

Ammonite “Quirks”

While Ammonite is great when it comes to additional features, such as multiple main’s and dynamically loading files, and in the past it was a rock-solid piece of software written by one of my Scala-heroes (Haoyi Li), it seems it has acquired some “quirks” when it transitioned to Scala 3. Interestingly, the problems listed below only occur inside the REPL. So running Scala-scripts on the command line with for example

```
$ amm cw02.sc
```

works fine. Similarly, the original Scala REPL seems to not suffer from these quirks. Unfortunately, this REPL does not allow you to dynamically load files which might be relevant in later CWs.

Problem with Implicits

In CW2 (and possibly others), if you use the template code and develop your code inside the REPL, a problem arises in the line

```
val KEYWORD : Rexp = "while" | "if" | "then"
```

with the error

```
-- [E008] Not Found Error: cmd17.sc:1:36 -----
1 | val KEYWORD : Rexp = "while" | "if" | "then"
  | ~~~~~
  | value | is not a member of ammonite.$sess.cmd4.ALT.
  | An extension method was tried, but could not be fully constructed:
  |
  |   ammonite.$sess.cmd16.|()
  |
  |   failed with:
  |
  |     value |: <overloaded ammonite.$sess.cmd16.|> does not take
  |     parameters
  | Compilation Failed
```

Fix: The code runs absolutely fine when called on the commandline. It also runs absolutely fine inside the standard Scala REPL. If you prefer to use to use the Ammonite REPL, then you need to make the following changes: (1) comment out the implicits below and change the \$ to for example &

```
extension (s: String) {  
  def & (r: Rexp) = RECD(s, r)  
  //def | (r: Rexp) = ALT(s, r)  
  //def | (r: String) = ALT(s, r)  
  //def % = STAR(s)  
  //def ~ (r: Rexp) = SEQ(s, r)  
  //def ~ (r: String) = SEQ(s, r)  
}
```

You need to then use & in the code for the WHILE_REGS