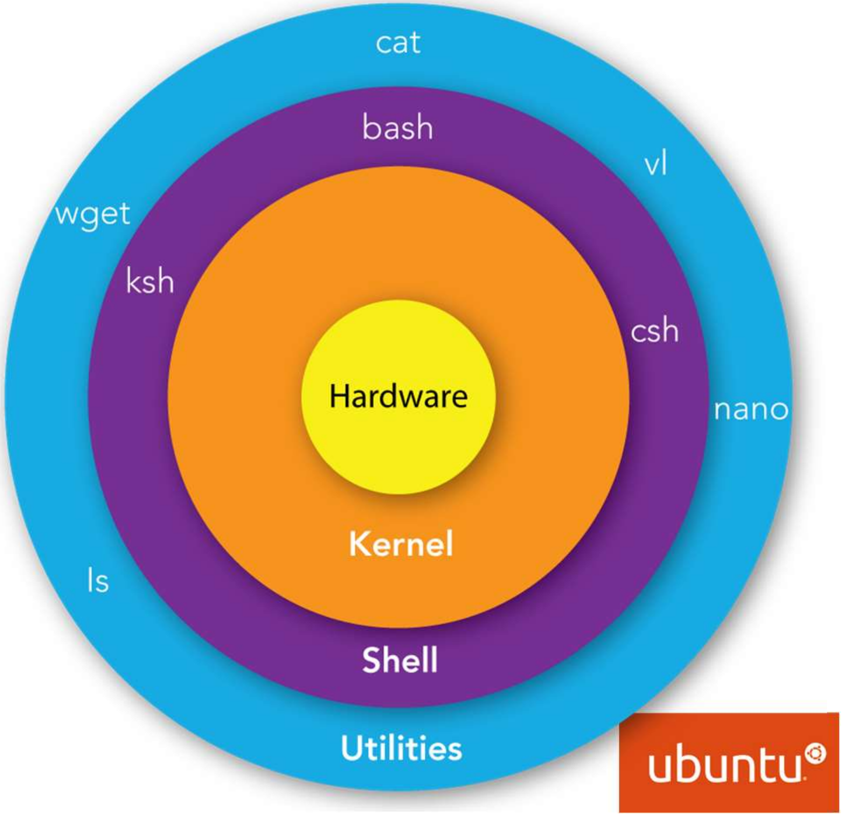
Linux Bash Shell

Bok, JongSoon
javaexpert@nate.com
https://github.com/swacademy/fss/tree/main/Linux

Linux Shell

- Accepts and interprets commands.
- Is the environment in which commands, programs, and shell scripts are executed.
- There are several Linux shells available.

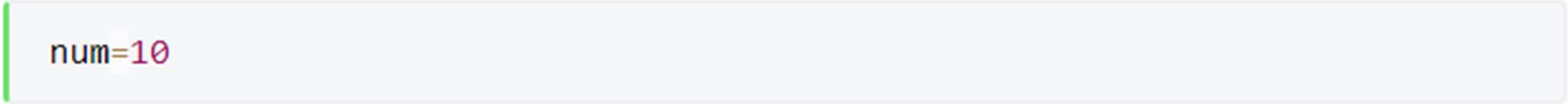


https://velog.io/@jewon119/01.%EB%A6%AC%EB%88%85%EC%8A%A4-%EA%B8%B0%EC%B4%88-%EC%89%98Shell

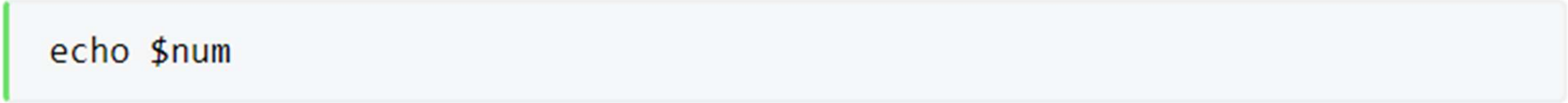
Bourne Again Shell: Bash

- Bash is the Linux default shell.
- Provides an efficient environment for operating system, scripting and interaction.

Shell Variable



```
num=10
```

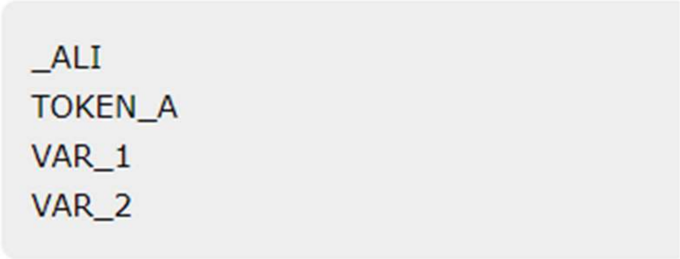


```
echo $num
```

- Variables are used by the shell to store values.
- Variable values can be strings, numbers, or special characters.
- Variables are set to default values is a string.
- Refer to https://www.tutorialspoint.com/unix/shell_scripting.htm

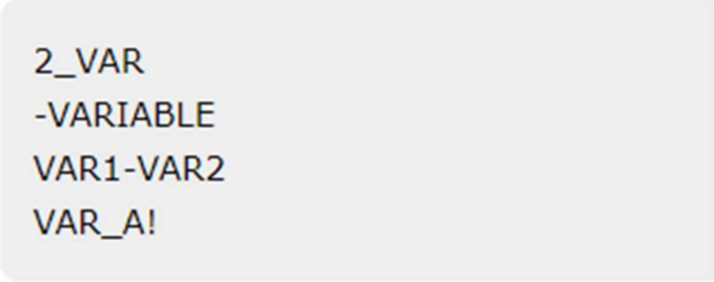
Variable Names

- The name of a variable can contain only letters (a to z or A to Z), numbers (0 to 9) or the underscore character (_).
- By convention, Unix shell variables will have their names in UPPERCASE.
- The following examples are valid variable names:



```
_ALI
TOKEN_A
VAR_1
VAR_2
```

- Following are the examples of invalid variable names:



```
2_VAR
-VARIABLE
VAR1-VAR2
VAR_A!
```

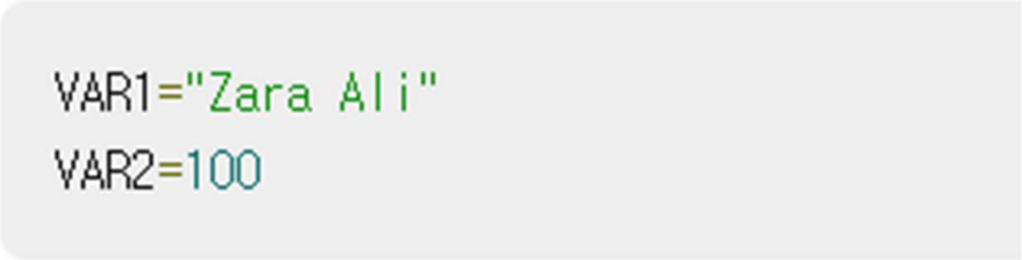
Defining Variables

- Variables are defined as follows :

```
variable_name = variable_value
```

- For example:

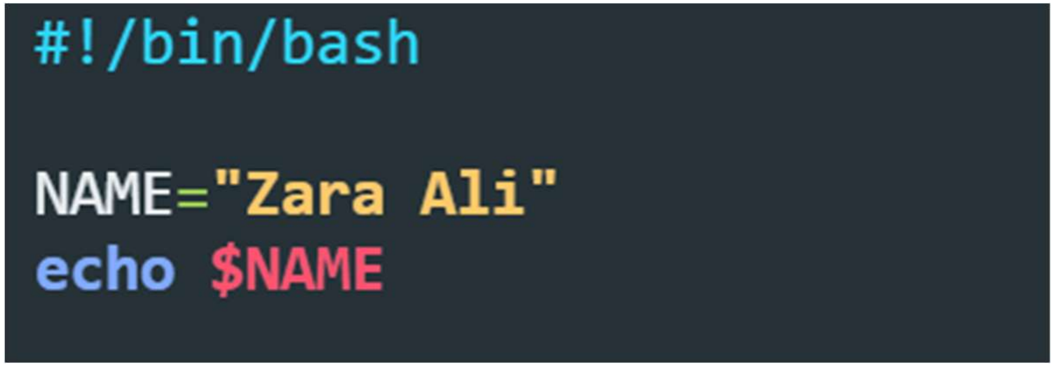
```
NAME=“Zara Ali”
```



```
VAR1="Zara Ali"
VAR2=100
```

Accessing Values

- To access the value stored in a variable, prefix its name with the dollar sign($).

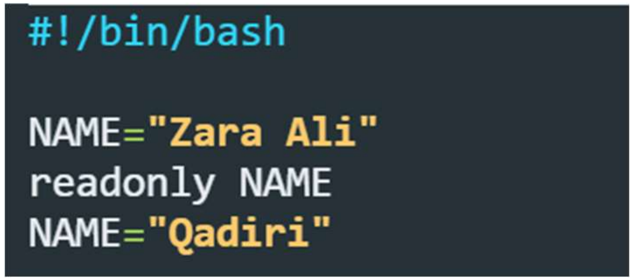


```
#!/bin/bash

NAME="Zara Ali"
echo $NAME
```

Read-only Values

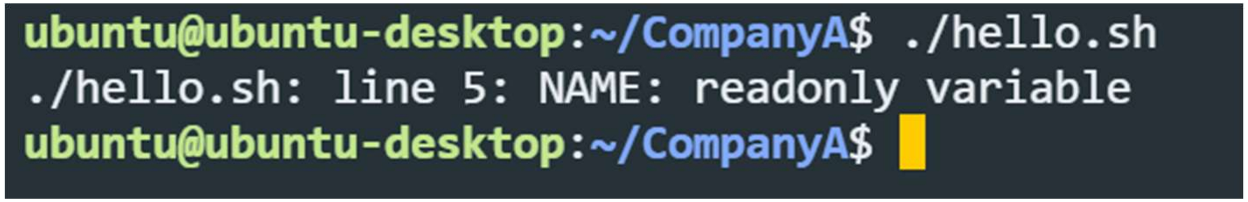
- Shell provides a way to mark variables as ***read-only*** by using the read-only command.
- After a variable is marked read-only, its value cannot be changed.
- For example, the following script generates an error while trying to change the value of **NAME**.



```
#!/bin/bash

NAME="Zara Ali"
readonly NAME
NAME="Qadiri"
```

- The above script will generate the following result:

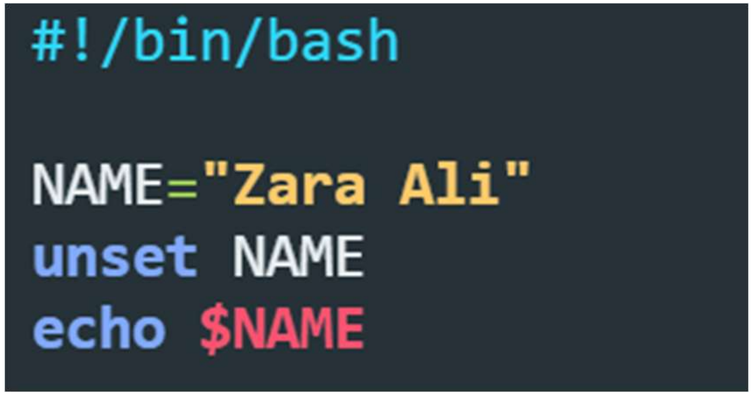


```
ubuntu@ubuntu-desktop:~/CompanyA$ ./hello.sh
./hello.sh: line 5: NAME: readonly variable
ubuntu@ubuntu-desktop:~/CompanyA$ 
```

Unsetting Variables

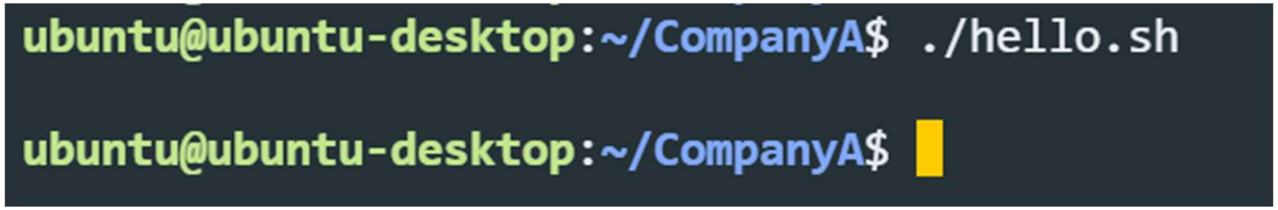
- Unsetting or deleting a variable directs the shell to remove the variable from the list of variables that it tracks.
- Once you unset a variable, you cannot access the stored value in the variable.
- Following is the syntax to unset a defined variable using the **unset** command:

unset variable_name



```
#!/bin/bash

NAME="Zara Ali"
unset NAME
echo $NAME
```



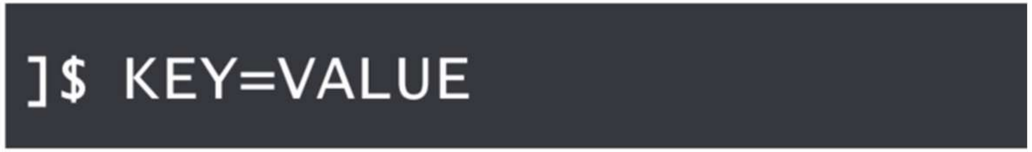
```
ubuntu@ubuntu-desktop:~/CompanyA$ ./hello.sh

ubuntu@ubuntu-desktop:~/CompanyA$ 
```

Variable Types

- Local Variables
  - Is present within the current instance of the shell.
  - Is not available to programs that are started by the shell.
  - They are set at the command prompt.
- Environment Variables
  - Is available to any child process of the shell.
  - Some programs need environment variables in order to function correctly.
- Shell Variables
  - Is a special variable that is set by the shell and is required by the shell in order to function correctly.
  - Some of these variables are environment variables whereas others are local variables.

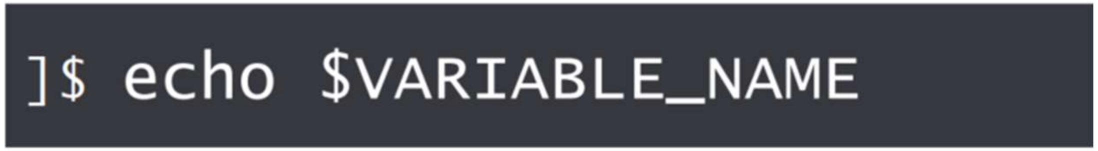
Environment Variables



```
]$ KEY=VALUE
```

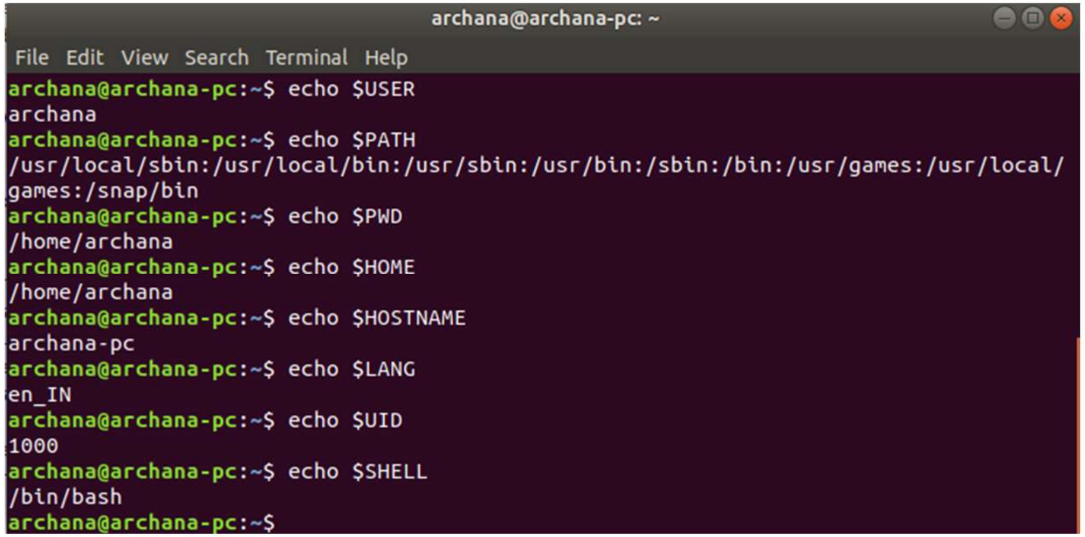
- In the shell, environment variables are the same as shell variables.
- Structurally these variables do not differ from each other.
- Both use key-value pairs, separated by an equal sign (=).

Environment Variables (Cont.)



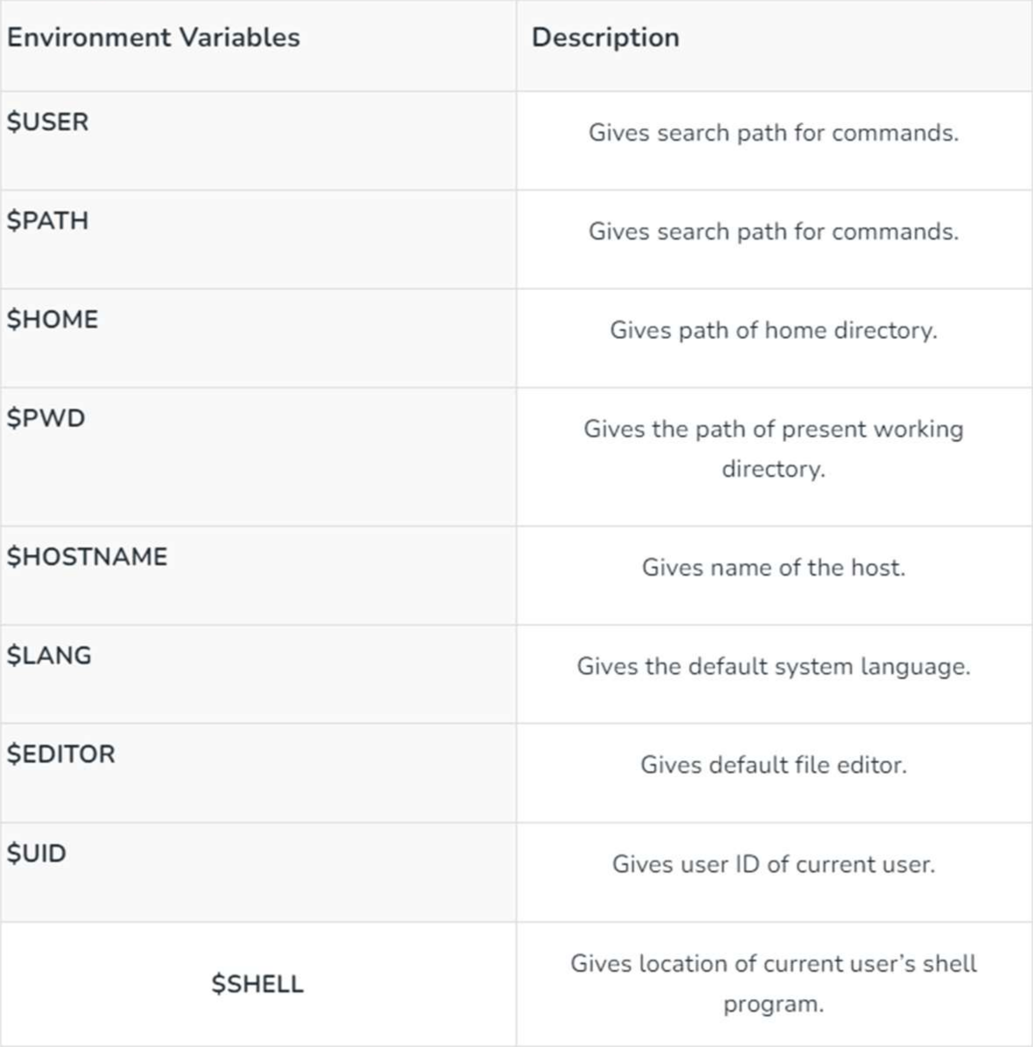
```
]$ echo $VARIABLE_NAME
```

- The echo command is used to view environment variables.



```
archana@archana-pc:~$ echo $USER
archana
archana@archana-pc:~$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/
games:/snap/bin
archana@archana-pc:~$ echo $PWD
/home/archana
archana@archana-pc:~$ echo $HOME
/home/archana
archana@archana-pc:~$ echo $HOSTNAME
archana-pc
archana@archana-pc:~$ echo $LANG
en_IN
archana@archana-pc:~$ echo $UID
1000
archana@archana-pc:~$ echo $SHELL
/bin/bash
archana@archana-pc:~$
```

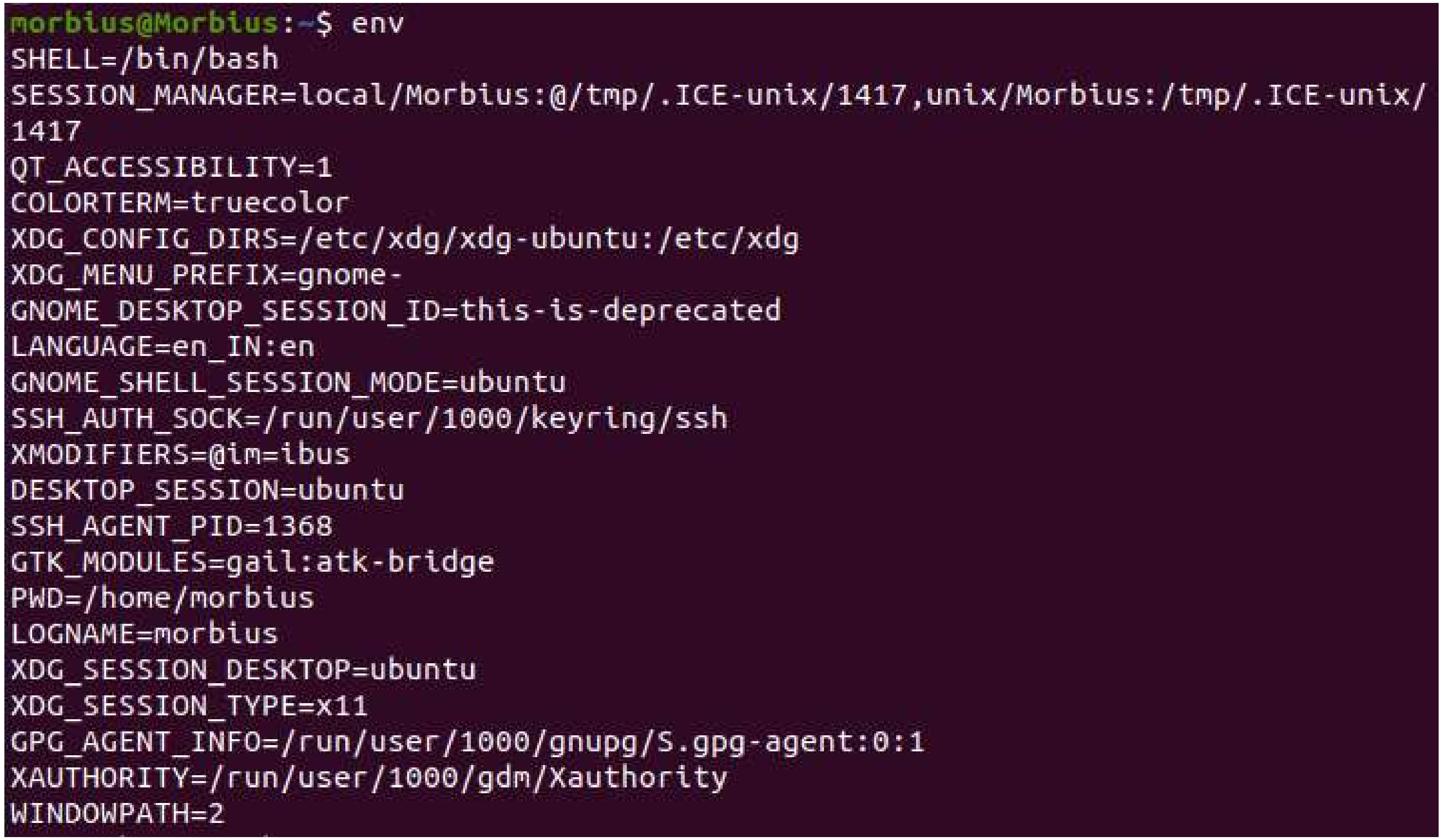
https://www.geeksforgeeks.org/environment-variables-in-linux-unix/



Environment Variables	Description
$USER	Gives search path for commands.
$PATH	Gives search path for commands.
$HOME	Gives path of home directory.
$PWD	Gives the path of present working directory.
$HOSTNAME	Gives name of the host.
$LANG	Gives the default system language.
$EDITOR	Gives default file editor.
$UID	Gives user ID of current user.
$SHELL	Gives location of current user's shell program.

env Command

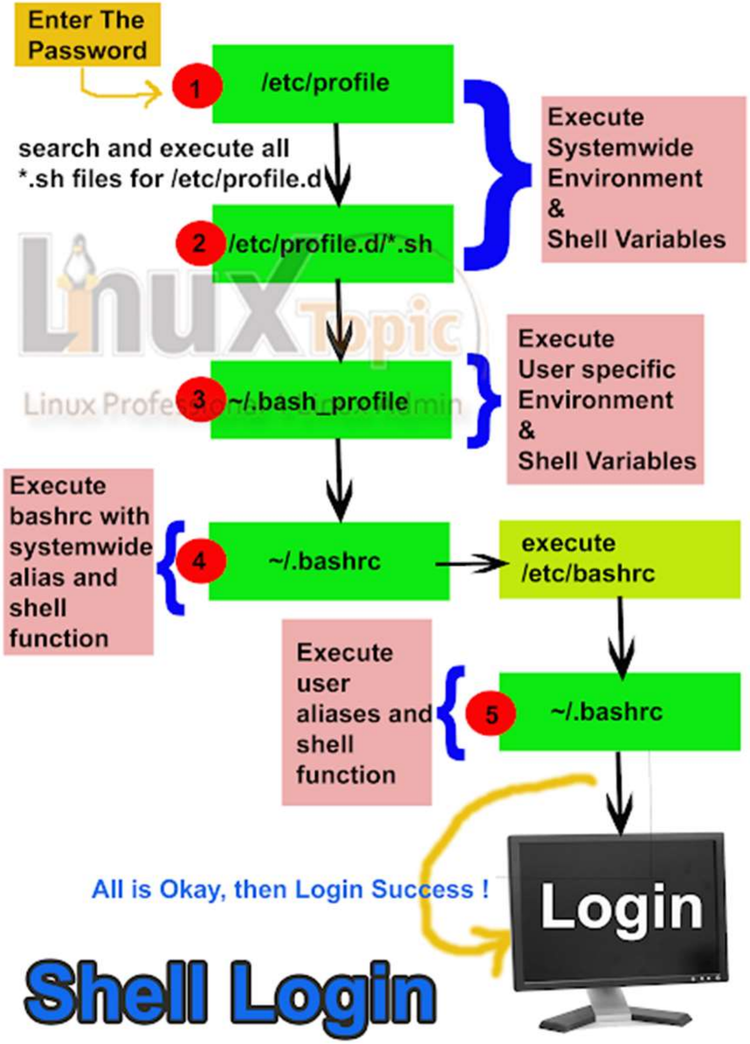
- Is used to view environment variables.



```
morbius@Morbius:~$ env
SHELL=/bin/bash
SESSION_MANAGER=local/Morbius:@/tmp/.ICE-unix/1417,unix/Morbius:/tmp/.ICE-unix/
1417
QT_ACCESSIBILITY=1
COLORTERM=truecolor
XDG_CONFIG_DIRS=/etc/xdg/xdg-ubuntu:/etc/xdg
XDG_MENU_PREFIX=gnome-
GNOME_DESKTOP_SESSION_ID=this-is-deprecated
LANGUAGE=en_IN:en
GNOME_SHELL_SESSION_MODE=ubuntu
SSH_AUTH_SOCK=/run/user/1000/keyring/ssh
XMODIFIERS=@im=ibus
DESKTOP_SESSION=ubuntu
SSH_AGENT_PID=1368
GTK_MODULES=gail:atk-bridge
PWD=/home/morbius
LOGNAME=morbius
XDG_SESSION_DESKTOP=ubuntu
XDG_SESSION_TYPE=x11
GPG_AGENT_INFO=/run/user/1000/gnupg/S.gpg-agent:0:1
XAUTHORITY=/run/user/1000/gdm/Xauthority
WINDOWPATH=2
```

https://linuxhint.com/bash-print-all-environment-variables-values/

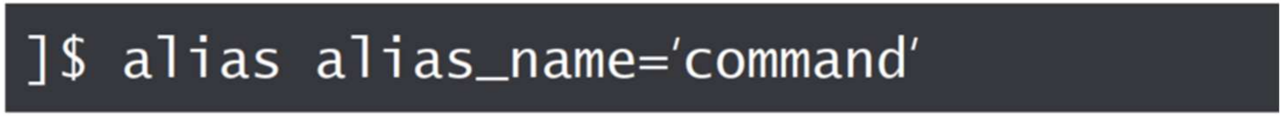
Bash Environment File Initialization Process


Enter The Password
1
/etc/profile
search and execute all *.sh files for /etc/profile.d
2
/etc/profile.d/*.sh
Execute Systemwide Environment & Shell Variables
3
~/.bash_profile
Execute User specific Environment & Shell Variables
Execute bashrc with systemwide alias and shell function
4
~/.bashrc
execute /etc/bashrc
Execute user aliases and shell function
5
~/.bashrc
All is Okay, then Login Success !
Login
Shell Login
LinuxTopic
Linux Professional's Sysadmin

https://www.linuxtopic.com/2020/04/shell-login.html

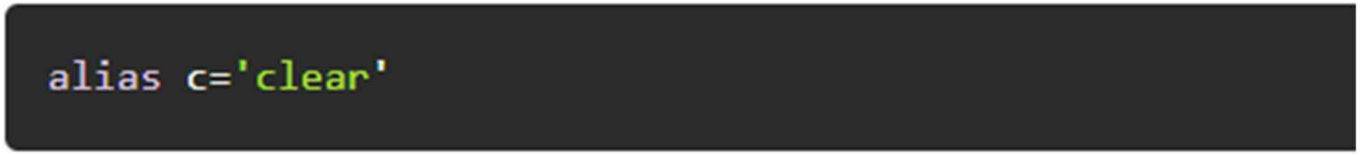
Aliases

- Allow to define new commands by replacing long commands with shorter ones.

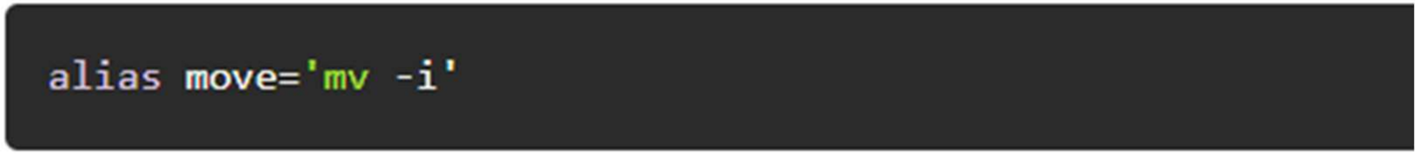


```
]$ alias alias_name=’command’
```

- Create alias Command in Linux temporary



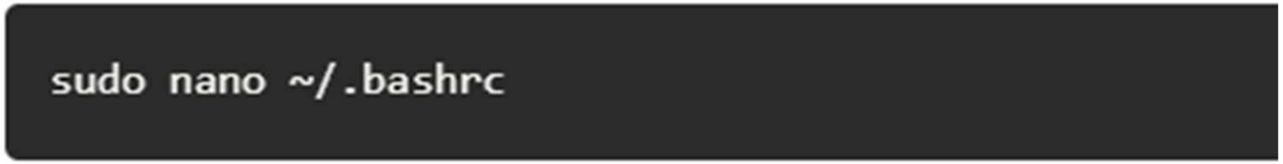
```
alias c='clear'
```



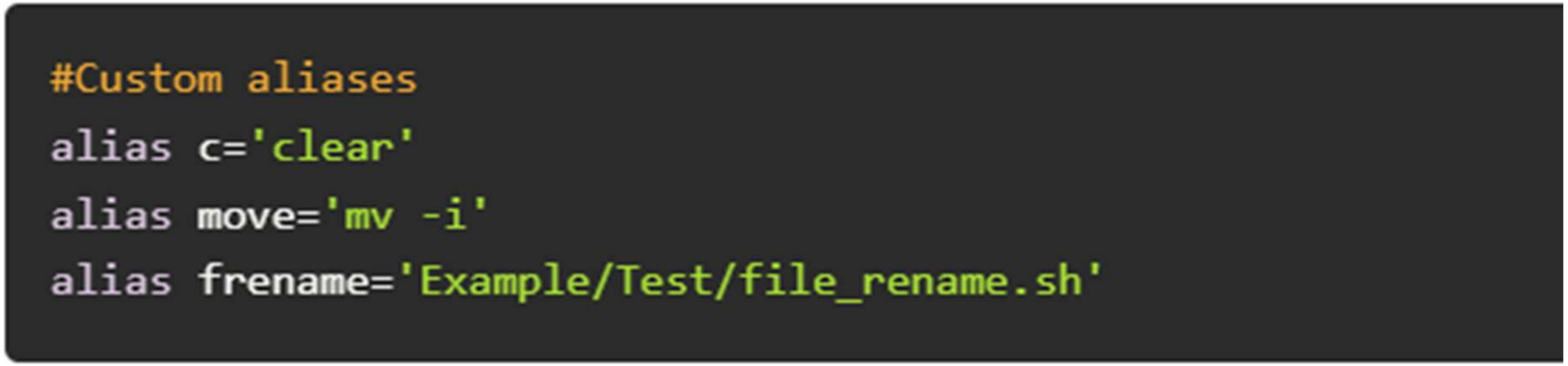
```
alias move='mv -i'
```

Aliases (Cont.)

- Create alias Command in Linux permanently.
  - Open the shell configuration file in a text editor to get started.
  - The Bash shell and the Nano text editor are used in this example:



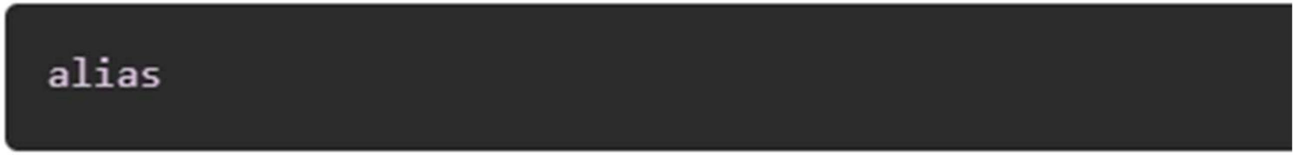
```
sudo nano ~/.bashrc
```



```
#Custom aliases
alias c='clear'
alias move='mv -i'
alias frename='Example/Test/file_rename.sh'
```

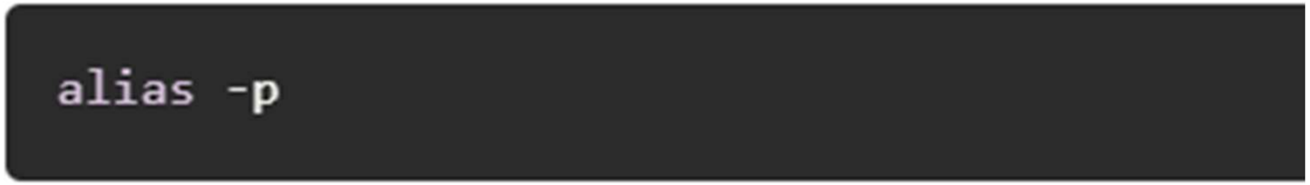
Aliases (Cont.)

- List all aliases in Linux
  - The list of all presently set aliases is displayed by the alias command on its own:



```
alias
```

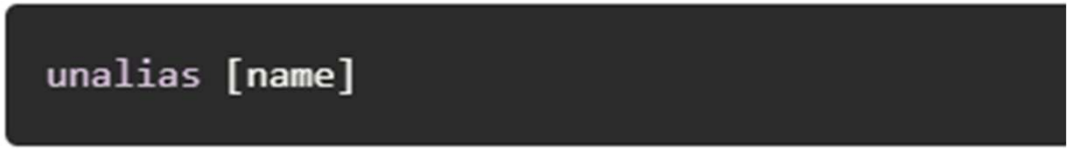
  - To list all alias in Linux, can also add the **-p** flag.
  - Running the following command, can view the list in a format:



```
alias -p
```

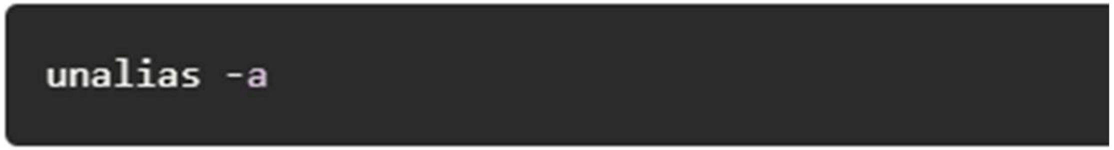
Aliases (Cont.)

- Remove aliases in Linux
  - Use the command below to remove an alias:



```
unalias [name]
```

  - If add the **–a** option, all aliases will be removed:



```
unalias -a
```

Lab. Bash Shell