

Shell Programming

A Necessity for all Programmers

Shell Variables

Nagesh Karmali | Firuza Karmali



Department of Computer Science and Engineering
IIT Bombay

You will learn to ...

- Create, Use, and Destroy variables
- Understand what are environment and shell variables

- can hold one value at a time
- name-value pairs
- Example
mobile=lenovo
Defines a variable **mobile** and assigns the value lenovo to it
- Values are accessed using **\$**. Not used to define the variable
echo \$mobile

- Naming rules. Can contain
 - letters: a to z, A to Z
 - numbers: 0 to 9
 - underscore character _
 - *but cannot start with a number

Invalid Names

- 3_Plus_5_Equals_8
- !lenovo
- *mobile
- mobile-cover

Valid Names

- _3_Plus_5_Equals_8
- not_lenovo
- star_mobile
- mobile_cover

- Can hold multiple values
- Used for grouping a set of variables
- Index starts from 0
- Creating Array Variables
array_name[index]=value
- Example: **devices=(HARD_DISK CDROM PENDRIVE)**
- Accessing array Variables (use * or @ to access all):
 - **echo \${devices[0]}** — prints first array element, HARD_DISK
 - **echo \${devices[*]}** — prints all the array elements
 - **echo \${devices[@]}** — same as above
- * treats all the array values as one variable
- @ treats all the array variables as multiple variables

- Use readonly command
- Example 1
 - **readonly device_path="/dev"**
 - Access value: **echo \$device_path**
 - Change value: **device_path="/device"** — ERROR
- Example 2
 - **readonly devices=(HARD_DISK CDROM PENDRIVE)**
 - Access value: **echo \$devices[*]**
 - Change value: **devices[3]=Floppy** — ERROR

- **unset <variableName>**
- Examples
 - mobile=lenovo
unset mobile
 - devices=(HARD_DISK CDROM PENDRIVE)
unset devices

- Describes the environment
- Defined in the shell are small subset of variables
- Program is passed a set of environment variables
- Named object used by one or more applications
- Example: sharing of configuration settings across applications
- `export APPPATH=/bin:/usr/bin`
- `export APPPATH=/bin APPHOME=/home`
- List of Environment variables: **env**

- Sets variables during shell initialization
- Some shell variables. Use echo to check, ex.

echo \$PWD

- PWD
- UID
- SHLVL
- RANDOM
- SECONDS
- PATH
- HOME
- IFS

Variables	Comparison
Local	set by users and not accessible by child processes
Environment	set by users and accessible by child processes
Shell	set by the shell and accessible by child processes

- IFS is a special shell variable
- Used for splitting lines into words based on the delimiter
- Let us demonstrate with an example

```
#!/bin/bash
IFS='|'
read Field1 Field2 Field3
printf "%-10s%-10s%12.2f\n" "$Field1" "$Field2" "$Field3"
```

Now, you should be able to ...

- create, use, and destroy variables
- understand what are local, environment, and shell variables

Thank you

