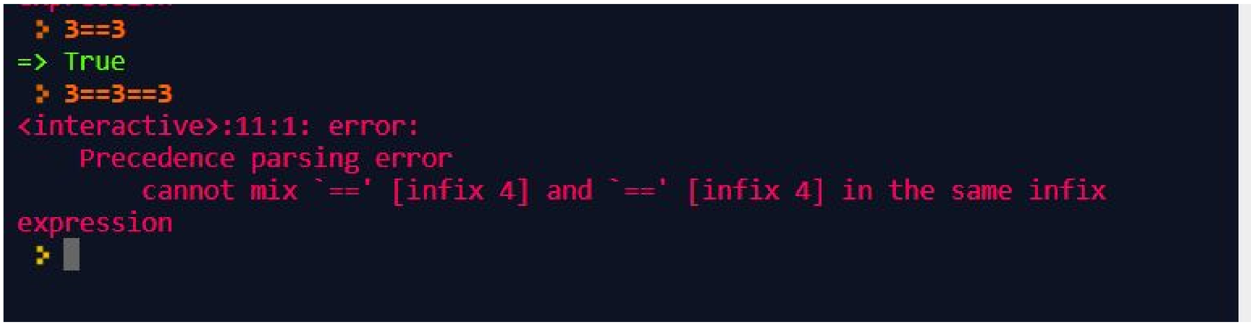
1.3
A.

If such operators appear multiple times, the compiler will show precedence praising error. It can't decide over the precedence of those operators.

For example, these are the basic instructions I ran in https://repl.it/languages/haskell.



```
 › 3==3
=> True
 › 3==3==3
<interactive>:11:1: error:
    Precedence parsing error
        cannot mix `==' [infix 4] and `==' [infix 4] in the same infix
expression
 › 
```

As stated in the Fixity Declaration in Haskell documentation, == has a priority of infix 4(Here 'infix"(with no r or l in end) means it has no associativity. Here, == appears twice and the compiler is confused about which operator to execute first.

B.

$ operator has a fixity of 0(lowest of all infixr) ie it operated at the end and is right-associative.

The expression (^) 2 $ (*) 5 $ (+) 2 3 can be re-written in infix notation as

2^(5*(2+3))

Reference: Reference: https://www.haskell.org/onlinereport/decls.html#prelude-fixities