Lexical Conventions

Lexical Conventions

- Identifiers (or names) in Lua can be any string of letters, digits, and underscores, not beginning with a digit. For instance:

```
I       j       x       i10        _ijx
aLongNameIsHere         _INPUT
```

We should avoid identifiers starting with an underscore followed by one or more upper-case letters (e.g., _VERSION); they are reserved for special uses in Lua.

Usually, the identifier _ (a single underscore) is reserved for anonymous variables.

Lexical Conventions

- The following words are reserved (we cannot use them as identifiers):

`and`	`break`	`do`	`else`	`elseif`
`end`	`false`	`for`	`function`	`goto`
`if`	`in`	`local`	`nil`	`not`
`or`	`repeat`	`return`	`then`	`true`
`until`	`while`			

Lexical Conventions

- Lua is case-sensitive: **`and`** is a reserved word, but `And` and `AND` are two different identifiers.

- A comment starts anywhere with two consecutive hyphens (`--`) and runs until the end of the line. Lua also offers long comments, which start with two hyphens followed by two opening square brackets and run until the first occurrence of two consecutive closing square brackets, like here:

```
--[[A multi-line
    long comment
]]
```

Lexical Conventions

- Lua needs no separator between consecutive statements, but we can use a semicolon if we wish. Line breaks play no role in Lua's syntax. For instance, the following four chunks are all valid and equivalent:

```
a = 1
b = a * 2

a = 1;
b = a * 2;

a = 1; b = a * 2

a = 1  b = a * 2    -- ugly, but valid
```