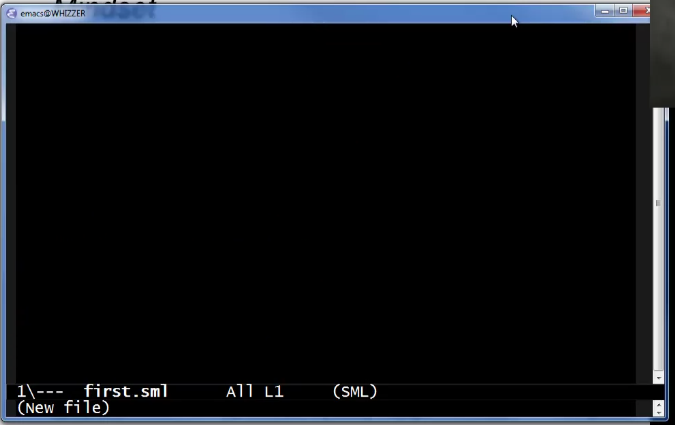
Create your new sml file

1. Open Emacs
2. C-x C-f and type in the path you want your new file to be at
3. Hit enter



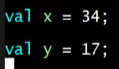
Or you can create your file in advance if you want to

Writing comments



(\* <comment here> \*)

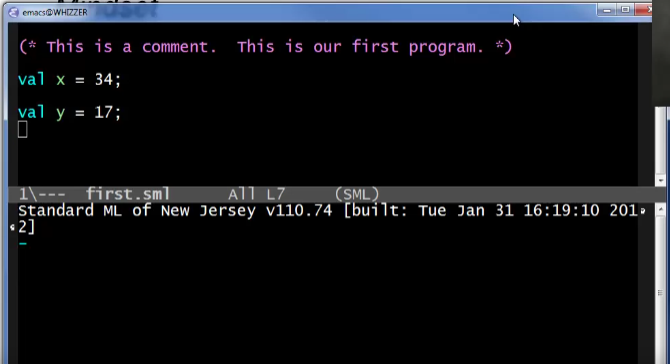
Creating a new variable (val Keyword)



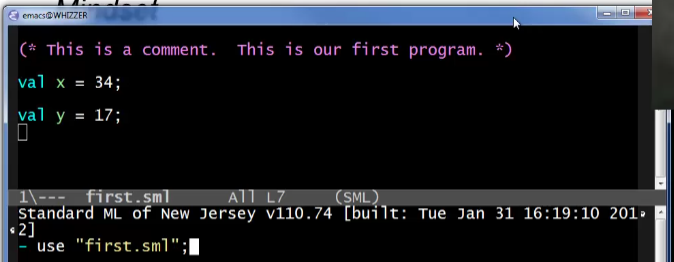
Saving: C-x C-s

Using REPL (Read Eval Print Loop) to run your program

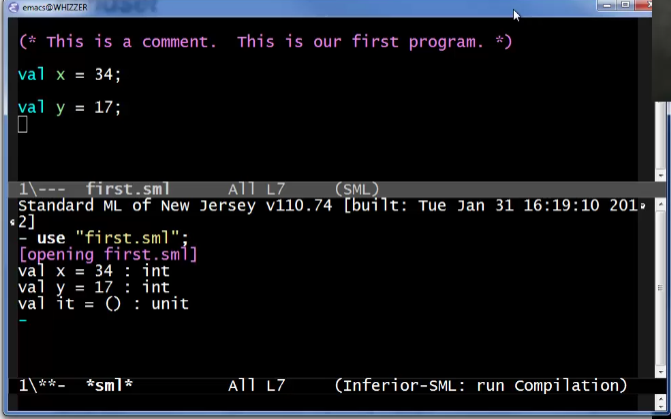
1. Hit C-c C-s and Enter to have a sml language terminal



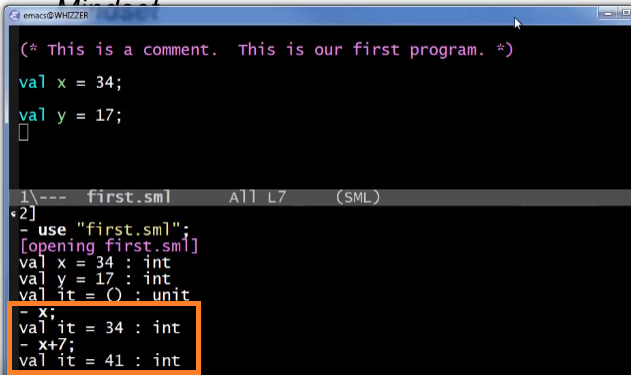
1. Use your sml file



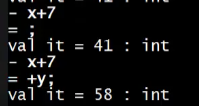
1. Hit Enter



You can also use this prompt to see/evaluate things without having to edit your sml file

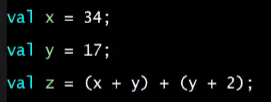


Multiple line splitting



End the line with “;’

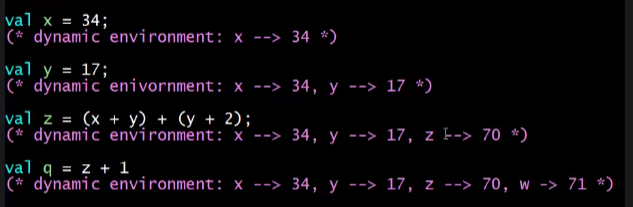
Variable bindings



val z can use x and y

* Variables can ONLY be used when they are declared first (above your line of statement)

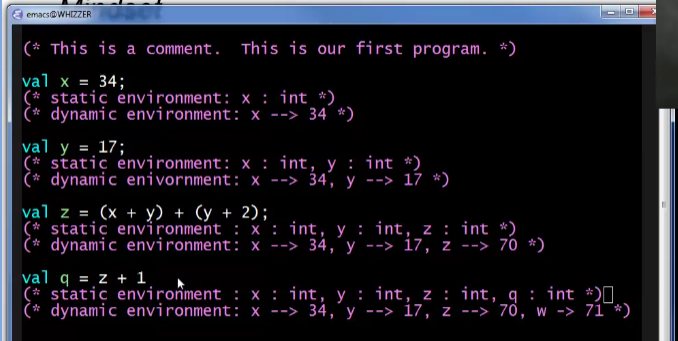
What is happening per line in our dynamic environment? (How expressions are evaluated)



* Note that the variables are stored in the dynamic environment after you declare it

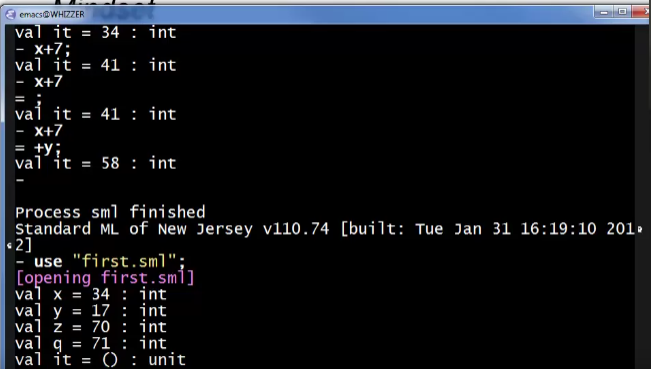
But before all of this, our entire program is TYPE-CHECKED

* Check consistency of types within the expressions
* This is in the static environment (note the comments are per line of expression but it is really before the execution of all our code)



Save and try running again

1. C-x C-s for saving the file
2. C-c C-s open sml terminal
3. Use sml file

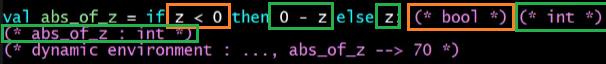


This looks like it was running it altogether but in reality, type checks are done first before all of the expressions!

Conditional Expressions (with evaluation comments)



Type checking:



Calling functions



Note: there are no ()

Additional Slides (Summary)

