

Pipes and filters

Filters are shell commands, which:

- Take input from standard input
- Send output to standard output
- Transform input data into output data
- Examples are wc, cat, more, head, sort, ...
- Filters can be chained together

IBM Developer

SKILLS NETWORK 

Pipes and filters

Pipe command - |

- For chaining filter commands

```
command1 | command2
```

- Output of command 1 is input of command 2
- Pipe stands for pipeline

```
$ ls | sort -r
Videos
Public
Pictures
Music
Downloads
Documents
Desktop
```

IBM Developer

SKILLS NETWORK 

Shell variables

- Scope limited to shell
- Set - list all shell variables

```
$ set | head -4
BASH=/usr/bin/bash
BASHOPTS=checkwinsize:cm
  dhist:complete_fullquot
  e:expand_aliases:extglo
  b:extquote:force_ignor
  e:globasciiranges:hista
  ppend:interactive_comme
  nts:progcomp:promptvars
  :sourcepath
BASH_ALIASES=()
BASH_ARGC=( [0]="0")
```

IBM Developer

SKILLS NETWORK 

Defining shell variables

```
var_name=value
```

- No spaces around '='

```
unset var_name
```

- deletes var_name

```
$ GREETINGS="Hello"
$ echo $GREETINGS
Hello

$ AUDIENCE='World'
$ echo $GREETINGS $AUDIENCE
Hello World

$ unset AUDIENCE
```

IBM Developer

SKILLS NETWORK 

Environment variables

- Extended scope

```
export var_name
```

- `env` - list all environment variables

```
$ export GREETINGS
```

```
$ env | grep "GREE"
```

```
$ GREETINGS=Hello
```

IBM Developer

SKILLS NETWORK 

Summary

In this video, you learned that:

- Filters are shell commands
- The pipe operator allows you to chain filter commands
- Shell variables can be assigned values with '=' and listed using 'set'
- Environment variables are shell variables with extended scope; create with 'export,' list with 'env'

IBM Developer

SKILLS NETWORK 