Fine!

You may use functions to create your own commands in the terminal. To do that, just define functions somewhere in your `~/.bashrc` file (`~/.bash_profile` , `~/.zshrc` for Zsh, etc). For example:

```
# ...
# other ~/.bashrc settings
# ...

# Make directory and jump inside
md() {
  mkdir -p $1 && cd $1
}
```

After that, update your settings using `. ~/.bashrc` and use this command as any other:

```
~ $ md Projects
~/Projects $
```

Aliases

By the way, sometimes you might type a long command sequence to do something. If you often do this, you may want to define an *alias*. An alias is essentially like a keyboard shortcut, a means to avoid typing a long command sequence.

For example, if you often type a `ls -alF` command feel free to define an alias in the `~/.bashrc` file:

```
# ...
# other ~/.bashrc settings
# ...

alias ll='ls -alF'
```

Update your settings using `. ~/.bashrc` and now you can use the `ll` command instead of the longer `ls -alF` command.

Now you have a good basic knowledge of Bash. In the next exercise we will take a look at the debugging of bash scripts.