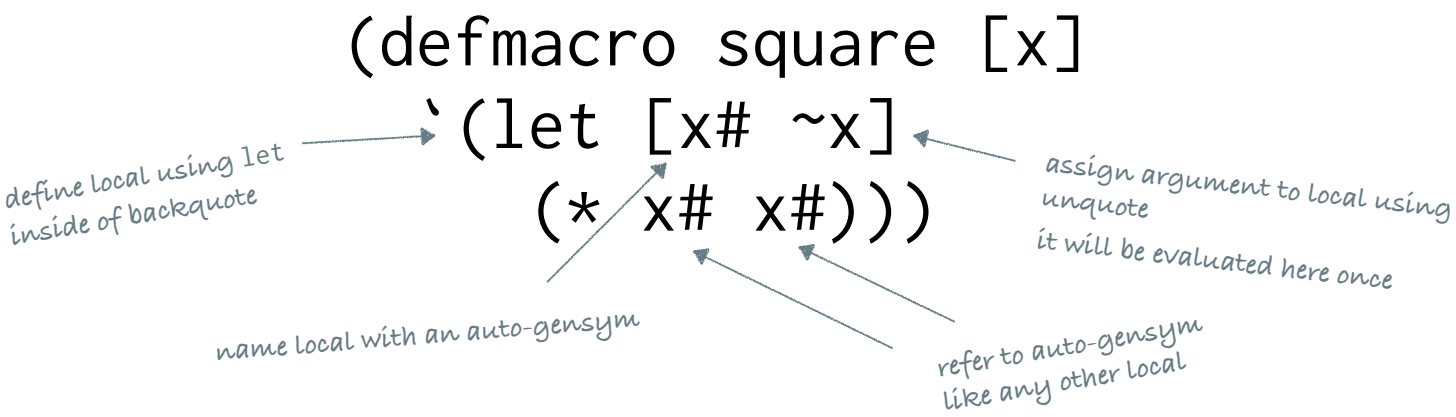
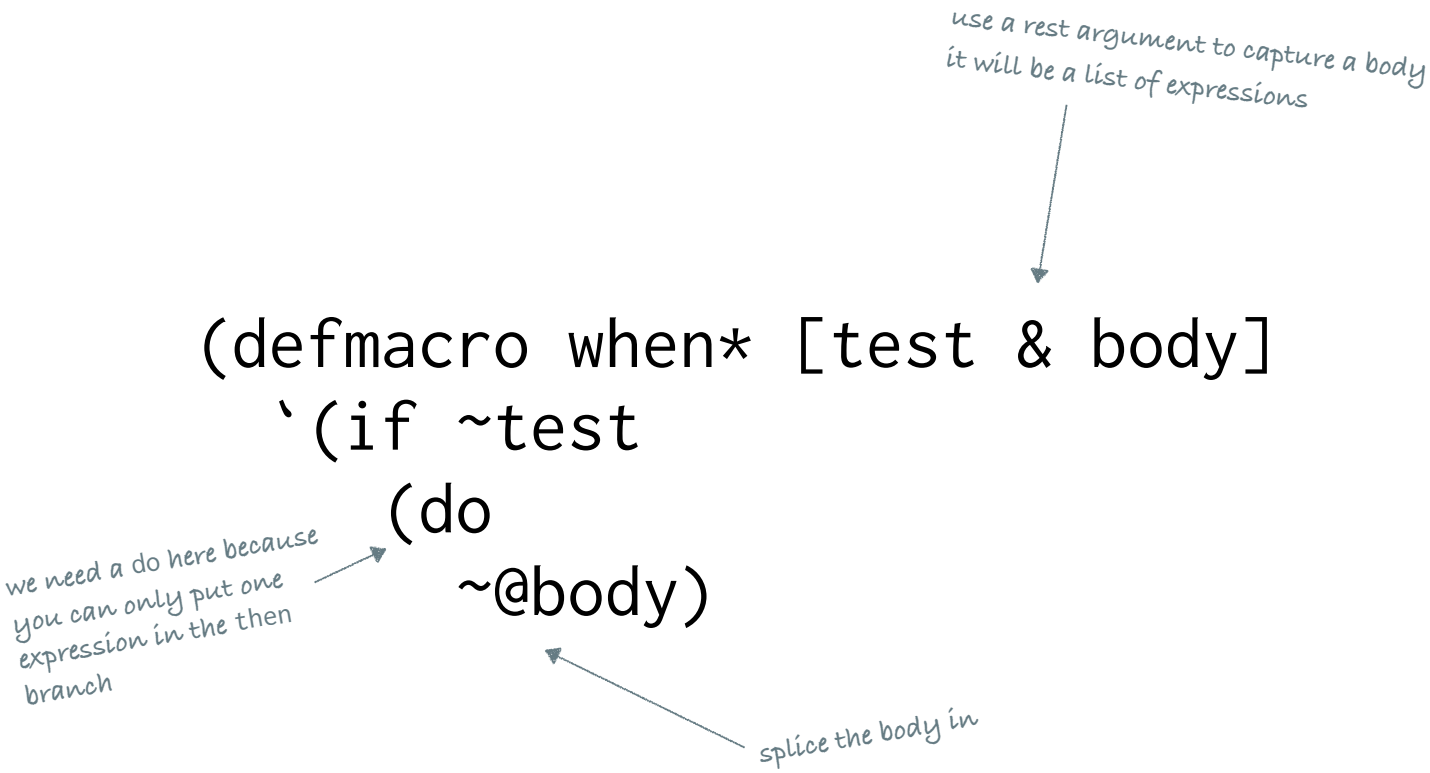
Auto-Gensym Local



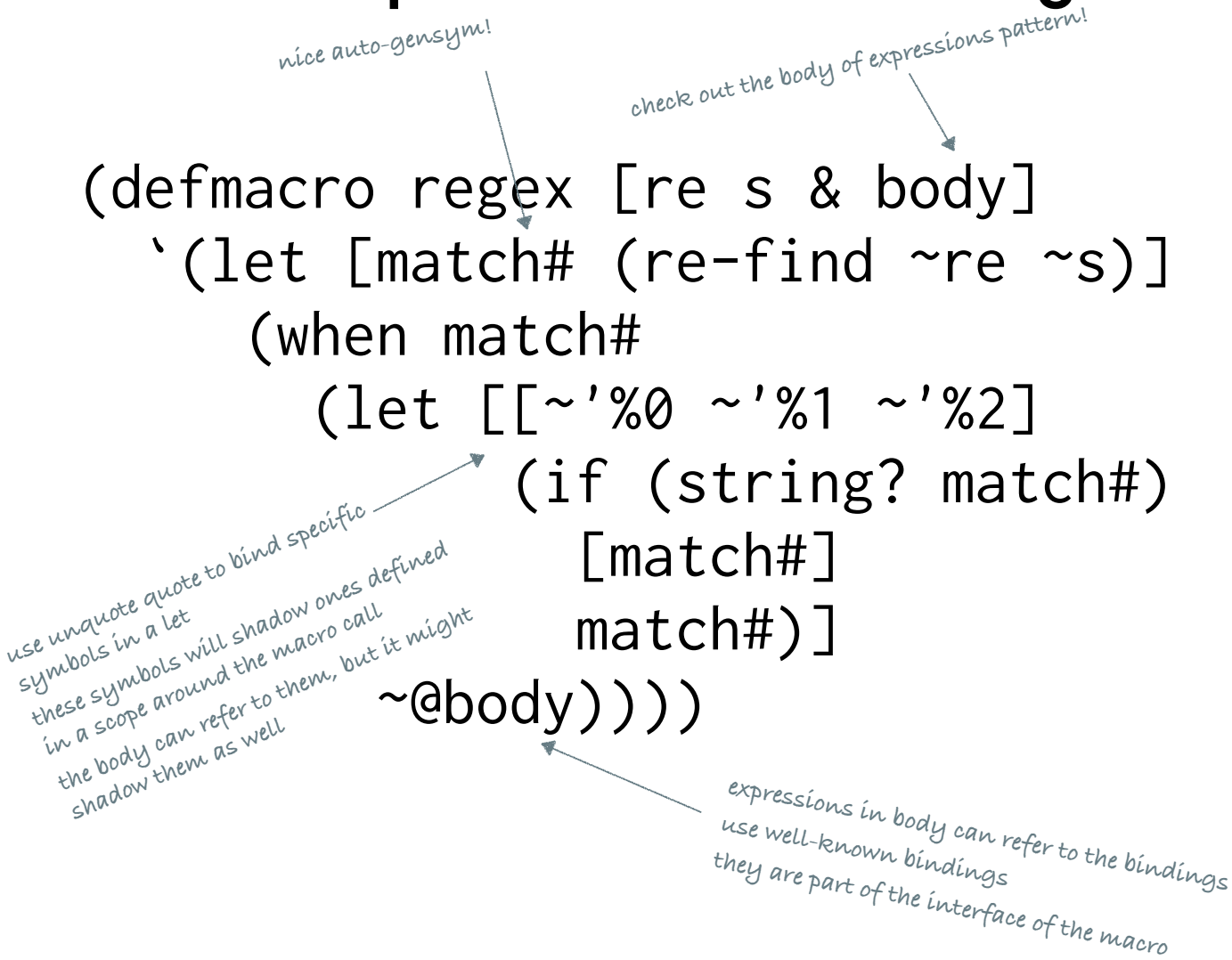
Use auto-gensyms to define locals to prevent evaluating arguments multiple times.

Body of Expressions



Use a rest argument to capture a "body" and splice the body into place. You may need to wrap multiple expressions in a do.

Unquote Quote Binding



Use unquote quote to get an namespace-unqualified symbol and bind it in a let. That way, you can refer to the local in your body.

Macro with Helper Function

define a helper function to do the heavy lifting
as a bonus, this function may be useful to call on its own

helper fn should take a function to call
the function is made in the macro using the body

```
(defn with-open*-fn [to-close f]
  (try
    (apply f to-close)
    (finally
      (doseq [c (reverse to-close)]
        (.close c)))))
```

applying f to these is equivalent to a let since the binding names were turned into arguments

macro will do mostly syntactic transformation

```
(defmacro with-open* [bindings & body]
  (let [pairs (partition 2 bindings)]
    `(with-open*-fn ~(mapv second pairs)
       (fn ~(mapv first pairs)
         ~@body))))
```

call the function in a backquote for it to be called at runtime

create a function using the body

construct the argument vector to correspond to the binding names

in this case we construct a vector of streams to close that we pass to the function
these are also the parameters for the function

Separate a complicated macro into a macro and a helper function. The helper function will be easier to debug.

Binding Macro

define a helper function to do the heavy lifting as a bonus, this function may be useful to call on its own

accept a binding vector just like in a let

```
(defmacro with-open* [bindings & body]
 `(let ~bindings
    (try
     ~@body
     (finally
      ~@(for [[sym _]
              (reverse (partition 2 bindings))]
          `(.close ~sym))))))
```

you can unquote the bindings right in the let

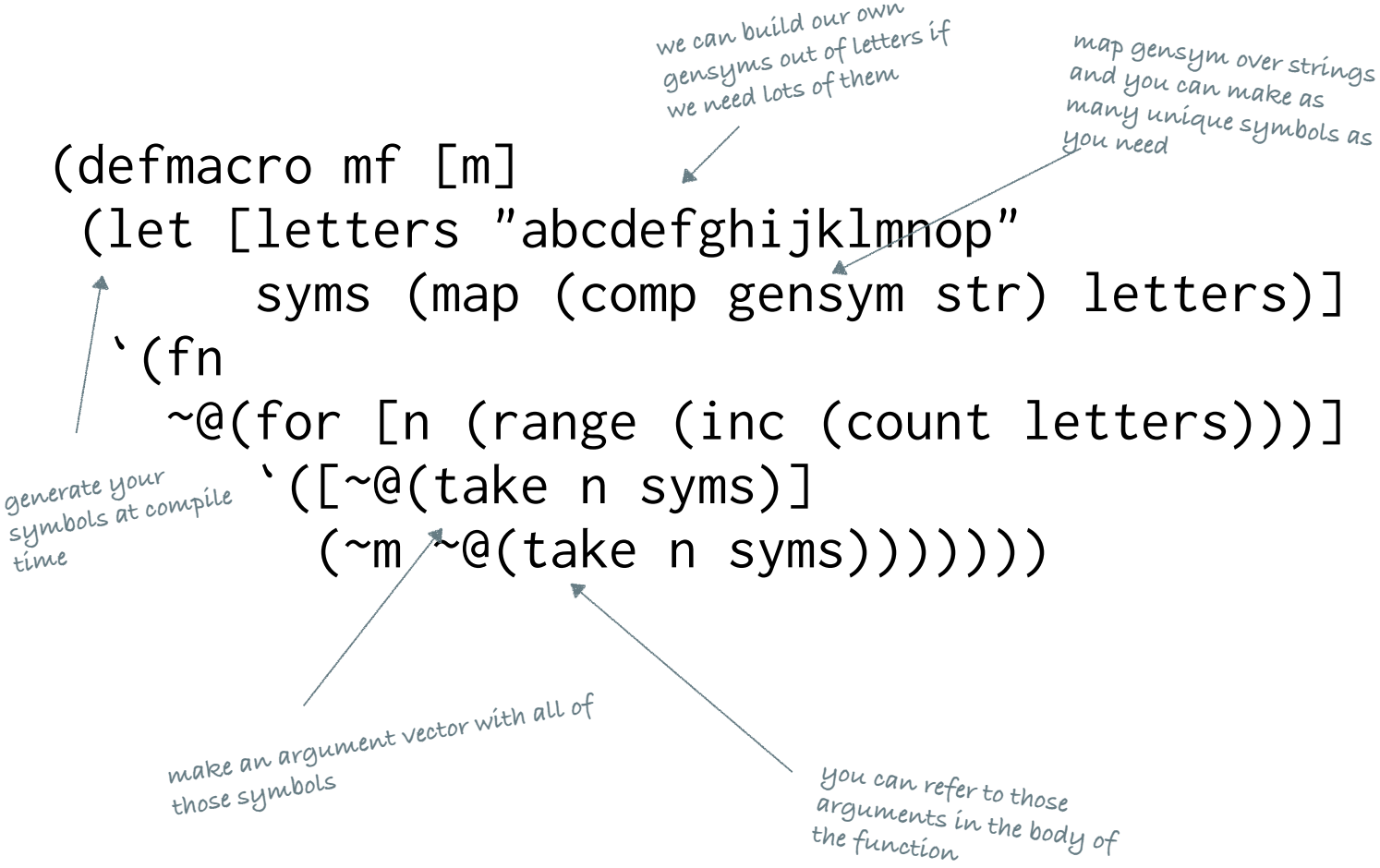
bindings will be available inside the body

unquote splice is great for use with for comprehensions the for will be evaluated at compile time

you can nest another level of backquote here to make sure this evaluates at runtime

Allow macro user to bind locals by using a binding vector, just like let.

Manual Gensyms



Use manual gensyms to create as many symbols as you need. They are useful for long argument lists.