

```

1 local m={}
2
3 m.fmt = string.format
4 function m.push(t,x) table.insert(t,x); return x end
5 function m.map(t,f, u) u={};for k,v in pairs(t) do u[1+#u]=f(v) end; return u end
6 function m.sort(t,f) table.sort(t,f); return t end
7 function m.reject(t,f) return m.select(t, function(x) return not f(x) end) end
8 function m.select(t,f)
9     u={}; for k,v in pairs(t) do if f(v) then push(u,v) end end; return u end
10
11 function m.thing(x)
12     x = x:match("^%s*(-)%s*$"
13     if x=="true" then return true elseif x=="false" then return false end
14     return tonumber(x) or x end
15
16 function m.things(x,sep, t)
17     t={}; for y in x:gmatch(sep or "[^,]+") do t[1+#t]=m.thing(y) end
18     return t end

```

```

19
20 function m.csv(file, x)
21     file = io.input(file)
22     return function()
23         x=io.read(); if x then return m.things(x) else io.close(file) end end end
24
25 function m.slots(t, u)
26     u={}; for k,v in pairs(t) do if k:sub(1,1)~="_" then u[1+#u]=k end end;
27     return m.sort(u) end
28
29 function m.o(t) print(m.o(t)) end
30 function m.o(t)
31     if type(t)~="table" then return tostring(t) end
32     local key=function(k) return m.fmt("%s%s",k,m.o(t[k])) end
33     local u = #t>0 and m.map(t,o) or m.map(m.slots(t),key)
34     return '{'..table.concat(u, " ")..'}' end
35
36 function m.new(k,t) k.__index=k; k.__tostring=o; return setmetatable(t,k) end
37
38 function m.settings(txt, t)
39     d={}
40     txt:gsub("\n [^%s+][^\n]*%s{([^\n]+)",function(key,x)
41         for n,flag in ipairs(arg) do
42             if flag:sub(1,1)=="-" and key:find("^"..flag:sub(2).."*") then
43                 x = x=="false" and true or x=="true" and "false" or arg[n+1] end end
44             d[key] = thing(x) end
45         if d.help then os.exit(print(txt)) end
46         return d end
47
48     return m

```