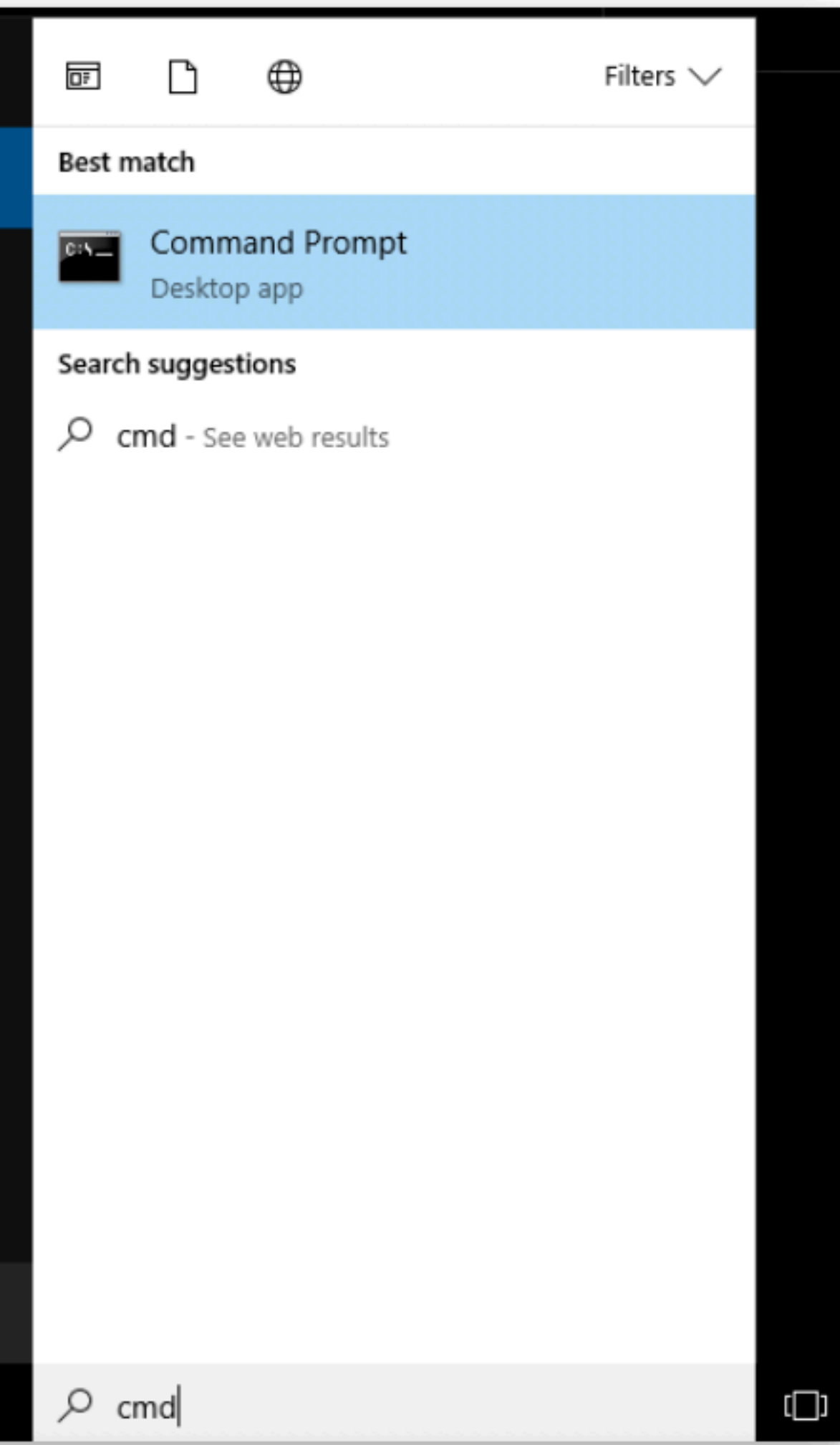




# Command Prompt / Terminal

A screenshot of the Windows Command Prompt window. The title bar reads 'Command Prompt'. The text inside shows the version 'Microsoft Windows [Version 10.0.14393]' and copyright information '(c) 2016 Microsoft Corporation. All rights reserved.'. The user has entered the command 'C:\Users\edele>dir'. The output shows the directory listing for 'C:\Users\edele', including subdirectories like 'Contacts', 'Desktop', 'Documents', 'Downloads', 'Favorites', 'Links', 'Music', 'OneDrive', 'Pictures', 'Saved Games', 'Searches', and 'Videos'. It also shows file statistics: '0 File(s) 0 bytes' and '14 Dir(s) 19,674,497,024 bytes free'. The prompt 'C:\Users\edele>' is visible at the bottom.A screenshot of a Mac Terminal window titled 'java'. The prompt is 'Eamonns-Mac-mini:playdemo edeleastar\$'. The user has entered 'play run'. The output shows a Play framework logo, the version 'play! 1.6.0', the URL 'https://www.playframework.com', and a warning about deprecated options. It then shows the server starting, listening for transport at port 8000 and HTTP on port 9000. The prompt 'Eamonns-Mac-mini:playdemo edeleastar\$' is visible at the bottom.

- This is the Command Line Interface for Windows (CLI)
- In Mac/Linux it is called “Terminal”
- Learning to use the CLI is an **essential** skill for a programmer

# Example DOS Commands

- **dir**
  - list all files in a directory
- **cd ..**
  - change to a parent directory
- **cd <directory name>**
  - change to a specific directory
- **mkdir <directory name>**
  - create a directory
- **rmdir <directory name>**
  - delete a directory
- On Mac/Linux:
  - use **ls** instead of **dir**

- These commands always have a 'current directory' in mind
- A directory is another name for a folder.
- On Windows, the current directory appears in the 'prompt'

A screenshot of a terminal window with a black background and white text. The prompt 'C:\workspace>\_' is visible, indicating the current directory in a Windows command prompt.

- In Mac/Linux, type '**pwd**' to find out the current directory.
- On Windows, '\' or '/' can be used to separate directory names
- On Mac/Linux, only '/' is accepted
- So --> always is '/' to avoid confusion



```
Command Prompt
Microsoft Windows [Version 10.0.17134.523]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Eamonn de Leastar>cd /

C:\>cd dev

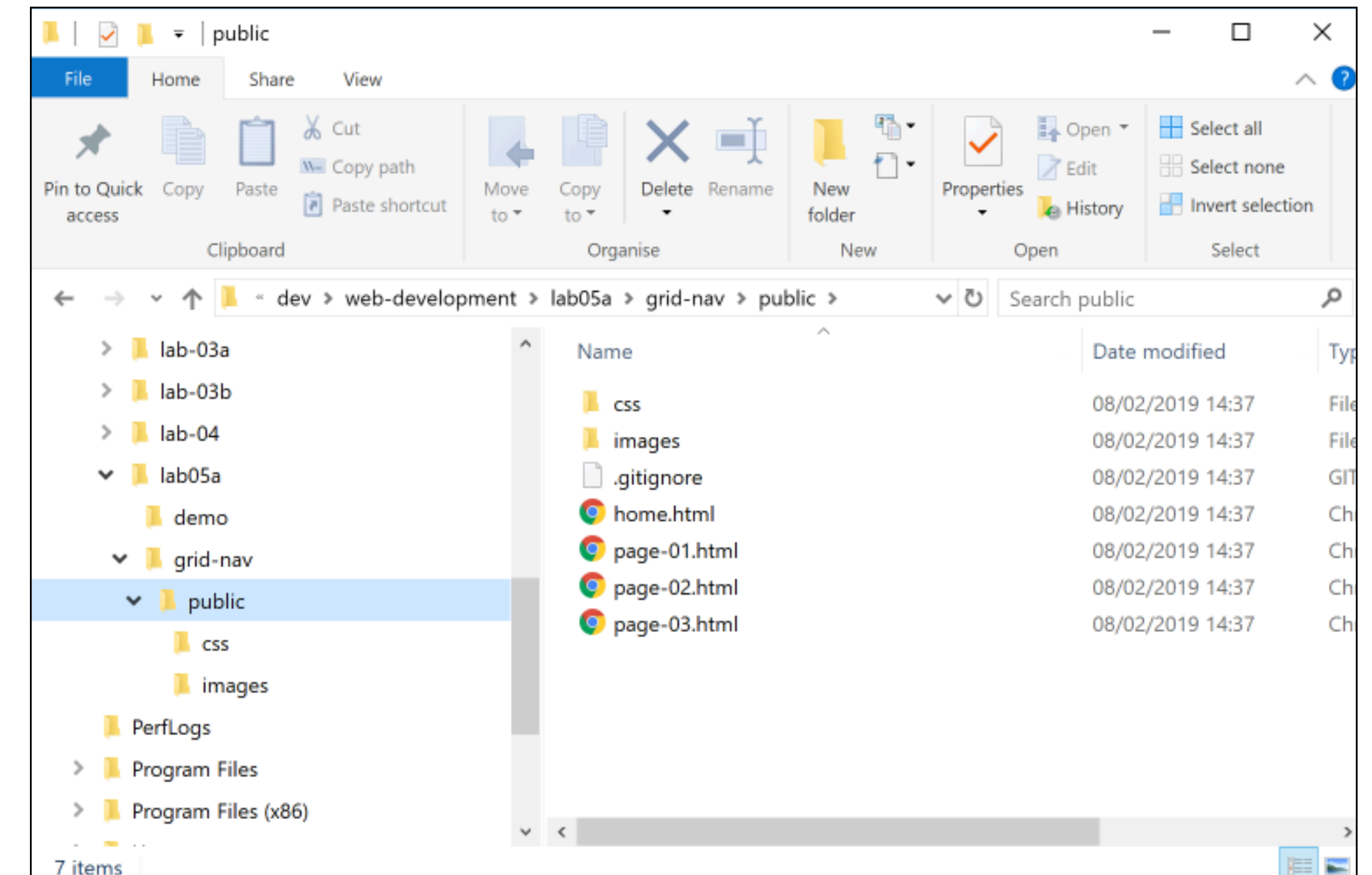
C:\dev>cd web-development

C:\dev\web-development>dir
Volume in drive C has no label.
Volume Serial Number is 52B4-5DE6

Directory of C:\dev\web-development

08/02/2019  12:01    <DIR>        .
08/02/2019  12:01    <DIR>        ..
08/02/2019  12:01    <DIR>        lab-00
08/02/2019  12:01    <DIR>        lab-01
08/02/2019  12:01    <DIR>        lab-02
08/02/2019  12:01    <DIR>        lab-03a
08/02/2019  12:01    <DIR>        lab-03b
08/02/2019  12:01    <DIR>        lab-04
               0 File(s)              0 bytes
               8 Dir(s) 49,239,642,112 bytes free

C:\dev\web-development>
```



*C:\Users\edeleastar>cd ..*

*C:\Users\>cd ..*

*C:\>cd workspace*

*C:\>workspace>dir*

*...*

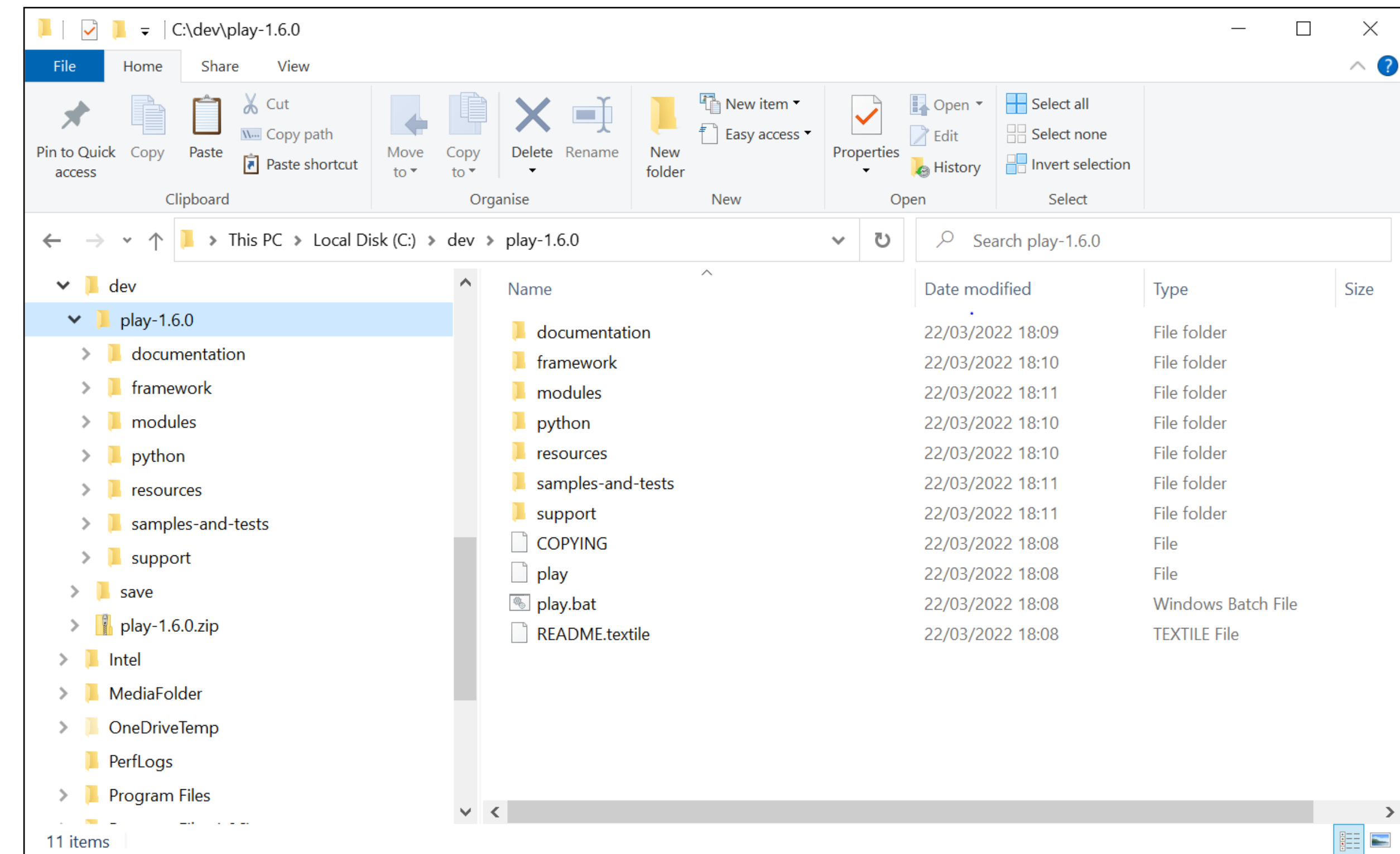
*...*

*C:\>*

- Never use spaces in directory or file names
- Never use upper case in directory or file names
- If you wish to use readable multiple words for a directory or file name, separate the words with '-'. e.g.
  - web-development
  - java-projects

# Installing Play

- Play must be downloaded and installed on your PC
  - [www.playframework.org](http://www.playframework.org)
  - Use version 1.6.0
- Installing Play on your PC is just expanding the zip archive into a folder
- This folder must be placed on the 'System Path'
- This will equip the PC with a new command



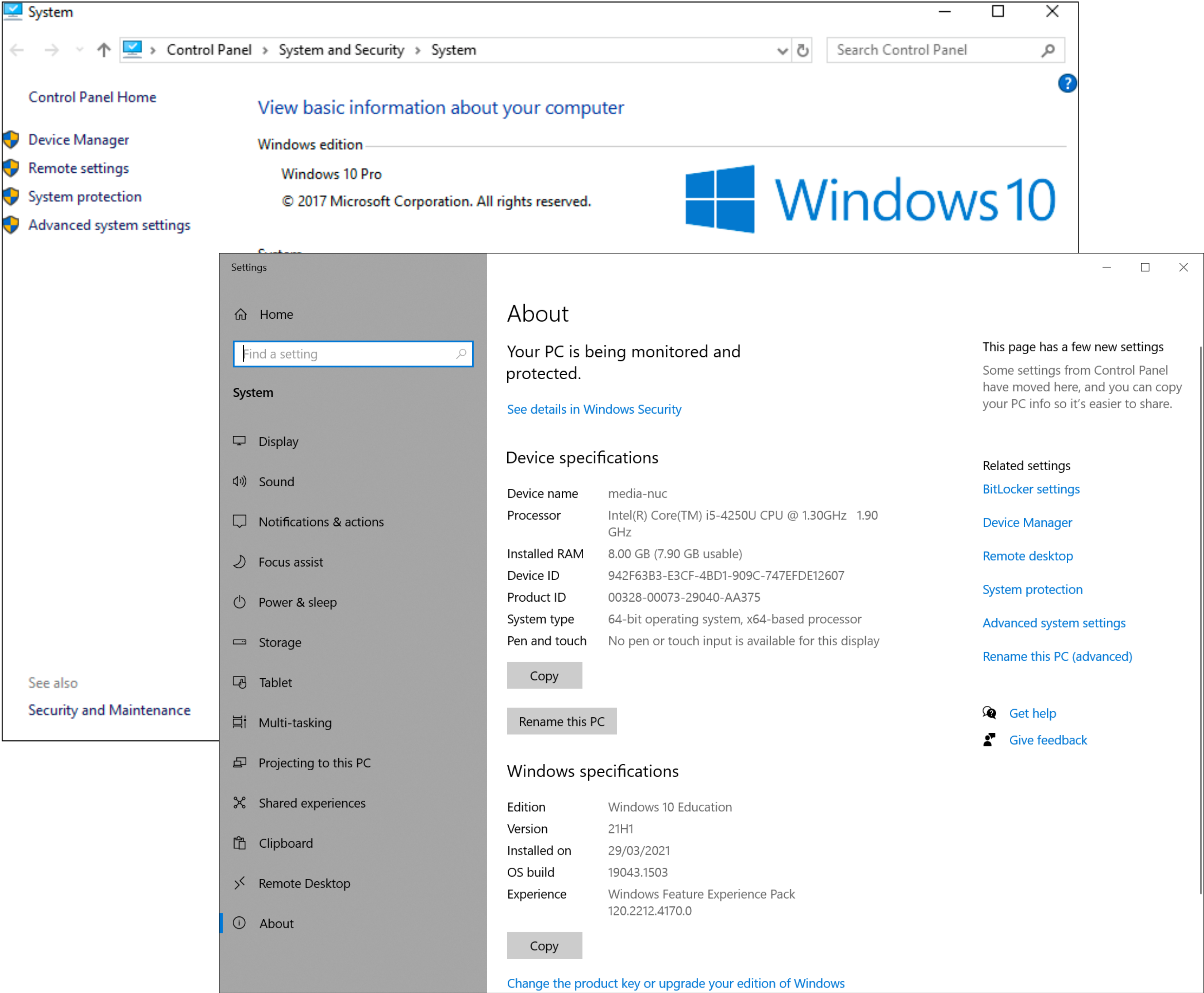
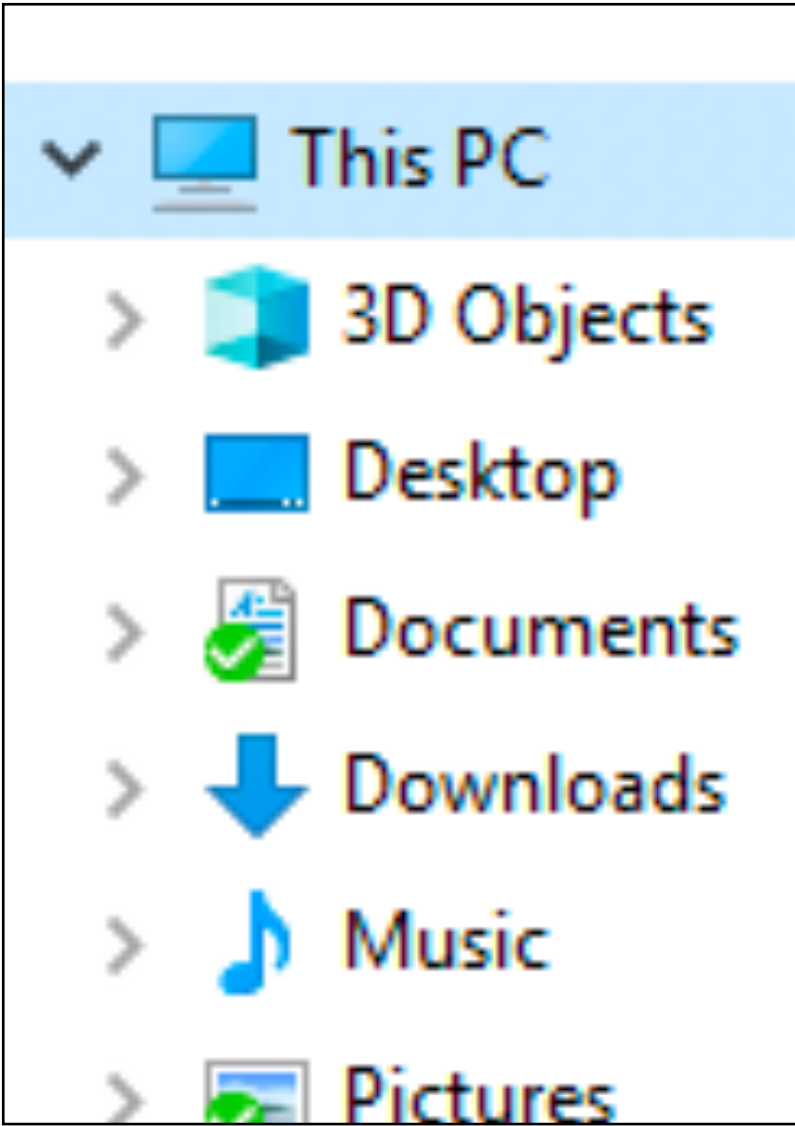
## 1.6 Setup Instructions

play-1.6.0.zip

Mar 15 2021

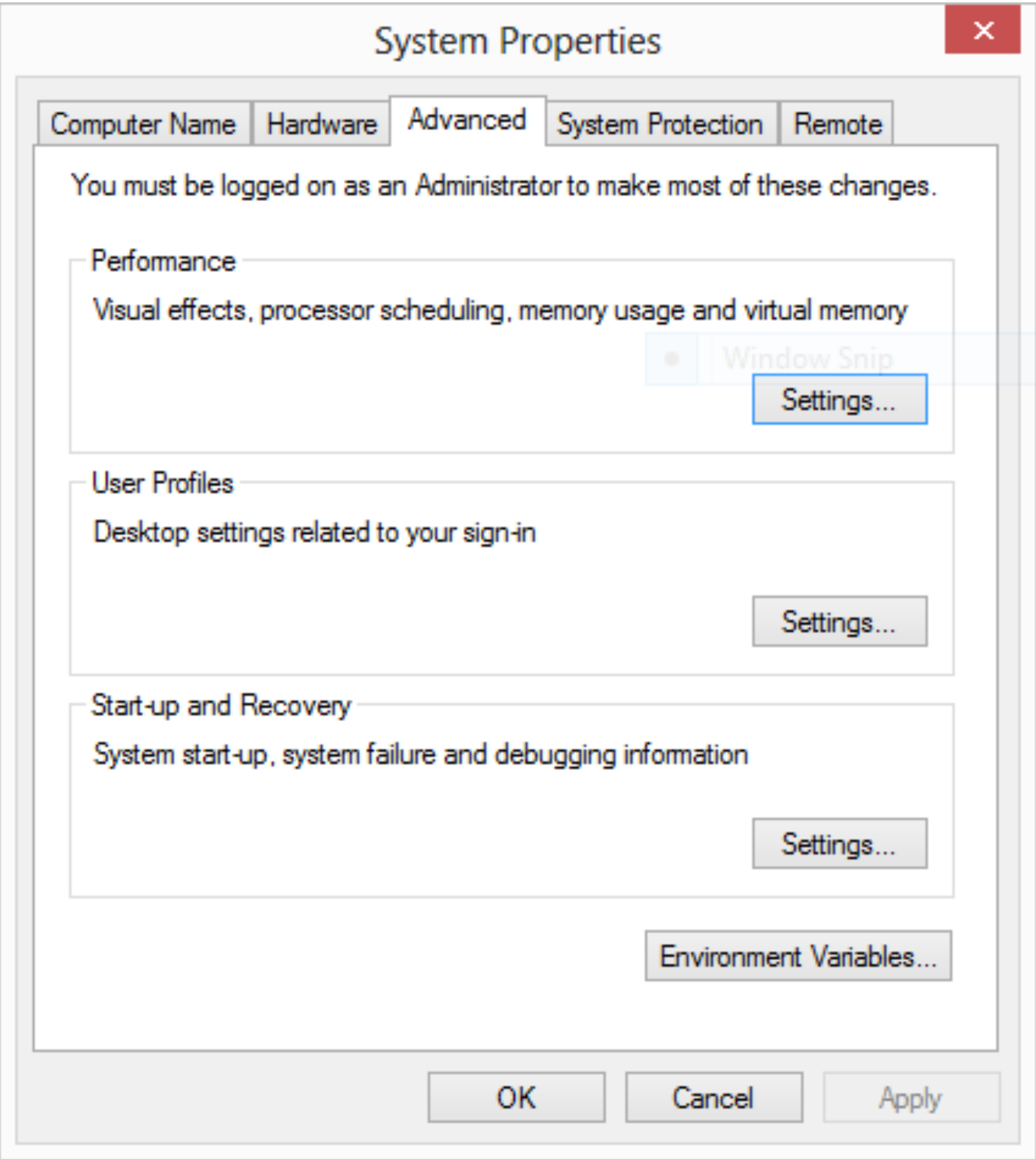
101M

# Path Configuration

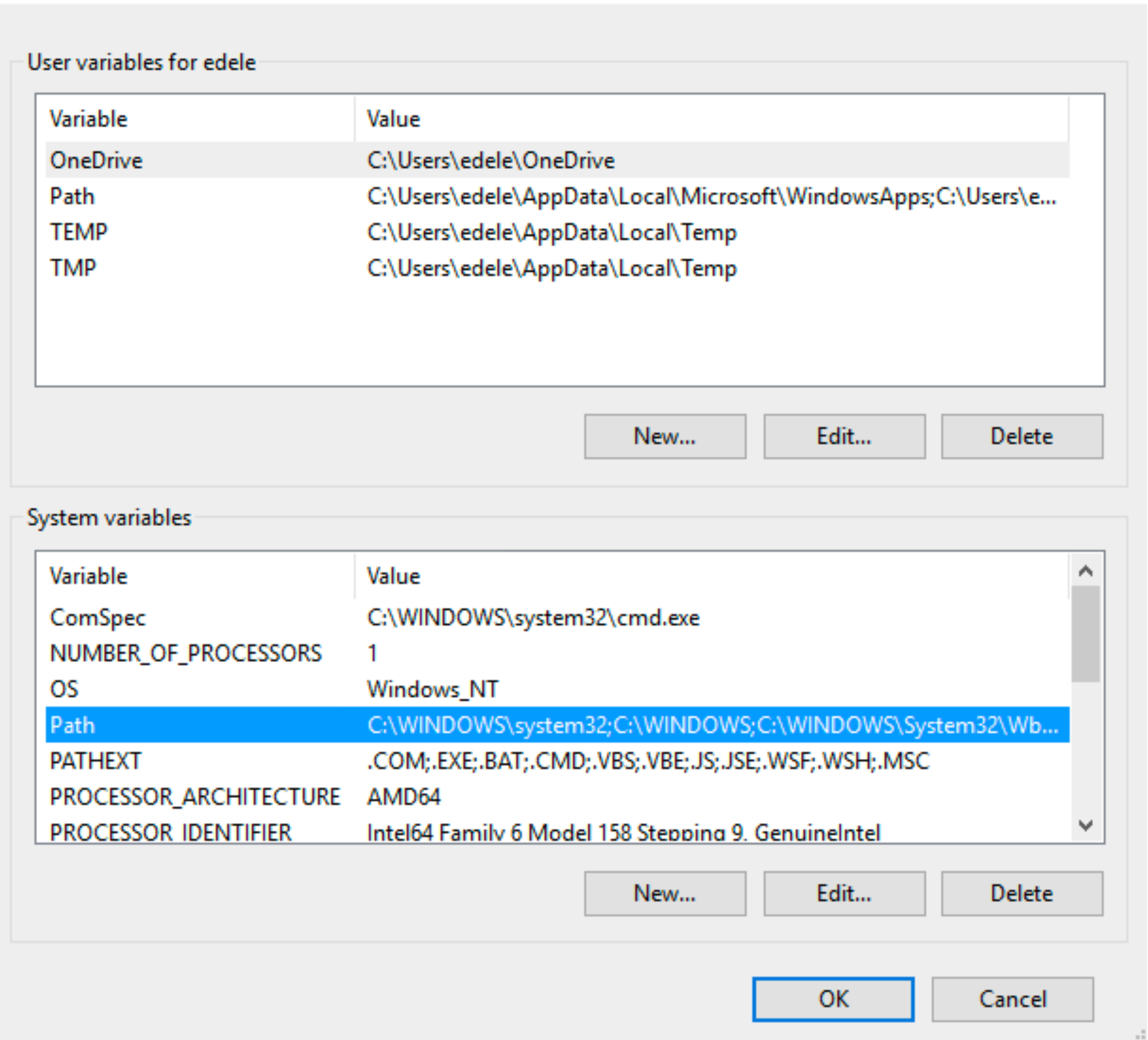


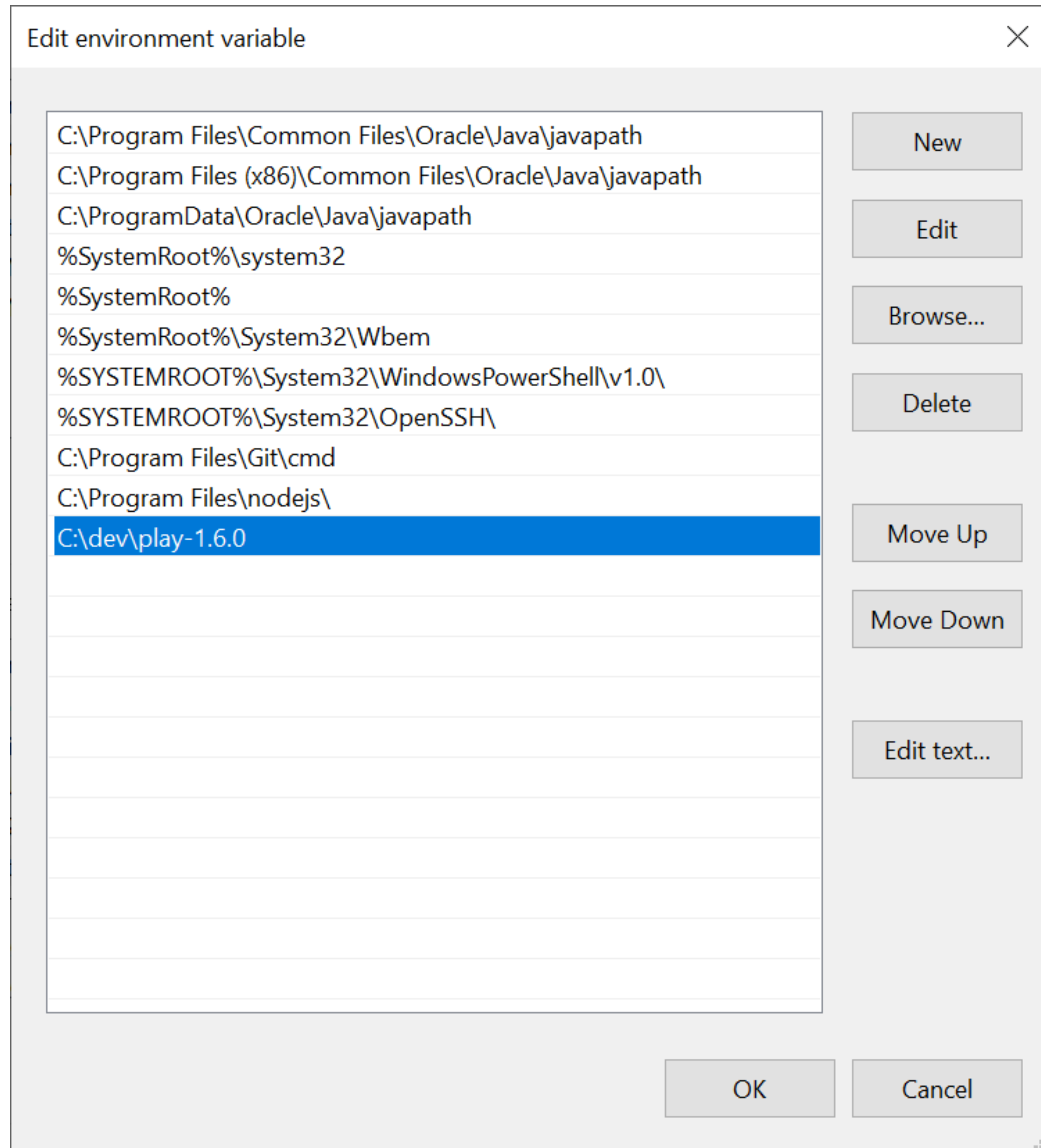


# Path Configuration



Environment Variables





- New entry in system path



# Verifying Play

- Play applications are created and executed using the shell
- You must be able to navigate your folder structure using the shell
- To verify play, just type **play** on command line
  - This works if play is 'on the path'
- If no response (error), then type
  - **c:\dev\play-1.6.0\play**
  - This is an 'explicit path' - ie. a fully qualified path to the play program

```
~
~  _ _ _ | | _ _ _ _ _ | |
~ | ' _ \ | / _ ' | | | _ |
~ | _ _ / | _ \ _ _ _ | \ _ ( _ )
~ | _ | _ _ _ _ _ | _ _ /
~
~
~ play! 1.6.0, http://www.playframework.org
~
~ Usage: play cmd [app_path] [--options]
~
~ with,  new      Create a new application
~       run      Run the application in the current shell
~       help     Show play help
```

# Create a Project

- This command creates a skeleton play application called 'playdemo' in the current directory
- Usually you will want to 'change into' this new directory for subsequent commands

```
c:\dev\>play new playdemo
```

```
c:\dev\>cd playdemo
```

```
c:\dev\playdemo>
```

```
play new playdemo
```

You should get:

```
~
~      -      -
~  _ _ _ | | _ _ _ _ | |
~  | ' _ \ | / _ ' | | | _ |
~  | _ _ / | _ \ _ _ | \ _ _ ( _ )
~  | _ |           | _ _ /
~
~ play! 1.6.0, http://www.playframework.org
~
~ The new application will be created in c:\dev\playdemo
~ What is the application name? [playdemo]
```

Press return when prompted for the name to accept the default (playdemo)

```
~
~ OK, the application is created.
~ Start it with : play run playdemo
~ Have fun!
~
```

# Running the Play Application

- We have generated an “Web App” NOT at “Web Site”
- A Web App requires an ‘Application Server’ to run
- Play has one built-in, so running the server + the web app is a single command

```
play run
```

Play will respond with something like this:

```
~
~  _ _ _ | | _ _ _ _ | |
~ | ' _ \ | / _ ' | | | _ |
~ | _ _ / | _ \ _ _ _ | \ _ _ ( _ )
~ | _ | _ _ _ _ _ | _ _ /
~
~
~ play! 1.6.0, https://www.playframework.com
~
~ Ctrl+C to stop
~
~ using java version "11.0.10"
Listening for transport dt_socket at address: 8000
08:02:25,795 INFO ~ Starting C:\dev\playdemo
:: loading settings :: url = jar:file:/C:/dev/play-1.6.0/framework/lib/ivy-2.4.0.jar!/org/apache/ivy/core/setting
08:02:26,302 INFO ~ Module docviewer is available (C:\dev\play-1.6.0\modules\docviewer)
08:02:27,449 WARN ~ You're running Play! in DEV mode
08:02:27,800 INFO ~ Listening for HTTP on port 9000 (Waiting a first request to start) ...
~ Server is up and running
```



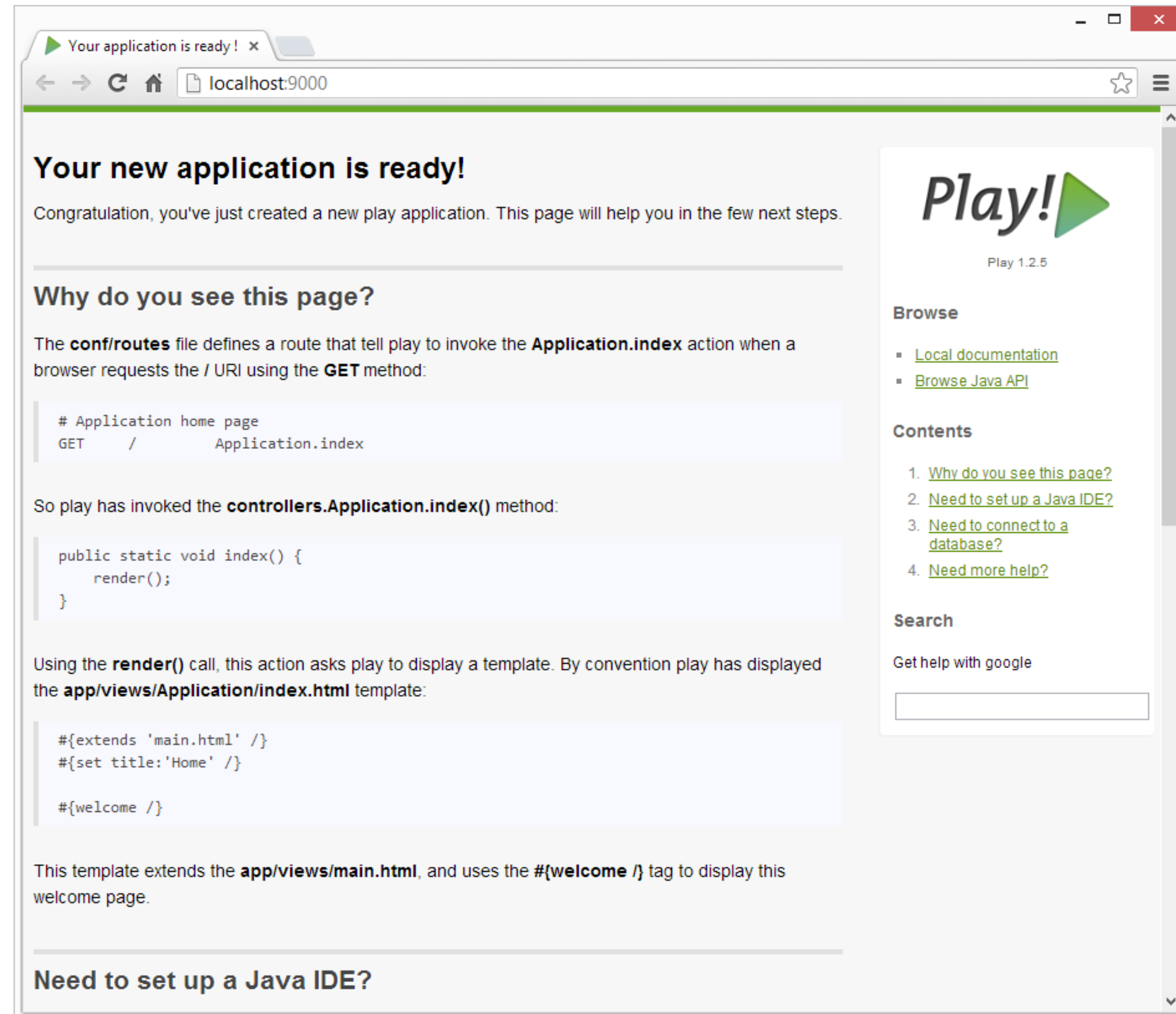
# The Skeleton App:

- The application is now 'hosted' on our local machine (localhost)

- To use it, we browse to:

**http://localhost:9000/**

- 9000 is a random 'port' number in which the app is being 'served'
- The 'default' app is documentation on play itself



# Idealize

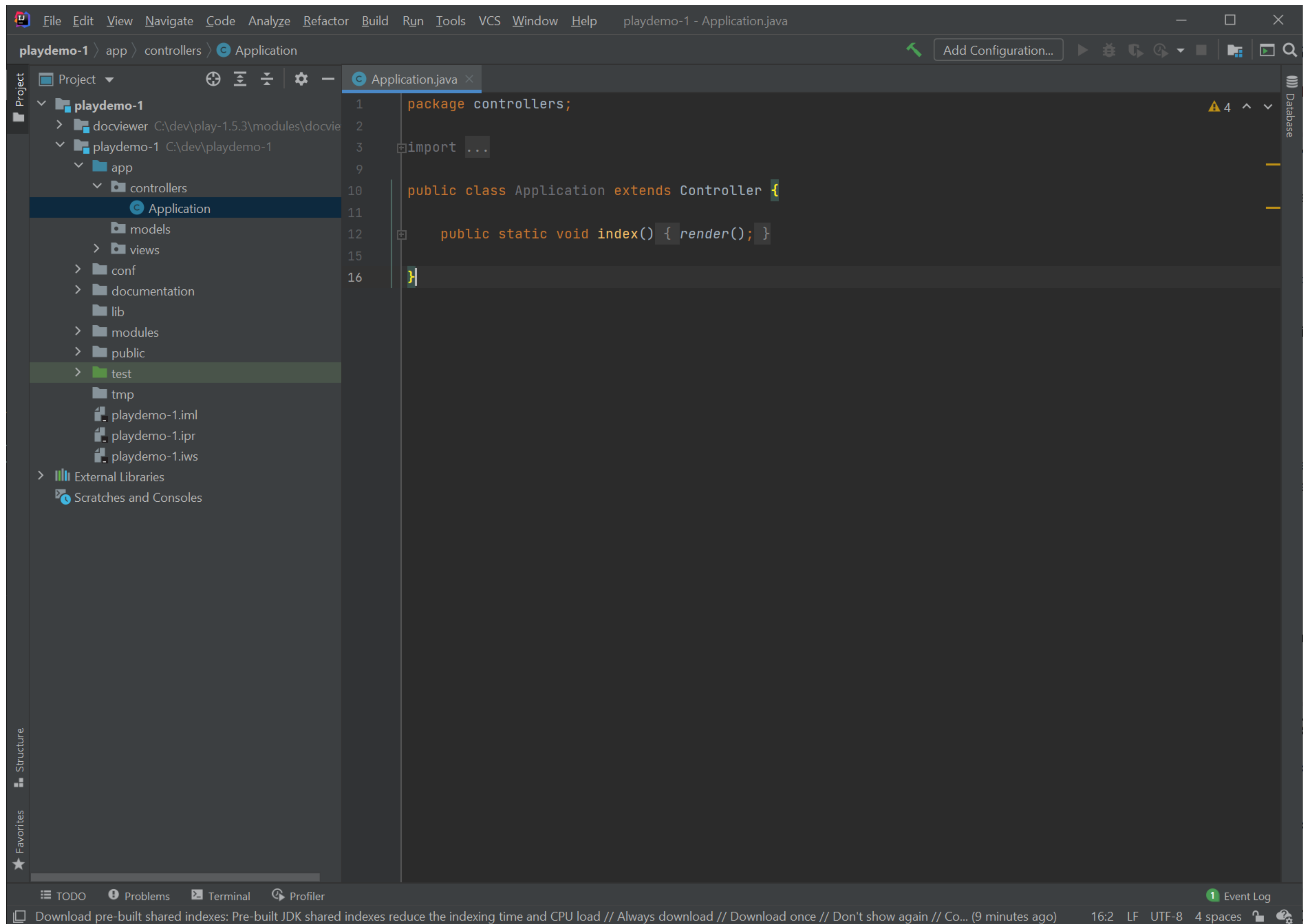
- Enable the project to be opened by Idea

```
play idealize
```

The system will respond with:

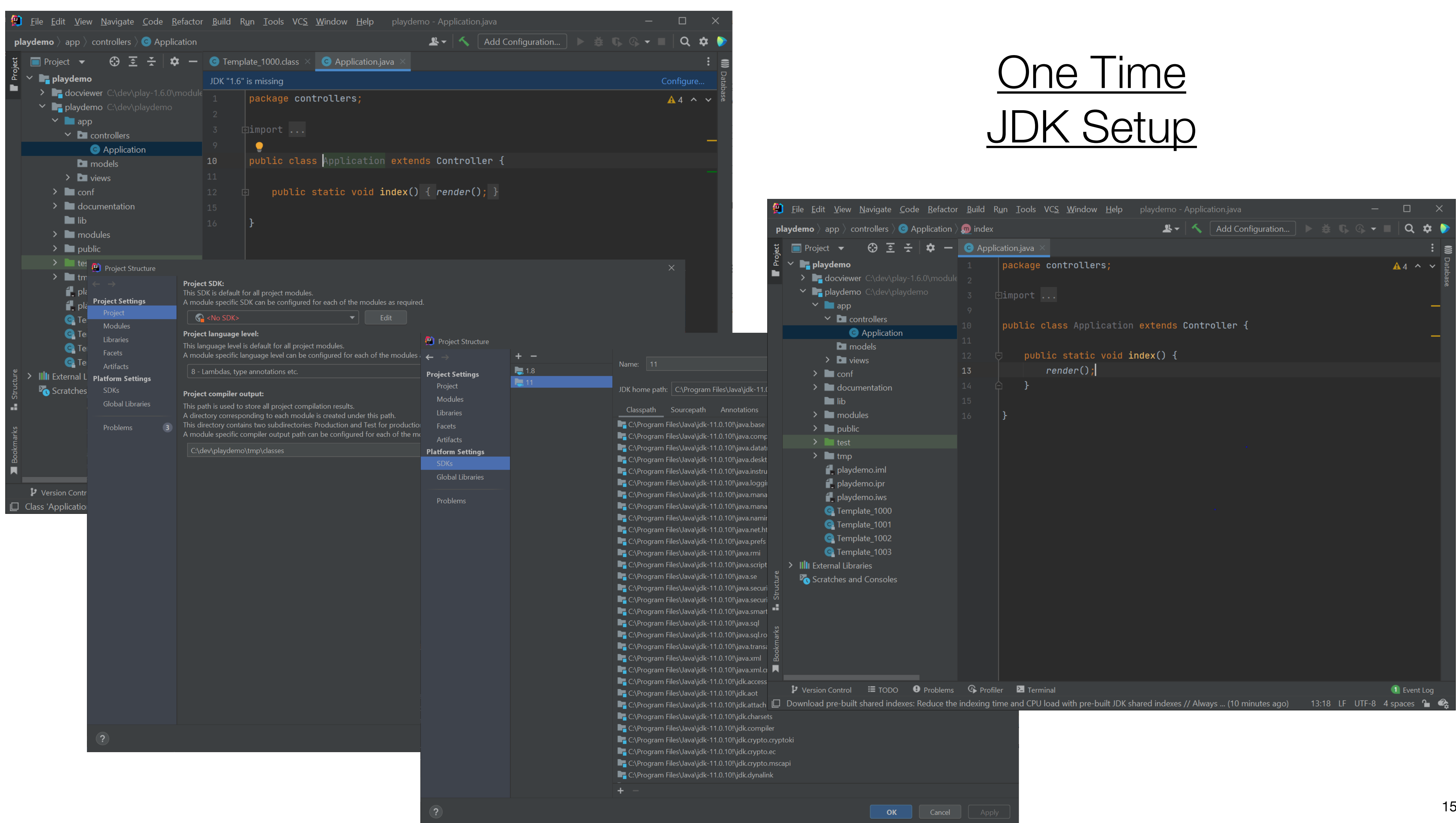
```
~      _      _  
~  _  _  | |  _  _  _  _  | |  
~  |  ' _ \ | | / _ ' | | | _ |  
~  |  _ _ / | _ | \ _ _ _ | \ _ ( _ )  
~  | _ |           | _ _ /  
~  
~  play! 1.6.0, https://www.playframework.com  
~  
~  OK, the application is ready for IntelliJ Idea  
~  Use File, Open Project... to open "playdemo.ipr"  
~
```

# In Idea





# One Time JDK Setup



# Starting to Play



```
~  
~  
~  _ _ _ _  
~ |'_\|/_|_|_|_|  
~ | _\|/_|_|_|_|  
~ |_|_|_|_|  
~  
~  
~ play! 1.5.0, http://www.playframework.org  
~  
~ Usage: play cmd [app_path] [--options]  
~  
~ with,  new      Create a new application  
~        run      Run the application in the  
~        help      Show play help
```

download · expand · path ·  
verify · run · idealize ·  
skeleton app