

Ammonite: Succinct Scala Shell Scripting

Markus Dale, medale@asymmetrik.com

October 2019

- Slides: <https://github.com/medale/prez-ammonite-scala-shell/blob/master/presentation/AmmoniteShell.pdf>
- Code Examples:
<https://github.com/medale/prez-ammonite-scala-shell/code>

Examples of Shell Scripting

- Automate a small but labor-intensive task
- Run clean-up jobs with cron
- Refresh all git repositories (git pull)
- build and publish all projects under a common dir

That Bash Syntax

```
# 3041 upvotes for question, 4259 for answer
if [ ! -f CommonImports.sc ]; then
    echo "File not found!"
fi
# elif?
```

```
find . -name *.foo -exec rm -f {} \;
```

Scala: file exists?

```
import java.nio.file._

val f = Paths.get("CommonImports.sc")
if (!Files.exists(f)) {
  println("File not found!")
}
```

Scala: find and delete

```
import java.nio.file._
import scala.collection.JavaConverters._

val cwd = Paths.get(sys.props("user.dir"))
val filesStream = Files.walk(cwd)
val files = filesStream.iterator.asScala.toList
val fooFiles = files.filter(_.toString.endsWith(".foo"))
fooFiles.foreach(Files.delete(_))

filesStream.close
```

```
if (!exists("CommonImports.sc")) {  
    println("File not found!")  
}
```

```
//also: cd!, ls!, rm!, mv!...
```

Installing Ammonite

- Prerequisite: Java 8 JDK in your path, e.g. from Azul Zulu Community Java 8
- Download 1.7.4 (2.12 or 2.13) binary from <https://github.com/lihaoyi/Ammonite/releases>
- Add as amm to your path
 - 1.6.7. last 2.11 version (if you are running older version of Spark...)
 - Can install multiple versions locally

```
cd ~/bin
```

```
ln -s amm212 amm
```

```
ls -l
```

```
amm -> amm212
```

```
amm211
```

```
amm212
```


Other ways of installing Ammonite

- From ammonite.io via curl

```
sudo sh -c '(echo "#!/usr/bin/env sh" && curl -L \  
↪ https://github.com/lihaoyi/Ammonite/releases/download/1.7.4/2.13-1.7.4) > \  
/usr/local/bin/amm && chmod +x /usr/local/bin/amm' && amm
```

- On Mac: `brew install ammonite-repl`
- Windows:

<https://github.com/lihaoyi/Ammonite/issues/119>

Configure the shell ~/.ammonite/predef.sc

- Ammonite source: <https://github.com/lihaoyi/Ammonite>
- shell/src/main/resources/ammonite/shell/example-predef.sc

```
interp.load.ivy(  
  "com.lihaoyi" %  
  s"ammonite-shell_${scala.util.Properties.versionNumberString}" %  
  ammonite.Constants.version  
)  
@  
val shellSession = ammonite.shell.ShellSession()  
import shellSession._  
import ammonite.ops._  
import ammonite.shell._  
ammonite.shell.Configure(interp, repl, wd)
```

Running Ammonite with extra memory

amm

`JAVA_OPTS=-Xmx10g; amm`

Major improvements over Scala REPL

- input and output code syntax highlighting
- output valid Scala code
- multi-line editing - move one statement at a time
- statement history search (by word/prefix + up arrow)
- Classpath search

Magic import \$file: Importing scripts

- vals, function defs, classes, objects or traits (not import statements)

```
import $file.CommonImports //in current dir
```

```
import $file.scripts.Utills //in subdir scripts
```

```
import $file.^ .print //in parent dir
```

Magic import \$exec: Bring in the defs AND imports

```
//contains: import java.nio.file._
```

```
import $exec.CommonImports
```

```
Paths.get("foo")
```

```
import CommonImports._
```

```
printlnWithNewlines(List(1,2,3))
```

Magic import \$ivy: Download libraries and their dependencies

```
import $ivy.`com.univocity:<TAB>`  
`com.univocity:univocity-api  
`com.univocity:univocity-parsers  
`com.univocity:univocity-common-api  
`com.univocity:univocity-common-parser-api
```

```
import $ivy.`com.univocity:univocity-parsers:<TAB>`  
`com.univocity:univocity-parsers:1.0.0  
`com.univocity:univocity-parsers:1.5.2  
`com.univocity:univocity-parsers:2.3.1
```

```
import $ivy.`com.univocity:univocity-parsers:2.8.3`
```

- `repl.imports` - user imports only
- `repl.fullImports` - also shows Ammonite imports
- `repl.clipboard.read/write`

- `interp.load.cp('lib/"baz.jar")` - load a jar from file system

Paths - Path vs. RelPath

```
Path("/tmp")
```

```
res27: Path = root/'tmp
```

```
RelPath("lib")
```

```
res28: RelPath = 'lib
```

```
RelPath("/lib")
```

```
IllegalArgumentException: /lib is not an relative path
```

```
Path("temp")
```

```
IllegalArgumentException: temp is not an absolute path
```

```
ls! "lib"
```

```
desugar(ls! "lib")
```

Special path variables

- home - `$HOME` or `user.home`
- root - `ls!` `root == Bash: ls /`
- pwd - `sys.props("user.dir")` //does not change
- wd - changes with `cd!` command
- up - `..` in path, e.g. `root/'tmp/up == root`

- `ls! == ls! wd == ls(wd)`
- `rm! dir - Bash: rm -R dir`
- `cp(src,dest)`
- `mv(src,dest)`
- `mkdir! newDirPath - Bash: mkdir -p newDir`
- `stat! filePath`

```
val imports = read("CommonImports.sc")
val lines = read.lines("CommonImports.sc")
val in = read.getInputStream("CommonImports.sc")

import $ivy.`com.typesafe.akka::akka-http:10.1.9`
val app = read(resource / "reference.conf")
```

- Print output to console: %
 - `%git 'status'`
 - `%git("status", "--help")`
- Output to `CommandResult` object: %%
 - `cmdRes.exitCode, .out, .err`

Pipes and grep!

- | - map
- || - flatMap
- |? - filter
- |& - reduce
- |! - foreach
- ls! wd |? grep! "[Ff]oo".r

- upickle - JSON and serialization/deserialization
- requests-scala - HTTP/S REST calls

- WikipediaVocab.sc
- Spreadsheet.sc

And now for something completely different: Colon Cancer



- Screening saves lives!
 - Colonoscopy - talk to your doc
 - Dave Barry: A journey into my colon — and yours
- Colorectal Cancer Alliance
- Ora Lee Smith Cancer Research Foundation

Questions?



- medale@asymmetrik.com
- <https://github.com/medale/prez-ammonite-scala-shell/blob/master/presentation/AmmoniteShell.pdf>

- Ammonite Shell
- Vector graphics for this presentation