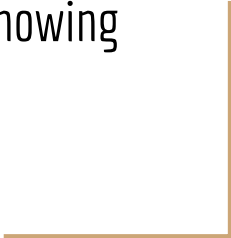




# Introduction to Bash video lecture

04 Moving around and knowing  
where you are



# Moving around

- In the shell, you always have a current directory, which is the starting-point for relative paths
- To change current directory, use `cd`
- Use `pwd` to print the absolute path to where you are

# Changing directory

- `cd` without arguments take you home
- `cd -` (dash) takes you to the last place, meaning that you can toggle between two directories very easily

```
rikard@newdelli:~/opt/progund/intro-it/bash-scripting-intro$ cd
rikard@newdelli:~$ pwd
/home/rikard
rikard@newdelli:~$ cd -
/home/rikard/opt/progund/intro-it/bash-scripting-intro
rikard@newdelli:~/opt/progund/intro-it/bash-scripting-intro$ pwd
/home/rikard/opt/progund/intro-it/bash-scripting-intro
rikard@newdelli:~/opt/progund/intro-it/bash-scripting-intro$
```

# Arguments to cd

- You use paths to tell cd *what* directory to change to
- Absolute paths and relative paths both work
- Typical use of relative path is that you create a new directory and cd to it

```
~$ mkdir bash-exercises
```

```
~$ cd bash-exercises
```

```
~/bash-exercises$ pwd
```

```
/home/rikard/bash-exercises
```

```
~/bash-exercises$
```

# Relative paths

- Since relative paths work fine, you can use `..` or `.` as part of the path

```
rikard@newdelli:~$ cd ..           # go up one directory to /home
rikard@newdelli:~/home$ cd ..      # go up one more directory to /
rikard@newdelli:/$ cd home         # go down to /home/
rikard@newdelli:~/home$ cd rikard/ # go down to /home/rikard
rikard@newdelli:~$ cd ../..        # go two directories up to /
rikard@newdelli:/$ cd home/rikard/ # go down all the way to /home/rikard
rikard@newdelli:~$                # and, you're back!
rikard@newdelli:~$ cd .            # you can do that, but...
rikard@newdelli:~$                # it won't take you anywhere
rikard@newdelli:~$ cd $HOME        # this will take you home
rikard@newdelli:~$ cd              # as will this, which is less to type
```