```
i local m={}

m.fmt = string.format
function m.push(t,x) table.insert(t,x); return x end
function m.map(t,f, u) u={};for k,v in pairs(t) do u[1+#u]=f(v) end; return u end
function m.scrt(t,f) table.sort(t,f); return t end
function m.reject(t,f) return m.select(t, function(x) return not f(x) end) end
function m.select(t,f)

u={}; for k,v in pairs(t) do if f(v) then push(u,v) end end; return u end

function m.thing(x)
x = x:match"%s*{-}%s*$"
if x==""rue" then return true elseif x=="false" then return false end
return tonumber(x) or x end

function m.things(x,sep, t)
function m.things(x,sep, t)
t={}; for y in x:gmatch(sep or"([^]+)") do t[1+#t]=m.thing(y) end
return tend
```

```
function m.csv(file, x)

file = io.input(file)

return function()

x=io.read(); if x then return m.things(x) else io.close(file) end end end

function m.slots(t, u)

u={}; for k,v in pairs(t) do if k:sub(1,1)~="_" then u[1+#u]=k end end;

return m.sort(u) end

function m.o(t)

if type(t)~="uble" then return tostring(t) end

local key=function(k) return m.fmt(":%s %s",k,m.o(t[k])) end

local u = #t>0 and m.map(t,o) or m.map(m.slots(t),key)

return '('.'.bable.concat(u," ")..")" end

function m.new(k,t) k.__index=k; k.__tostring=o; return setmetatable(t,k) end

function m.settings(txt, t)

d={}

tx:tgsub("\n [-]([\(^{\infty}\)]*\(^{\infty}\)]*\(^{\infty}\),", function(key,x)

for n,flag in ipairs(arg) do

if flag:sub(1,1)=="-" and key:find("^*..flag:sub(2)..".*") then

x = x=="false" and true or x=="fune" and "false" or arg[n+1] end end

d[key] = thing(x) end)

return m
```