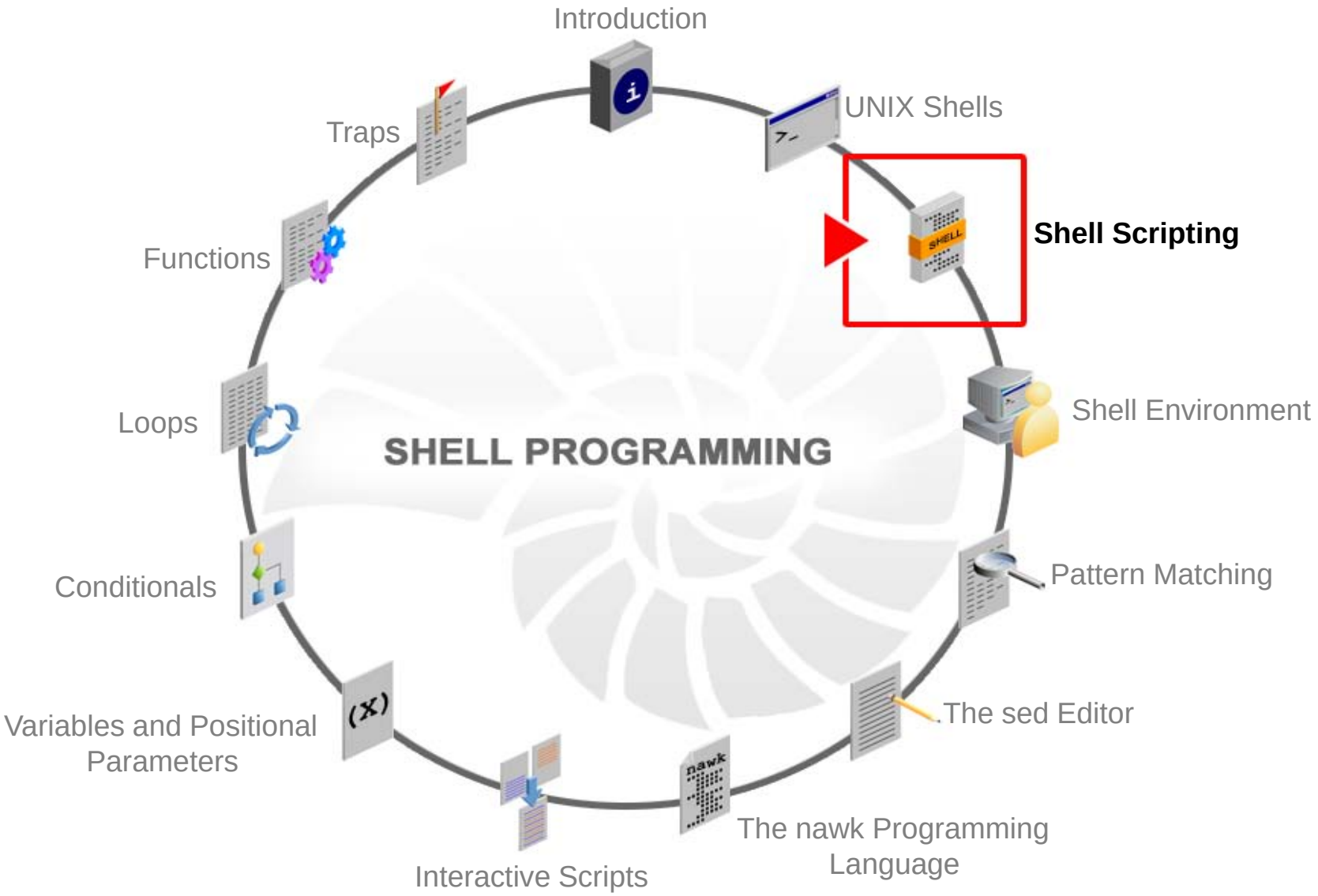
Shell Scripting

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Objectives

After completing this lesson, you should be able to:

- Describe the structure of a shell script
- Create a simple shell script
- Implement the various debugging options in a shell script

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Agenda

- Describing the structure of a shell script
- Creating a simple shell script
- Implementing the various debugging options in a shell script

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

What Is a Shell Script?

- A shell script is a noncompiled program written for the shell or the command-line interpreter of an operating system (OS).
- Shell scripts are:
  - Text files that contain shell and UNIX commands
  - Programs that have a specific purpose
  - Interpreted by a shell (for example, bash shell)
  - Useful in automating repetitive tasks

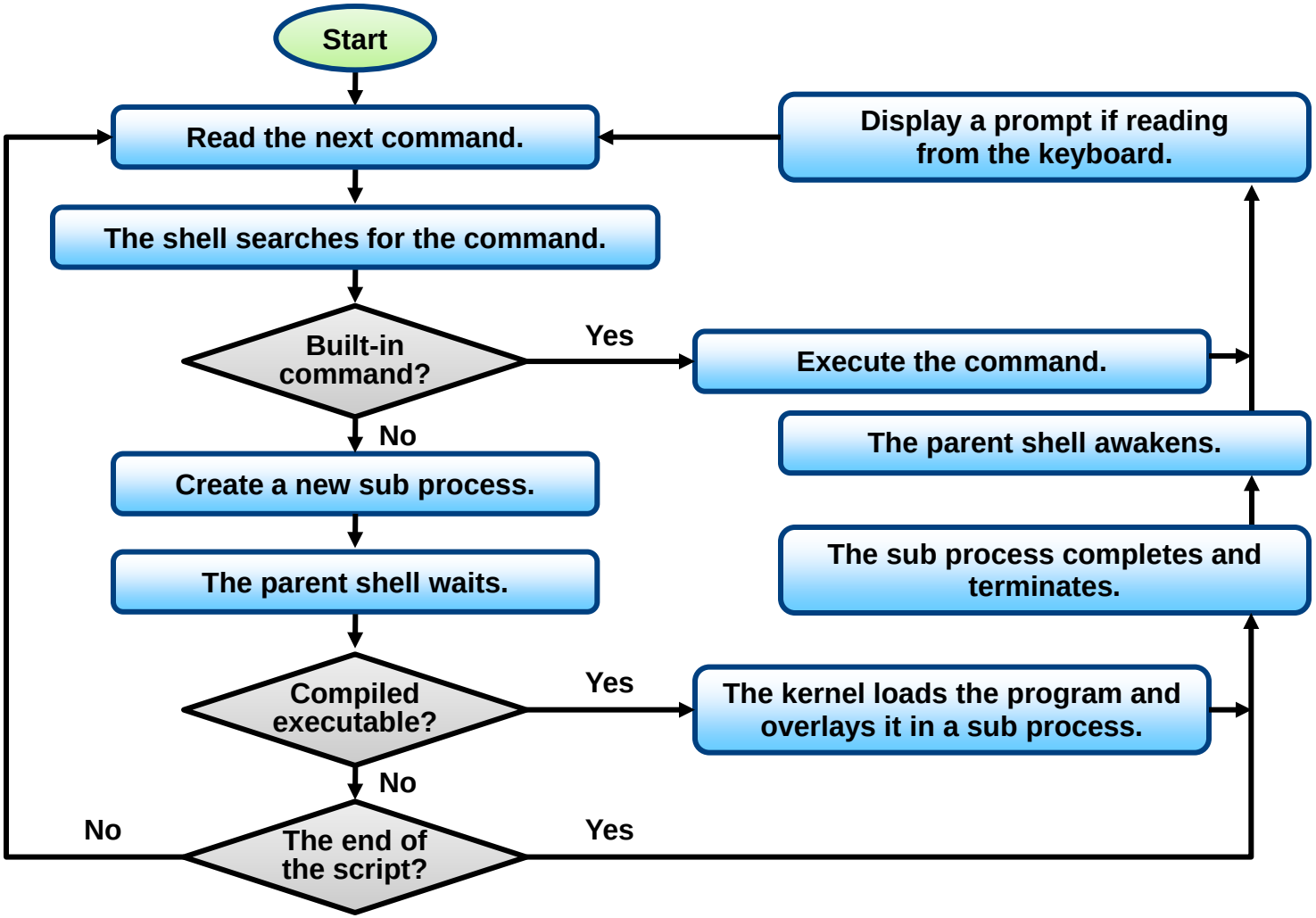
Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Components of a Shell Script

- For a script to execute, you must ensure that it has correct logic and flow control.
- Beyond that, the structure of a shell script is flexible and some of the components include:
  - Comments
  - Information displays
  - Conditional testing
  - Looping constructs
  - Arithmetic operations
  - String manipulation
  - Variable manipulation and testing
  - Argument and option handling

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Execution Order of a Shell Script



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Agenda

- Describing the structure of a shell script
- Creating a simple shell script
- Implementing the various debugging options in a shell script

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Simple Shell Script: Example

In this sample shell script named `my_script`:

- `#!/bin/bash`: Specifies the shell the script is written for. In this case it is the bash shell.
- `# My first script`: Is a comment entry, which is not interpreted by the shell
- `echo`: Is a command to display text on the screen
- `exit`: Is a command that denotes the end of script and tells the shell to exit the script

```
# cat my_script.sh
#!/bin/bash
# My first script

echo “Hello World!”
exit 0
```


Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Requirements: Tools and Resources

Before you begin to write a shell script, you need to:

- Choose an OS
  - UNIX-based OS (Oracle Solaris 11.1, Oracle Linux 6.5)
- Choose an editor
  - `vi`, `vim`, `sed`, `gedit` (graphical)
- Set up the environment
  - The shell profile, aliases, history by using the `vi` editor

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Programming Terminologies

Term	Meaning
Logic flow	The overall design of the program or procedure. The logic flow determines the logical sequence of the tasks that are involved in the program or procedure so that a successful and controlled outcome is achieved.
Loop	A portion of the procedure that is performed zero or more times.
User input	Information that is provided by an external source, during the running of the program or procedure, that is used within the program or procedure.
Conditional branch	A logical point in the program or procedure when a condition determines the actions that are subsequently performed.
Command control	Testing the `exit` status of a command to control whether a portion of code should be performed.

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Instance of a Logic-Flow Design

- When developing a script, use the logic-flow design.
- For example:
  1. Do you want to add a user?
     a. If Yes:
        1. Enter the user's name.
        2. Choose a shell for the user.
        3. Determine the user's home directory.
        4. Determine the group to which the user belongs.
     b. If No, go to Step 3.
  2. Do you want to add another user?
     a. If Yes, go to Step 1.a.
     b. If No, go to Step 3.
  3. Exit.

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The echoscript1.sh Script: Example

```
# cat echoscript1.sh
#!/bin/bash

clear
echo "SCRIPT BEGINS“

echo "Hello $LOGNAME"
Echo

echo "Todays date is: \c"
date ’+%m/%d/%y’

echo "and the current time is: \c"
date ’+%H:%M:%S%n’

echo "Now a list of the processes in the current shell"
ps

echo "SCRIPT FINISHED!!"
```


Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Creating a Shell Script

To create a shell script, perform the following steps:

1. Create a file.
2. Invoke the shell.
3. Include commands in the script file.
4. Add comments to the script file, if required.
5. Execute the script file.

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Creating a File

- A script file can have any name that follows the conventions of regular file names in the OS environment.
- When naming your shell scripts, avoid using names that conflict with existing UNIX commands or shell functions, unless you wish to modify the behavior of an existing command.

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Invoking the Shell

- The first line of the script contains the shebang characters that invoke the shell interpreter.
- The shebang characters on the first line are `#!`, followed by the path name for the subshell to execute the script.

```
$ cat hello.sh
#!/bin/bash
……
```

Note: The shebang characters are *the first two characters on the first line,* meaning that there must be no blank lines or spaces before these characters.

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Including Commands in a Shell Script

UNIX commands include standard utilities, user programs, and the names of other scripts that you need in order to accomplish your task.

Note: If you put the names of other scripts in your script, ensure they have `read` and `execute` permissions so that they will run.

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Adding Comments in a Shell Script

- Any line other than the first line in the script that starts with a # is a comment.
- Comments provide useful documentation to programmers and others who need to understand or debug the code.
- For example:

```
$ cat hello.sh
#!/bin/bash
# This is a hello world program
```


Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Executing a Shell Script

There are various ways to execute a shell script:

- To execute a shell script in the subshell, first assign the `execute` permission to the shell script.

```
$ chmod 744 scriptname
$ scriptname
```

- To execute a shell script in the current shell, you do not need the `execute` permission on the script.

```
$ . ./scriptname
```

- To invoke certain options in the script, execute a script by using the shell command.

```
$ bash –s scriptname
```


Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Example: hello.sh Script

```
$ vi hello.sh

#!/bin/bash
# This and following line is a comment
# This is a hello world program

echo "Hello World"
~
~
~
:wq

$ chmod 700 hello.sh

$ ./hello.sh
Hello World
```


Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Quiz

What is the purpose of the shebang characters?

a. Creating a file
b. Invoking the shell
c. Adding comments to the script file, if required
d. Executing the script file

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Agenda

- Describing the structure of a shell script
- Creating a simple shell script
- Implementing the various debugging options in a shell script

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Debugging a Shell Script

- The shell provides a debugging mode, which allows you to locate problems in a script.
- To run an entire script in the debug mode:
  - Add `-x` after `#!/bin/bash` on the first line.

```
#!/bin/bash -x
```

- To run an entire script in the debug mode from the command line:
  - Add `-x` to the bash command used to execute the script.

```
$ bash -x scriptname
```


Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Debug Statement Options

To run several portions of a script in the debug mode, place `set -option` at the debugging start point and `set +option` where it needs to stop.

Option	Description
`set -x`	Prints the statements after interpreting metacharacters and variables
`set -v`	Prints the statements before interpreting metacharacters and variables
`set -f`	Disables file name generation (using metacharacters)

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Example: Debug Mode Specified on the #! Line

```
$ cat debug1.sh
#!/bin/bash –x

echo "Your terminal type is set to: $TERM"
echo

echo "Your login name is: $LOGNAME"
echo

echo "Now we will list the contents of the /etc/security directory"
ls /etc/security
echo
```


Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Example: Debug Mode Specified on the #! Line

```
$ ./debug1.sh
+ echo Your terminal type is set to: ansi
Your terminal type is set to: ansi
+ echo

+ echo Your login name is: root
Your login name is: root
+ echo

+ echo Now we will list the contents of the /etc/security directory
Now we will list the contents of the /etc/security directory
+ ls /etc/security

audit                   audit_user      dev             policy.conf
audit_class             audit_warn      device_policy   priv_names
audit_control           auth_attr       exec_attr       prof_attr
audit_event             bsmconv         extra_privs     spool
audit_record_attr       bsmunconv       kmfpolicy.xml   tsol
audit_startup           crypt.conf      lib
+ echo
```


Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Example: Debug Mode with the `set –x` Option

```
$ cat debug2.sh
#!/bin/bash

set -x
echo "Your terminal type is set to: $TERM"
echo
set +x

echo "Your login name is: $LOGNAME"
cho

echo "Now we will list the contents of the /etc/security directory"
ls /etc/security
echo
```


Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Example: Debug Mode with the `set –x` Option

```
$ ./debug2.sh
+ echo Your terminal type is set to: ansi
Your terminal type is set to: ansi
+ echo

+ set +x
Your login name is: root

Now we will list the contents of the /etc/security directory
audit                   audit_user       dev              policy.conf
audit_class             audit_warn       device_policy    priv_names
audit_control           auth_attr        exec_attr        prof_attr
audit_event             bsmconv          extra_privs      spool
audit_record_attr       bsmunconv        kmfpolicy.xml    tsol
audit_startup           crypt.conf       lib
```


Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Example: Debug Mode with the `set –v` Option

```
$ cat debug3.sh
#!/bin/bash

set -v
echo "Your terminal type is set to: $TERM"
echo

echo "Your login name is: $LOGNAME"
echo

echo "Now we will list the contents of the /etc/security directory"
ls /etc/security
echo
```


Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Example: Debug Mode with the set –v Option

```
$ ./debug3.sh
echo "Your terminal type is set to: $TERM"
Your terminal type is set to: ansi
echo

echo "Your login name is: $LOGNAME"
Your login name is: root
echo

echo "Now we will list the contents of the /etc/security directory"
Now we will list the contents of the /etc/security directory
ls /etc/security
audit audit_              user              dev               policy.conf
audit_class               audit_warn        device_policy     priv_names
audit_control             auth_attr         exec_attr         prof_attr
audit_event               bsmconv           extra_privs       spool
audit_record_attr         bsmunconv         kmfpolicy.xml     tsol
audit_startup             crypt.conf        lib
echo
```


Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Quiz

Which of the following commands would you use to run an entire script in the debug mode?

a. `$ bash -x scriptname`

b. `#!/bin/bash –x`

c. `set –x`

d. `set -v`

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Summary

In this lesson, you should have learned how to:

- Describe the structure of a shell script
- Create a simple shell script
- Implement the various debugging options in a shell script

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Practice 3 Overview: Shell Scripting

This practice covers the following topics:

- Reviewing Shell Scripts
- Writing and Debugging Shell Scripts

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.