

Legenda	Comment
New in Moon#	Function has been introduced in Moon# and is not present in Lua 5.2
Done	Implemented completely
Done, with differences	Implemented, but has differences from the Lua version (detailed in comments).
Not done, issues	Not completed yet.
Unsupported, stub	Unsupported, but implemented as a stub for compatibility.
Unsupported	Unsupported and likely support will be limited or non-existent also in the future
Not done yet	Not even started.

API	Yield	CLR	Mono	Unity	Differences	Yielding notes
_G	-	*	*	*		
_VERSION	-	*	*	*		
_MOONSHARP	-	*	*	*	A table, containing several fields including moonsharp version, platform and emulated lua version.	
assert	-	*	*	*		
collectgarbage	-	*	*	*	Some arguments ignored	
dofile	YES	*	*	*		
error	-	*	*	*		
getmetatable	-	*	*	*		
ipairs	YES	*	*	*		
load	NO	*	*	*		A function as input cannot yield.
loadsafe	NO	*	*	*	Same as loadfile, but with better handling of default _ENV for sandboxing.	
loadfile	-	*	*	*	stdin as input is not supported.	
loadfilesafe	-	*	*	*	Same as loadfile, but with better handling of default _ENV for sandboxing.	
next	-	*	*	*		
pairs	YES	*	*	*		
pcall	YES	*	*	*		
print	NO	*	*	*		__tostring cannot yield when called from print
rawequal	-	*	*	*		
rawget	-	*	*	*		
rawlen	-	*	*	*		
rawset	-	*	*	*		
require	YES	*	*	*	Minor internal differences	
select	-	*	*	*		
setmetatable	-	*	*	*		

tonumber	-	*	*	*	Only bases between 2 and 10, and base 16 are supported.	
tostring	YES	*	*	*		
type	-	*	*	*		
xpcall	NO	*	*	*	Double faults are appended to error and ignored.	Code can yield, but the error message handler cannot.
bit32.arshift		*	*	*		
bit32.band		*	*	*		
bit32.bnot		*	*	*		
bit32.bor		*	*	*		
bit32.btest		*	*	*		
bit32.bxor		*	*	*		
bit32.extract		*	*	*		
bit32.lrotate		*	*	*		
bit32.lshift		*	*	*		
bit32.replace		*	*	*		
bit32.rrotate		*	*	*		
bit32.rshift		*	*	*		
coroutine.create		*	*	*		
coroutine.resume		*	*	*		
coroutine.running		*	*	*		
coroutine.status		*	*	*		
coroutine.wrap		*	*	*		
coroutine.yield		*	*	*		
debug.debug		*	*	*		
debug.getuservalue		*	*	*		
debug.gethook					Use the debugger infrastructure.	
debug.getinfo					Function is heavily dependent on implementation details.	
debug.getlocal					Function is heavily dependent on implementation details.	
debug.getmetatable		*	*	*		
debug.getregistry		*	*	*		
debug.getupvalue		*	*	*		
debug.setuservalue		*	*	*		
debug.sethook					Use the debugger infrastructure.	
debug.setlocal					Function is heavily dependent on implementation details.	
debug.setmetatable		*	*	*	Cannot set on userdata as now.	

debug.setupvalue		*	*	*		
debug.traceback		*	*	*		
debug.upvalueid		*	*	*	Works ok, but returns a number instead of userdata (still, can be used in pretty much the same way).	
debug.upvaluejoin						
dynamic.eval		*	*	*	Evaluates the code passed as a parameter dynamically. All accesses are raw,function calls will raise an error.	
dynamic.prepare		*	*	*	Prepares an expression for faster evaluation with dynamic.eval.	
file.close		*	*	NO		
file.flush		*	*	NO		
file.lines		*	*	NO		
file.read		*	*	NO		
file.seek		*	*	NO		
file.setvbuf		*	*	NO		
file.write		*	*	NO		
io.close		*	*	NO		
io.flush		*	*	NO		
io.input		*	*	NO		
io.lines		*	*	NO		
io.open		*	*	NO		
io.output		*	*	NO		
io.popen		*	*	NO	Interprocess comms, stdin and stdout are NOT supported. This however is ok by the Lua standard.	
io.read		*	*	NO		
io.stderr		*	*	NO		
io.stdin		*	*	NO		
io.stdout		*	*	NO		
io.tmpfile		*	*	NO		
io.type		*	*	NO		
io.write		*	*	NO		
math.abs	-	*	*	*		
math.acos	-	*	*	*		
math.asin	-	*	*	*		
math.atan	-	*	*	*		
math.atan2	-	*	*	*		

math.ceil	-	*	*	*		
math.cos	-	*	*	*		
math.cosh	-	*	*	*		
math.deg	-	*	*	*		
math.exp	-	*	*	*		
math.floor	-	*	*	*		
math.fmod	-	*	*	*		
math.frexp	-	*	*	*		
math.huge	-	*	*	*		
math.ldexp	-	*	*	*		
math.log	-	*	*	*		
math.max	-	*	*	*		
math.min	-	*	*	*		
math.modf	-	*	*	*		
math.pi	-	*	*	*		
math.pow	-	*	*	*		
math.rad	-	*	*	*		
math.random	-	*	*	*	Minor differences in accepted inputs (MoonSharp is more tolerant)	
math.randomseed	-	*	*	*		
math.sin	-	*	*	*		
math.sinh	-	*	*	*		
math.sqrt	-	*	*	*		
math.tan	-	*	*	*		
math.tanh	-	*	*	*		
os.clock		*	*	*		
os.date		*	*	*	Running on Mono systems might lead to erroneous output due to a Mono bug (11817)	
os.difftime		*	*	*		
os.execute		*	*	NO		
os.exit		*	*	NO		
os.getenv		*	*	NO		
os.remove		*	*	NO		
os.rename		*	*	NO		
os.setlocale		*	*	NO	Unsupported, currently a stub	
os.time		*	*	*		
os.tmpname		*	*	NO		

package.config		*	*	*		
package.cpath					See "Loaders" in MoonSharp guide to see how to customize loading of scripts and packages	
package.loaded		*	*	*		
package.loadlib					See "Loaders" in MoonSharp guide to see how to customize loading of scripts and packages	
package.path					See "Loaders" in MoonSharp guide to see how to customize loading of scripts and packages	
package.preload					See "Loaders" in MoonSharp guide to see how to customize loading of scripts and packages	
package.searchers					See "Loaders" in MoonSharp guide to see how to customize loading of scripts and packages	
package.searchpath					See "Loaders" in MoonSharp guide to see how to customize loading of scripts and packages	
string.byte		*	*	*	Character codes are cropped to 0-255. Use string.unicode to have the unicode code-point.	
string.char		*	*	*	Values > 255 are silently supported (as unicode codepoints)	
string.dump		*	*	*	Functions with upvalues raise an error as Lua 5.0 did and as Lua 5.2 should do according to documentation.	
string.find		*	*	*	Implementation taken from KopiLua. Patterns cannot contain \0. Use %z instead.	
string.format	NO	*	*	*	Implementation taken from KopiLua.	__tostring metamethod cannot yield.
string.gmatch		*	*	*	Implementation taken from KopiLua. Patterns cannot contain \0. Use %z instead.	
string.gsub	NO	*	*	*	Implementation taken from KopiLua. Patterns cannot contain \0. Use %z instead.	Callback function (if used) cannot yield.
string.len		*	*	*		
string.lower		*	*	*		
string.match		*	*	*	Implementation taken from KopiLua. Patterns cannot contain \0. Use %z instead.	
string.rep		*	*	*		
string.reverse		*	*	*		
string.sub		*	*	*		
string.unicode		*	*	*	Works just as "string.byte" would do, but returns the unicode code-point without truncation.	
string.upper		*	*	*		
table.concat	NO	*	*	*		__len cannot yield
table.insert	NO	*	*	*		__len cannot yield
table.pack		*	*	*		

table.remove	NO	*	*	*		__len cannot yield
table.sort	NO	*	*	*		__lt, __len and the comparison function cannot yield
table.unpack		*	*	*		