Recall:

lambda is used to

* define functions
* define recursion functions (via Y combinator)
* define pairs (like C structures) (via Church encoding)
* represent integers (via Church encoding)
* go-to

as shown in lect14.scm

* assignment

Continuation Passing Style (CPS)

Only ‘tail calls’ are allowed, no regular calls. This removes the need for a stack, implementing a self-contained stack. An example of this is shown in lect15.scm ‘cfact’. CPS shows explicitly the order in which functions are evaluated.

As a consequence of ‘all calls are tail calls’, arguments are either

* constants
* variables
* lambda expressions

CPS has several important uses, one of which is in *Web Computing*.

CPS allows programming to done remotely, e.g. when filling out an online form, rather than having to request info across a network.

In lect15.scm ‘hypot’ is another example of CSP in action. Mathematically we want z= sqrt ( x2 + y2 ) , given ( x, y).

Using CSP we need to calculate, in order:

1. x2 , y2
2. x2 + y2
3. sqrt (x2 + y2 )

This is explicitly shown in the layout of ‘hypot’.