

EXPORTING/SOURCING

Monday, 3 May 2021

4:35 PM

GROUPING IN BASH: BRACES AND CURLY BRACES

Grouping in Bash

- Bash functions use braces and can modify variables of the shell that calls the function.

```
a=1
(  
a=2  
)  
echo $a  
# prints 1
```

```
a=1  
{  
a=2  
}  
echo $a  
# prints 2
```

BASH STARTUP:

Bash Startup

- `.bash_profile` is read when Bash is invoked as a login shell.
- `.bashrc` is executed when a new shell is started.
- If you extend an exported variable, like `PATH`, in `.bashrc`, it will grow with each nested shell invocation.

Bash Startup

- `PATH=$PATH:/usr/local/bin`

This would keep adding `/usr/local/bin` to the end of `PATH` within nested shells.

- Aliases and functions should normally be defined in `.bashrc`.

SOURCE AND ALIAS COMMAND USAGE:

Using "source" command to source variables/functions in the current process/shell.

```
[guest@kismet ch01]$ echo $x
[guest@kismet ch01]$ source ./setx.sh
[guest@kismet ch01]$ echo $x
22
[guest@kismet ch01]$ x=1
[guest@kismet ch01]$ echo $x
1
[guest@kismet ch01]$ . ./setx.sh
[guest@kismet ch01]$ echo $x
22
[guest@kismet ch01]$ █
```

Example of sourcing a variable defined in a file:

```
C02XF4HDJHD3:Learning-Bash-Fundamentals anish.walia$ cat ./source.sh
a=10
C02XF4HDJHD3:Learning-Bash-Fundamentals anish.walia$ echo $a
10
C02XF4HDJHD3:Learning-Bash-Fundamentals anish.walia$ bash
bash-5.0$ echo $a

bash-5.0$ . ./source.sh
bash-5.0$ echo $a
10
```


USING ECHO:

Using the Echo Command

- Built into Bash and doesn't start a new process
 - n → don't print a trailing newline
 - e → enable backslash escaped characters like \n and \t
 - E → disable backslash escaped characters in case they were enabled by default

