

```

1 local l={}
2 -- Cache names -----
3 local b4={}; for k,_ in pairs(_ENV) do b4[k]=k end
4 function l.rogues()
5   for k,v in pairs(_ENV) do if not b4[k] then print("?",k,type(v)) end end end
6
7 -- Lists -----
8 function l.push(t,x) t[1+#t]=x; return x end
9
10 -- Print table -----
11 function l.cat(t, show,u,pub)
12   if type(t)~="table" then return tostring(t) end
13   function show(k,v)
14     if not tostring(k):find("[A-Z]" then
15       v=l.cat(v)
16       return #t==0 and string.format("%s%s",k,v) or tostring(v) end end
17   u={}; for k,v in pairs(t) do u[1+#u] = show(k,v) end
18   table.sort(u)
19   return (t._is or "").."{"..table.concat(u," " ).."}" end
20
21 -- Update slots in 't' from command line -----
22 function l.cli(t)
23   for slot,v in pairs(t) do
24     v = tostring(v)
25     for n,x in ipairs(arg) do
26       if x=="." (slot:sub(1,1)) or x=="-"..slot then
27         v = v=="false" and "true" or v=="true" and "false" or arg[n+1] end end
28     t[slot] = l.coerce(v) end
29   return t end
30
31 -- Define classes -----
32 function l.obj(name)
33   local function new(k,...)
34     local self = setmetatable({},k)
35     return setmetatable(k.new(self,...) or self,k) end
36   local t={_is = name, __tostring = l.cat}
37   t.__index = t
38   return setmetatable(t,{__call=new}) end
39
40 -- Coerce -----
41 function l.coerce(str)
42   local function coercel(str)
43     if str=="true" then return true end
44     if str=="false" then return false end
45     return str end
46   return tonumber(str) or coercel(str:match("^%s*(-)%s*$") end
47
48 -- Coerce lines from csv file (filtering result through 'fun').
49 function l.csv(filename, fun)
50   l.lines(filename, function(t) fun(l.words(t,"",l.coerce)) end) end
51
52 --- Call 'fun' on all lines from 'filename'.
53 function l.lines(filename, fun)
54   local src = io.input(filename)
55   while true do
56     local str = io.read()
57     if not str then return io.close(src) else fun(str) end end end
58
59 -- Split 'str' on 'sep', filtering parts through 'fun'.
60 function l.words(str,sep,fun, t)
61   fun = fun or function(z) return z end
62   sep = l.string.format("[%s]+",sep)
63   t={};for x in str:gmatch(sep) do t[1+#t]=fun(x) end;return t end
64
65 return l

```