

EECS-111 Advanced Section Exercise 1:

Extending the simple interpreter

Out: Friday, October 23

Due: Saturday, October 31, **noon**

In this assignment you will extend the interpreter written in class to handle special forms.

Getting started

Start by opening the file `Simple interpreter.rkt`. This code is slightly different from the version shown in class. Since you'll be adding special forms, the interpreter can no longer assume that expressions that are lists are automatically procedure calls. So we've changed `eval` so that when the expression is a list, it calls `eval-complex-expression`. The current version of `eval-complex-expression` just calls `eval-procedure call`. You'll change that.

Part 1

Change `eval-complex-expression` so that it can also handle **if** expressions. That is, when the expression is an if expression, it should execute it correctly, but otherwise it should treat it as a procedure call. Note that to test this, you will want to add some predicates to the **default-environment** list, since it currently only has arithmetic procedures.

Part 2

Now change it so that it will also handle **cond** expressions. If you're not clear on how `cond` works, you can read about it [here](#).

Part 3

Now change it so that it can also handle **local** expressions. Don't worry about handling the case where you have several defines in the local expression that use one another's values. That is, you don't have to handle:

```
(eval '(+ 1
          (local [(define n-squared (* n n))
                  (define tns (* n-squared 2))]
                tns))
      n)
(program '((define n 7))))
```

But your program should be able to handle something like:

```
(eval '(+ 1
```

```
(local [(define n-squared (* n n))]  
  (* n-squared 2))  
n)  
(program '((define n 7))))
```

Or:

```
(eval '(+ 1  
  (local [(define n-squared (* n n))]  
    (local [(define tns (* n-squared 2))]  
      tns))  
  n)  
(program '((define n 7))))
```

Hint: the **program** procedure we wrote in class does most of what you need. It's not exactly what you need, but you can start by copying the code for it. A good name for the new version might be **make-environment-from-definitions**.