Shell

Definition

Command line interpreter program that functions as the default interface when you connect to a server over a network

bash

Default shell on Ubuntu and OSX. "Bourne-again shell."

Terminal Prompt

`user>@host:path$`

`user`	Logged in user name
`host`	Machine name
`path`	File system position
`$`	Symbol (e.g. `$` bash)

Commands

About

`cmd -help`	Full usage for `cmd`
`whatis cmd`	One-line explanation of `cmd`
`man cmd`	Manual for `cmd` if available
`info cmd`	Info about `cmd`; use if `man` is unavailable
`str [2x tab]`	List commands with `str` prefix

History

`history n`	Show last `n` commands
`!!`	Repeat last command
`!str`	Repeat last command with `str` prefix
`up arrow`	Scroll through history list

Directing Output

`cmd >> file`	Append output of `cmd` to `file`
`cmd > file`	Overwrite `file` with output of `cmd`
`cmd1	cmd2 ...`	Direct output of `cmd1` to `cmd2`

Example

```
$ echo "My new note" >> notes.txt
$ cat notes.txt
# My new note
```

Manuals

`man cmd`			
`-k key`	search for keyword `key`		

Navigate

`h`	Help	`q`	Quit
`u`	Up	`d`	Down
`g`	Beginning	`G`	End

Search

`/pattern`	Start	`esc-u`	Stop
`n`	Next	`N`	Prev

Directory Hierarchy

Absolute Path

Route to desired dir. Always starts w slash or tilde, e.g. `/home/john/myfile.txt`

Relative Path

From working dir. Never starts w slash, e.g. `john/myfile.txt`

Shortcuts

`..`	Path of parent dir of current working dir
`.`	
`~`	Path of home dir, e.g. `$ echo ~` `# user@host:/home/user/`
`~user`	Another user's home dir
`/`	Root

Environmental Variables

`$OLDPWD`	
`$PATH`	Ordered, colon-delimited list of dirs in search when call is made from working dir

Parsing

Colon-delimited; use `tr` to replace, e.g.
`$ echo $PATH | tr ":" "\n"`

By **birdbysnow**
cheatography.com/birdbysnow/

Not published yet.
Last updated 12th July, 2022.
Page 1 of 3.

Sponsored by **Readable.com**
Measure your website readability!
https://readable.com

Directory Hierarchy (cont)

	Home Directory	Home dir (e.g. `/home/john`) is never in path, but its subdirs may be if working dir is a subdir of home.
`$PYTHONPATH`		

Useful Commands

`pwd`		Print working dir
`cd dir`		Move working dir to `dir`
	`-`	To previous wd (i.e. `$OLDPATH`)
`ls dir`		List `dir` contents
	`-l`	Get a long list - permissions, links count,
	`-F`	Indicate file type * exe / dir
	`-b`	Print special chars, including - `| & ' ; ( ) < > space tab`
	`-a`	All including hidden files with `.` prefix
	`-1`	All output in one column
	`-R`	Recursive; include subdirs and contents
	`-r`	Reverse sort order
	`-t`	Sort by most recent modified
	`-d`	Dirs only

`find dir opts`		Search `dir` and its subdirs for file specified by `opts` expression, e.g.
		`$ find . -name 'chap*'`
		`$ find . -mtime -2`
		`$ find . -user john`

Account

Configuration

Update hidden programs run at login/shell start.

`.bashrc`	
`.bash_profile`	Run at bash login before `.profile`
`.profile`	Run at login by most UINX shells

Example

Add lib to `$PYTHONPATH` in `~/.bashrc` with
`export PYTHONPATH=/home/john/module/lib$PYTHONPATH`

Useful Commands

`ssh user@host`	connect to `host` as `user`
`whoami`	who you are logged in as
`passwd`	set password
`exit`	logout of session and close terminal

Permissions

Useful Commands

`chmod`			
	Usage		
	`$ chmod [ugoa][+-=][rwx], …, [path]`		
	Groups		
	User owner	**o**	Other/system
	All		
		+	Add
		-	Remove
		=	Set Explicit
	Mode		

By **birdbysnow**

cheatography.com/birdbysnow/

Not published yet.
Last updated 12th July, 2022.
Page 2 of 3.

Permissions (cont)

`r, 4`	Read		`x, 1`	Execute file	
`w, 2`	Write		`x, 1`	Access dir	

Octal

`0`	`---`		`4`	`r--`
	`--x`	`5`	`r-x`	
`2`	`-w-`		`6`	`rw-`
`3`	`-wx`		`7`	`rwx`

Examples

Both the following examples grant all permissions to the owner of bin and read + execute permissions to the owner's group and others.

```
$ chmod u=rwx,go=rx /home/john/bin
$ chmod 755 /home/john/bin
```