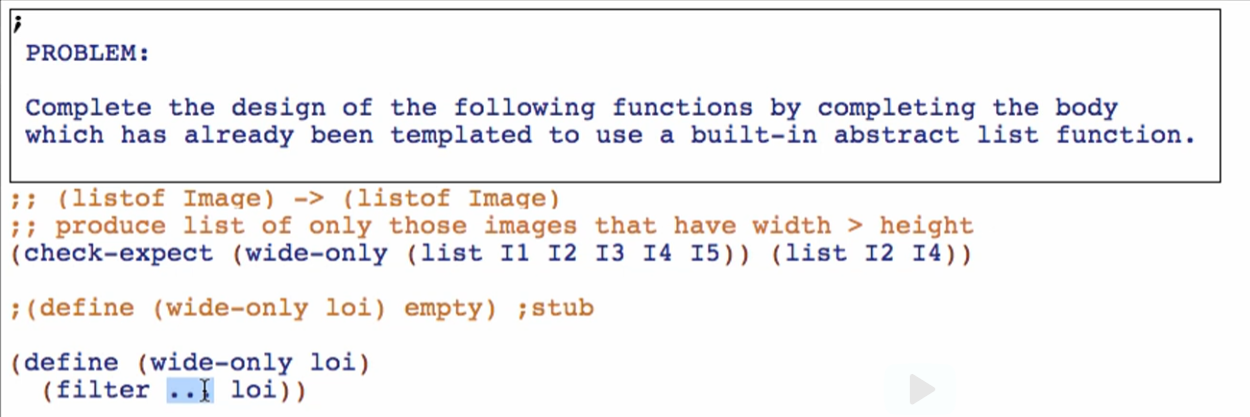
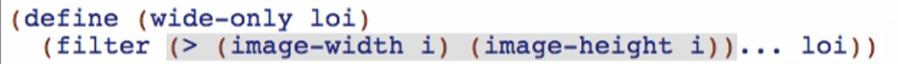
When the function passed to an abstract function requires access to a parameter of the enclosing function, it must be locally defined.

**Let’s try the wide-only function**



* This time, we don’t have the function we need to put in the dots
* wide? does not exist yet

We try to think what we like to put there:



But we can’t put it there because filter only accepts FUNCTION as its first argument

So we can define that expression as a function definition LOCALLY



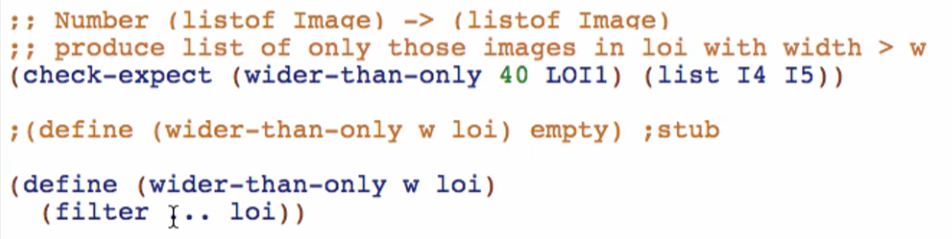
Replace the … with the defined function



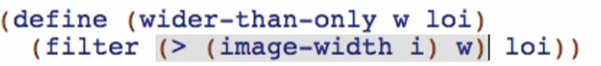
Run and check if passing!

Note: you can design wide? also in the top-level if you’re more comfortable that way

**Let’s try wider-than-only Function**

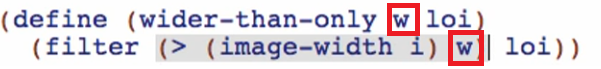
****

Try formulating what condition your filter must have

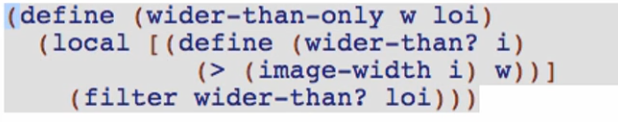


But filter again must accept a function as its first argument

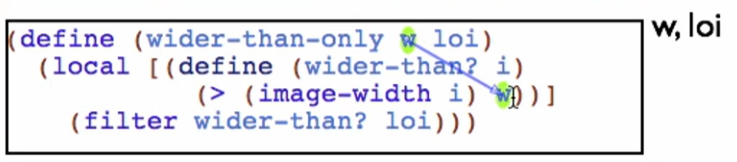
Then notice parameter “w” is passed here in our condition wish for filter



If a parameter is passed in your condition wish, you can ONLY define the function inside a LOCAL. You cannot and must not define it in a global state

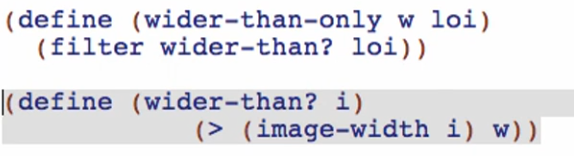


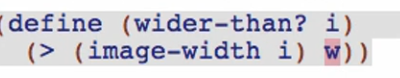
Checking the syntax:



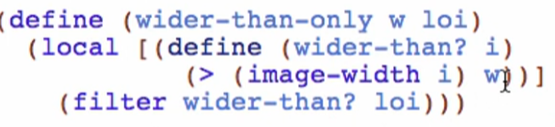
We can refer w and loi here because of lexical scoping

Suppose we try to define the wider-than function globally:





We cannot access the “w” argument of wider-than-only Function here!



In this case, wider-than? Is called a CLOSURE

* “closes over” the value of w passed to wider-than-only function

Overview

* We have 2 cases for defining a non-existing function for our built-in functions
  + Non-closure case
    - Can be defined global or local
    - Doesn’t use a parameter of the main function
  + Closure case
    - Can be defined LOCALLY only
    - Uses a parameter of the main function in its local definition