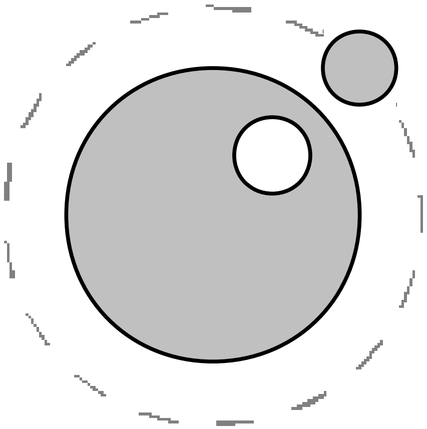
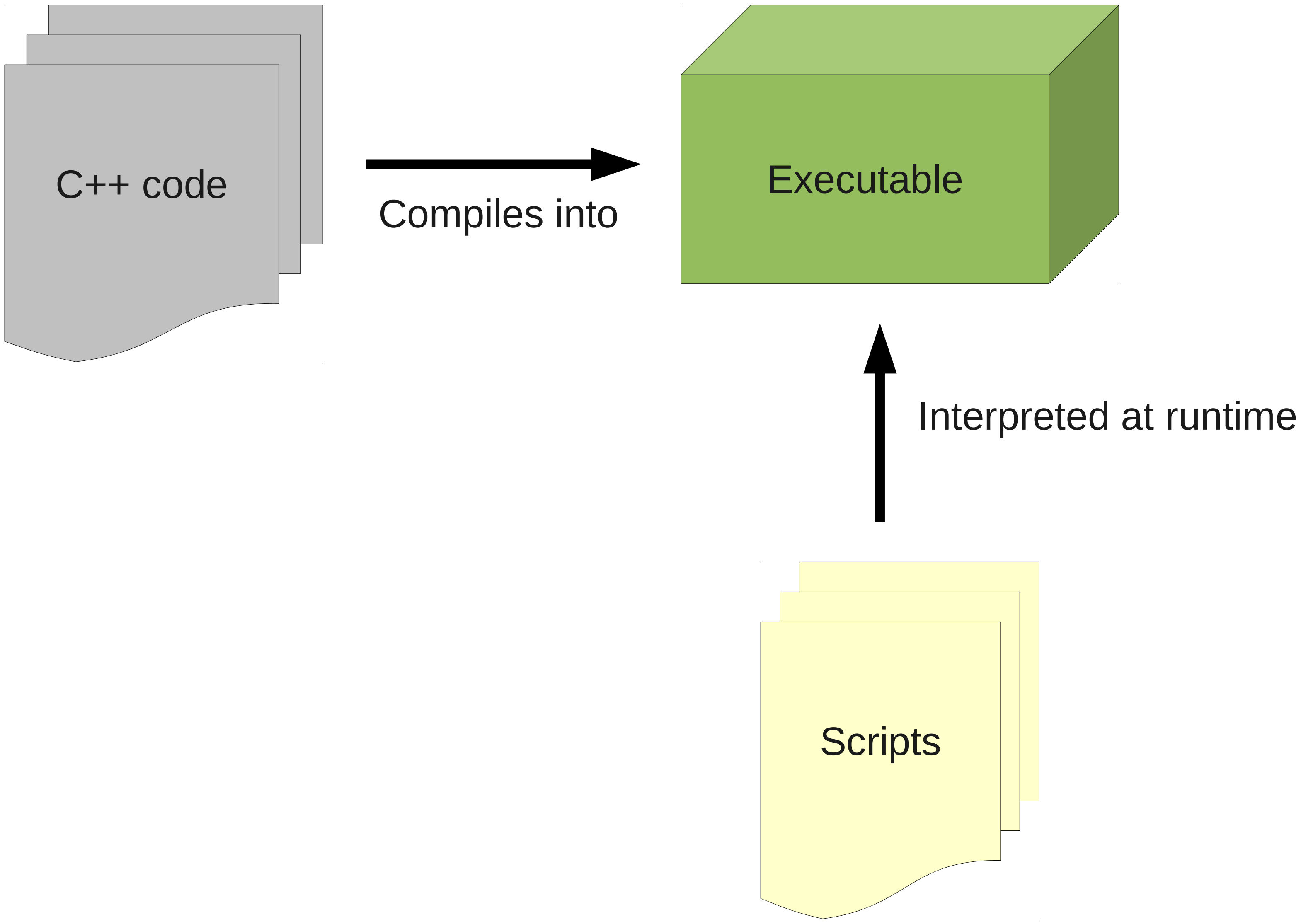
Intro to Scripting in C++



Contents

- What is scripting
- Why use scripting
- How Lua works
- Demo
- Alternatives

What is Scripting?



C++ code

Compiles into

Executable

Interpreted at runtime

Scripts

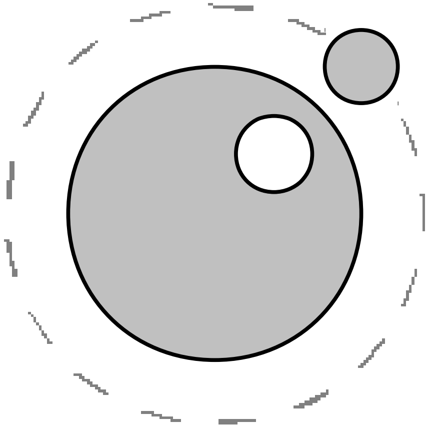
Why use Scripting?

- Separate an engine from its business logic
- Extract complexity, expose simple interface
- Allow users to create plugins

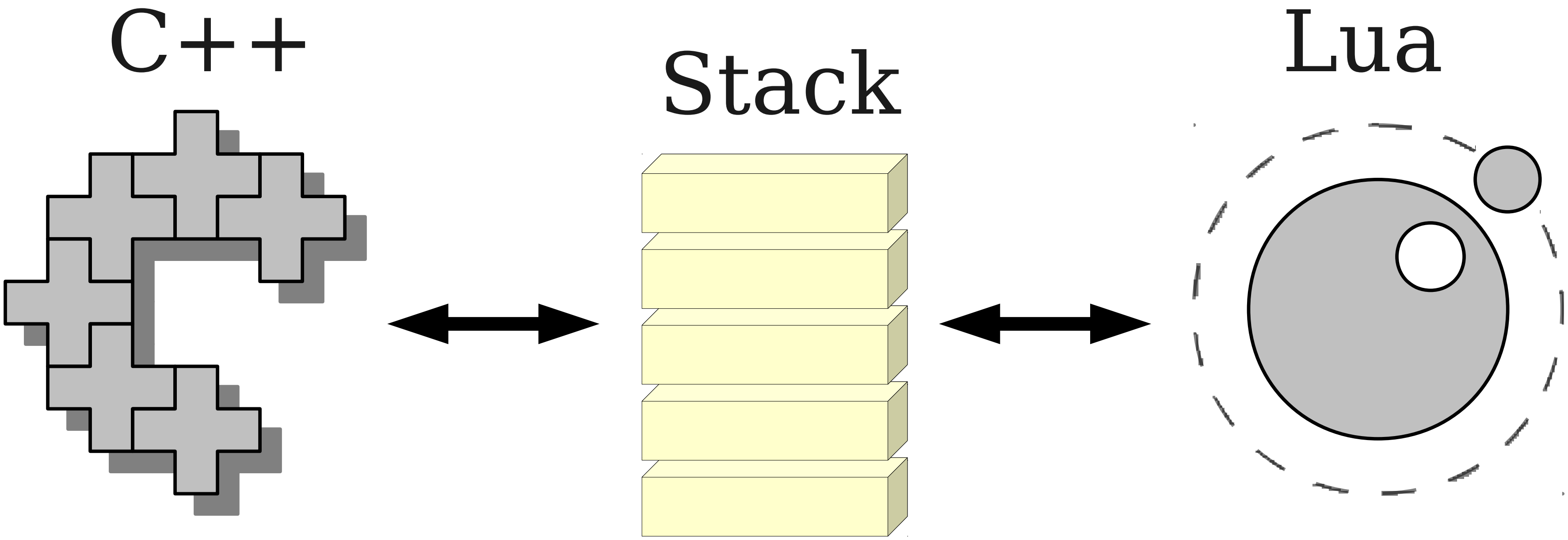
Lua Programming Language

- Our scripting language
- Background

Using the Lua API



The Lua API Stack



C++

Stack

Lua

All communication between C++ and Lua is done through an intermediate stack.

You are responsible for managing this stack.

script.lua

```lua
foo = ”Hello”
bar = 6.28

function add(a,b)
   return a+b
end
```

main.cpp

```
lua_State* L =
   luaL_newstate();

luaL_loadfile(
   L, ”script.lua”);

lua_pcall(L, 0, 0, 0);
```

script.lua

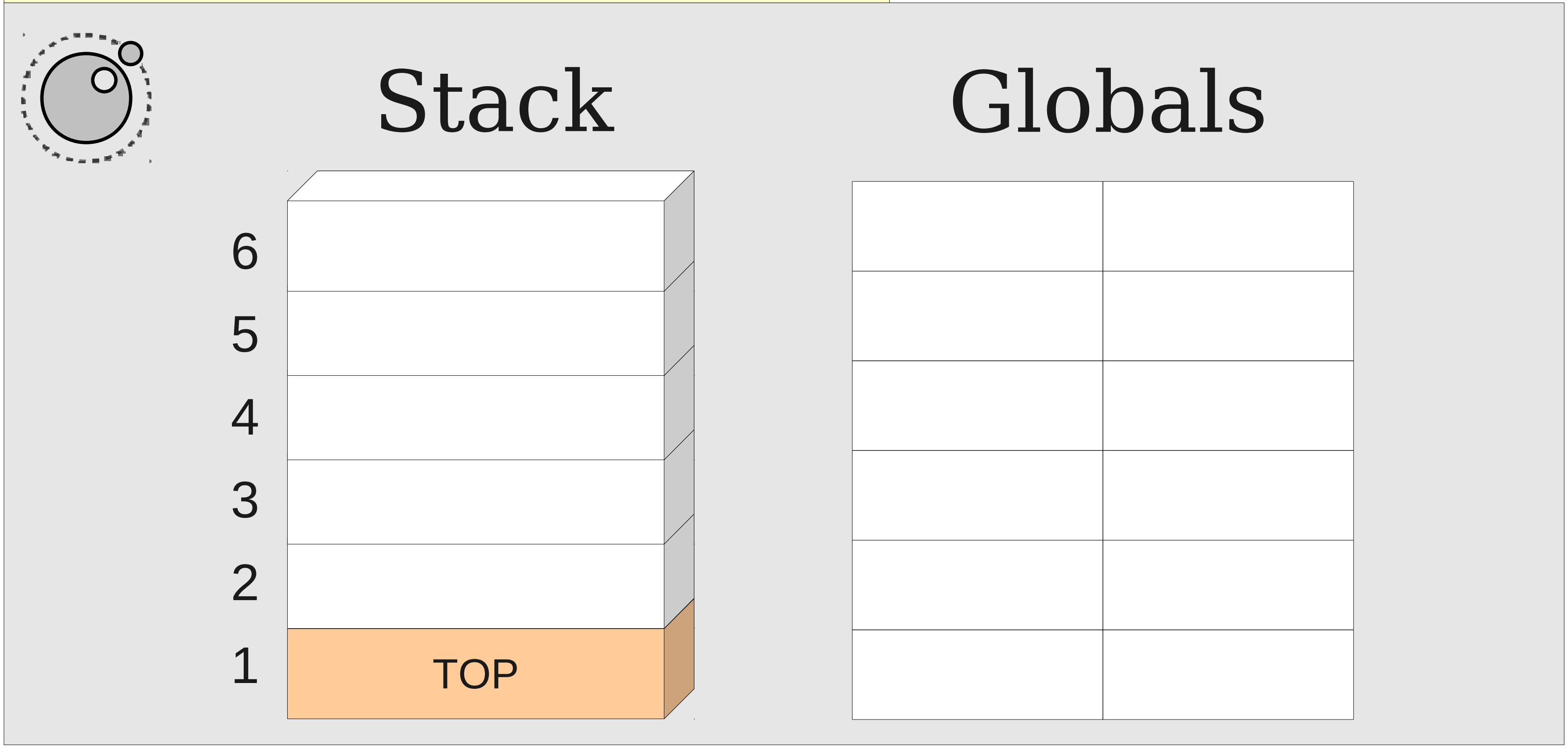
```
foo = ”Hello”
bar = 6.28

function add(a,b)
   return a+b
end
```

main.cpp

```
lua_State* L =
    luaL_newstate();

luaL_loadfile(
    L, "script.lua");

lua_pcall(L, 0, 0, 0);
```

script.lua

```
foo = "Hello"
bar = 6.28

function add(a,b)
    return a+b
end
```

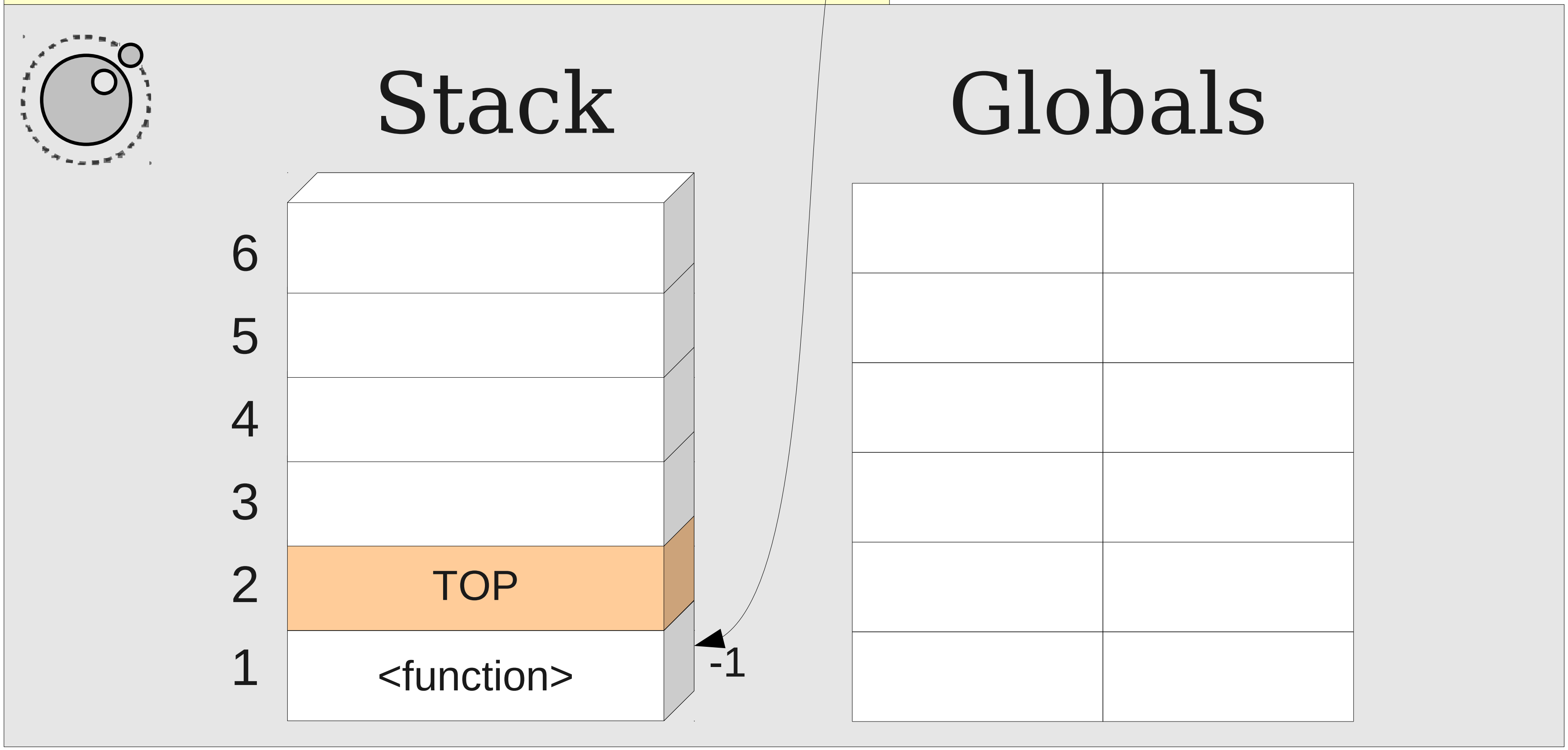


main.cpp

```
lua_State* L =
   luaL_newstate();

luaL_loadfile(
   L, "script.lua");

lua_pcall(L, 0, 0, 0);
```

script.lua

```
foo = "Hello"
bar = 6.28

function add(a,b)
   return a+b
end
```



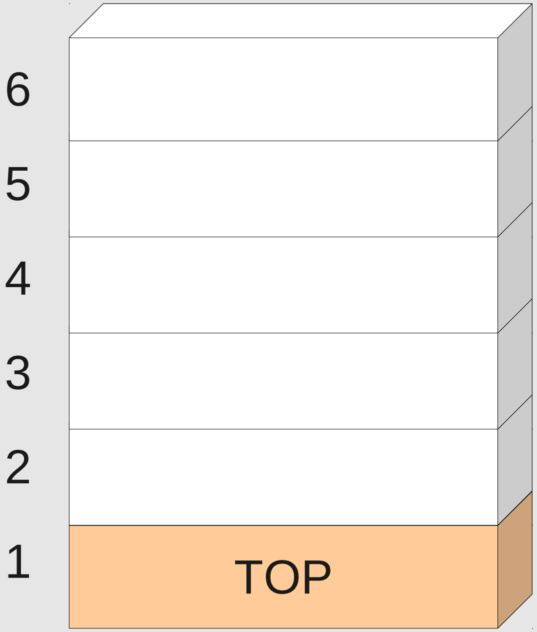
main.cpp

```
lua_State* L =
   luaL_newstate();

luaL_loadfile(
   L, "script.lua");

lua_pcall(L, 0, 0, 0);
```

script.lua

```lua
foo = "Hello"
bar = 6.28

function add(a,b)
   return a+b
end
```

Stack



Globals

foo	"Hello"
bar	6.28
add	<function>

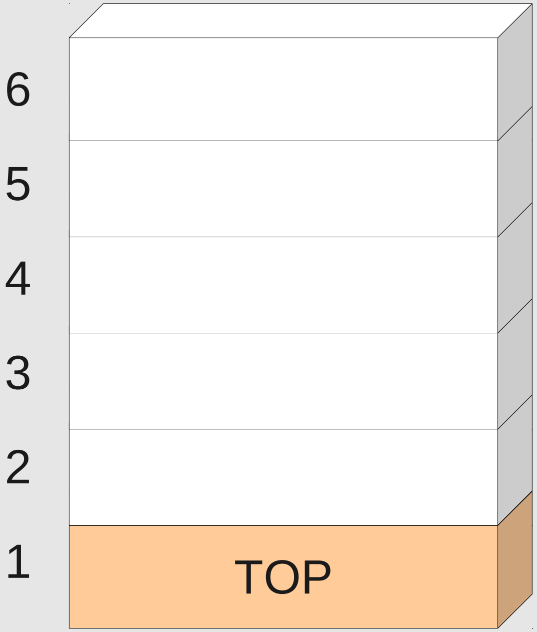
main.cpp

```
lua_getglobal(L, ”foo”);

const char* foo =
   lua_tostring(L, -1);
```

script.lua

```
foo = ”Hello”
bar = 6.28

function add(a,b)
   return a+b
end
```

Stack



Globals

foo	”Hello”
bar	6.28
add	<function>

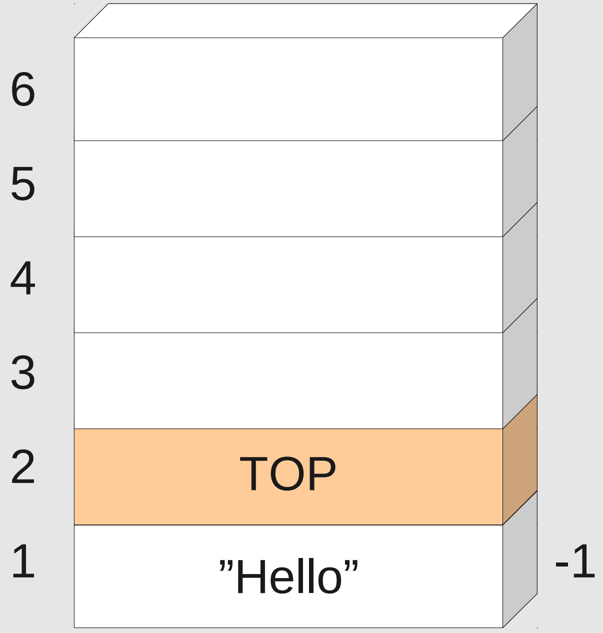
main.cpp

```
lua_getglobal(L, ”foo”);

const char* foo =
   lua_tostring(L, -1);
```

script.lua

```
foo = ”Hello”
bar = 6.28

function add(a,b)
   return a+b
end
```

Stack



Globals

foo	"Hello"
bar	6.28
add	<function>

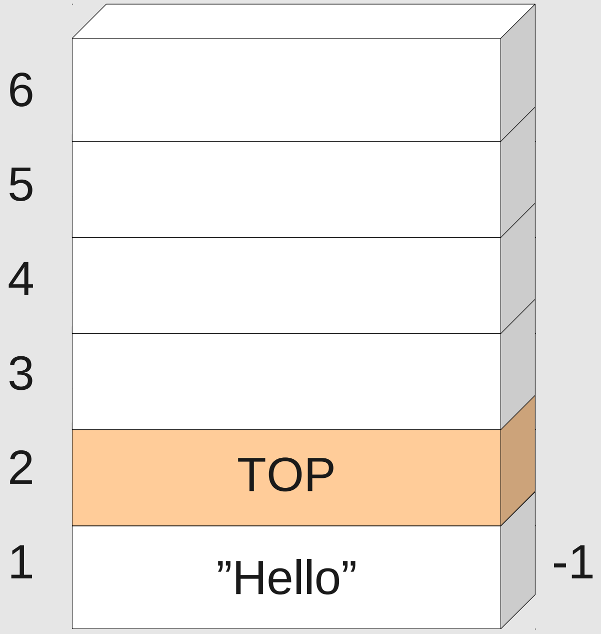
main.cpp

```
lua_getglobal(L, ”foo”);

const char* foo =
   lua_tostring(L, -1);
```

script.lua

```
foo = ”Hello”
bar = 6.28

function add(a,b)
   return a+b
end
```

Stack



Globals

foo	"Hello"
bar	6.28
add	<function>

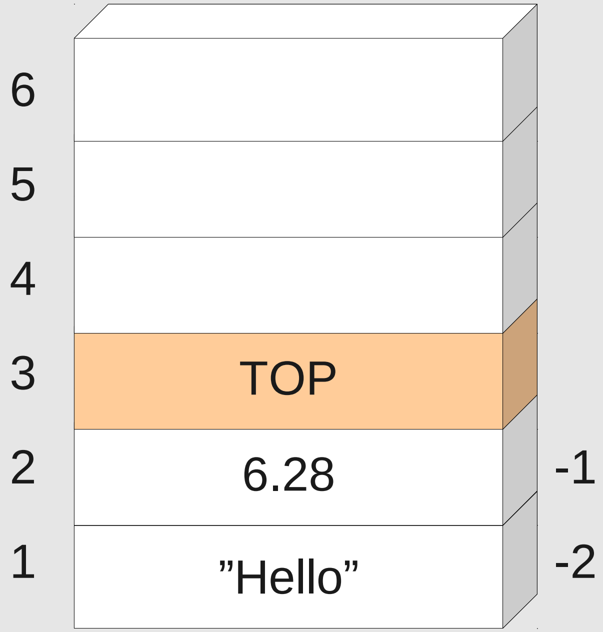
main.cpp

```
lua_getglobal(L, ”bar”);

float bar =
   lua_tonumber(L, -1);
```

script.lua

```
foo = ”Hello”
bar = 6.28

function add(a,b)
   return a+b
end
```

Stack



Globals

foo	"Hello"
bar	6.28
add	<function>

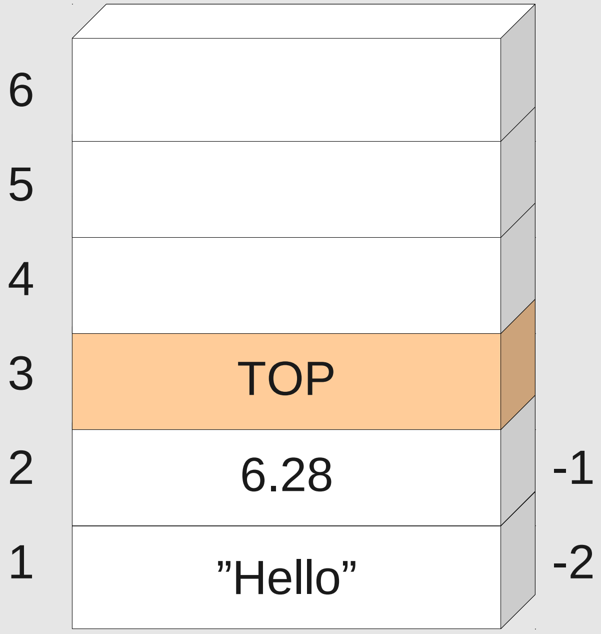
main.cpp

```
lua_getglobal(L, ”bar”);

float bar =
   lua_tonumber(L, -1);
```

script.lua

```
foo = ”Hello”
bar = 6.28

function add(a,b)
   return a+b
end
```

Stack



Globals

foo	"Hello"
bar	6.28
add	<function>

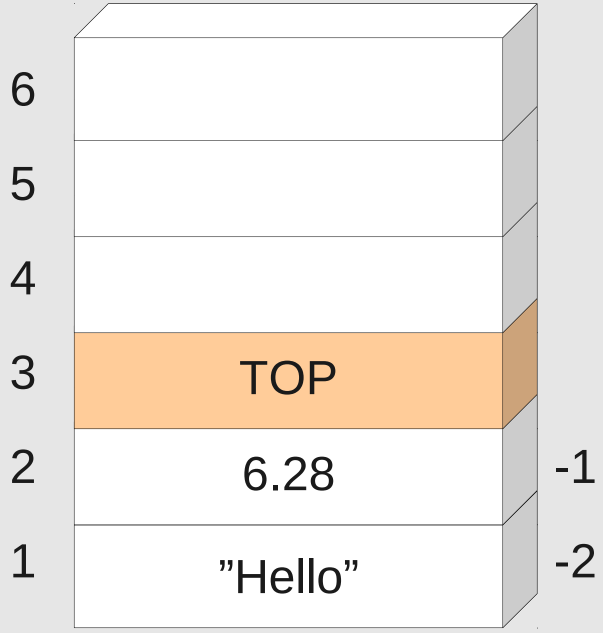
main.cpp

```
lua_pop(L, 2);
```

script.lua

```
foo = ”Hello”
bar = 6.28

function add(a,b)
   return a+b
end
```

Stack



Globals

foo	”Hello”
bar	6.28
add	<function>

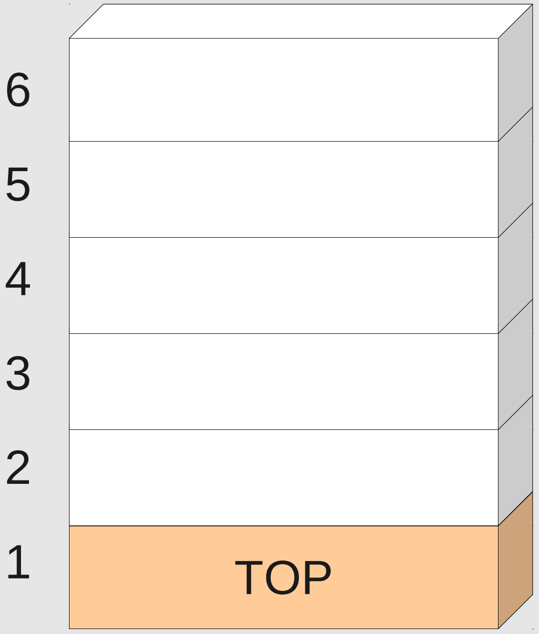
main.cpp

```
lua_pop(L, 2);
```

script.lua

```
foo = ”Hello”
bar = 6.28

function add(a,b)
   return a+b
end
```

Stack



Globals

foo	”Hello”
bar	6.28
add	<function>

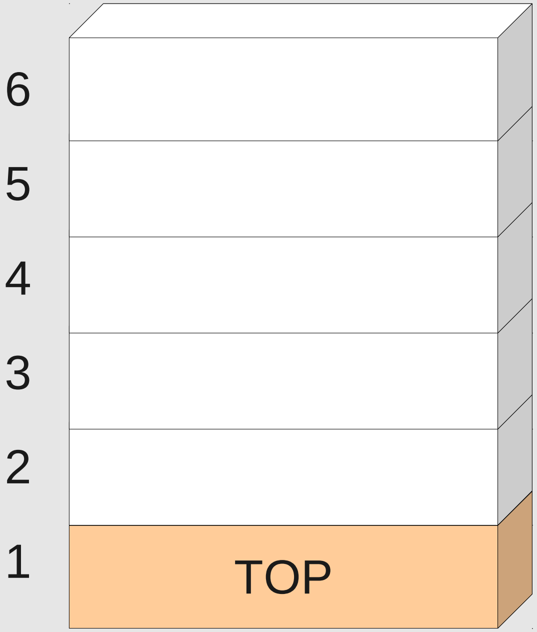
main.cpp

```cpp
lua_getglobal(L, "add");
lua_pushnumber(L, 3.5);
lua_pushnumber(L, 8.5);
lua_call(L, 2, 1);

float result =
   lua_tonumber(L, -1);
```

script.lua

```lua
foo = "Hello"
bar = 6.28

function add(a,b)
   return a+b
end
```

Stack



Globals

foo	"Hello"
bar	6.28
add	<function>

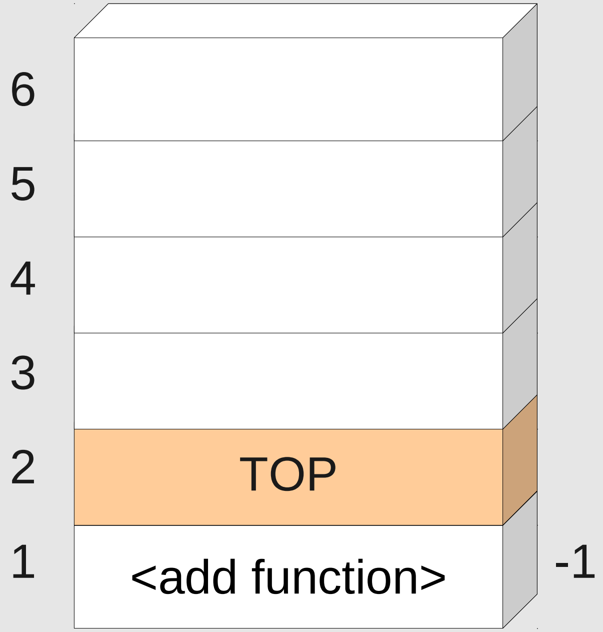
main.cpp

```
lua_getglobal(L, ”add”);
lua_pushnumber(L, 3.5);
lua_pushnumber(L, 8.5);
lua_call(L, 2, 1);

float result =
   lua_tonumber(L, -1);
```

script.lua

```
foo = ”Hello”
bar = 6.28

function add(a,b)
   return a+b
end
```

Stack



Globals

foo	”Hello”
bar	6.28
add	<function>

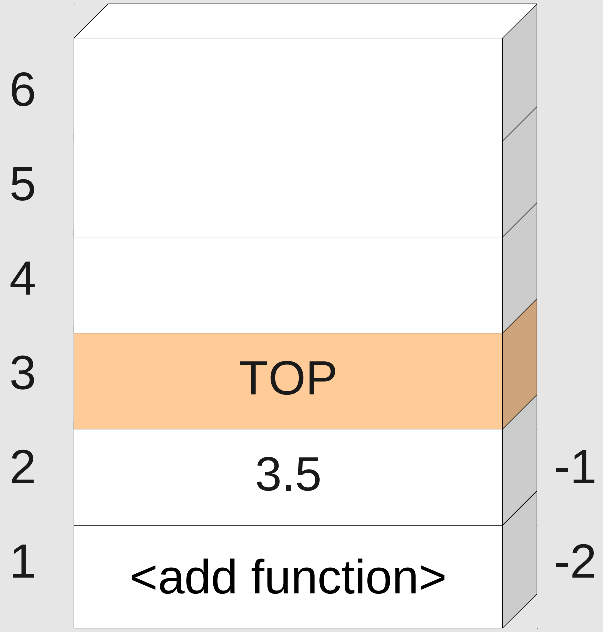
main.cpp

```
lua_getglobal(L, ”add”);
lua_pushnumber(L, 3.5);
lua_pushnumber(L, 8.5);
lua_call(L, 2, 1);

float result =
   lua_tonumber(L, -1);
```

script.lua

```
foo = ”Hello”
bar = 6.28

function add(a,b)
   return a+b
end
```

Stack



Globals

foo	”Hello”
bar	6.28
add	<function>

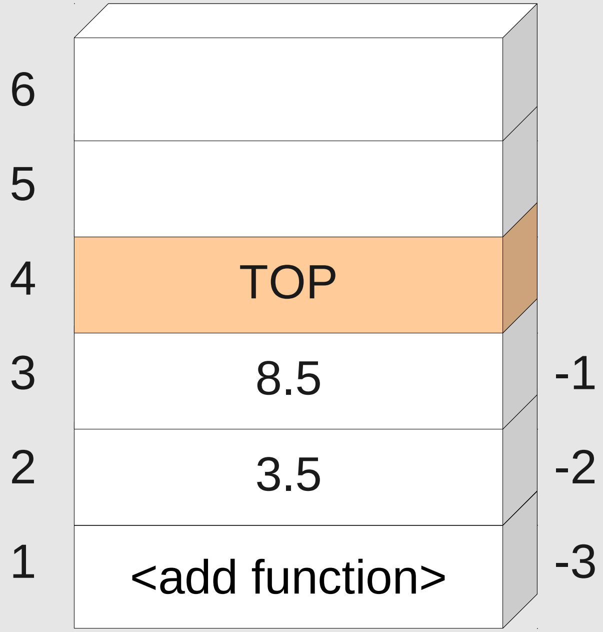
main.cpp

```
lua_getglobal(L, "add");
lua_pushnumber(L, 3.5);
lua_pushnumber(L, 8.5);
lua_call(L, 2, 1);

float result =
   lua_tonumber(L, -1);
```

script.lua

```
foo = "Hello"
bar = 6.28

function add(a,b)
   return a+b
end
```

Stack



Globals

foo	"Hello"
bar	6.28
add	<function>

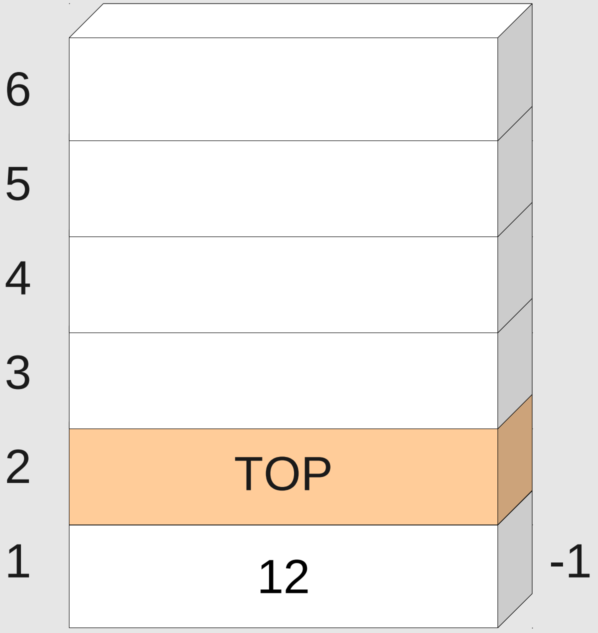
main.cpp

```
lua_getglobal(L, ”add”);
lua_pushnumber(L, 3.5);
lua_pushnumber(L, 8.5);
lua_call(L, 2, 1);

float result =
   lua_tonumber(L, -1);
```

script.lua

```
foo = ”Hello”
bar = 6.28

function add(a,b)
   return a+b
end
```

Stack



Globals

foo	”Hello”
bar	6.28
add	<function>

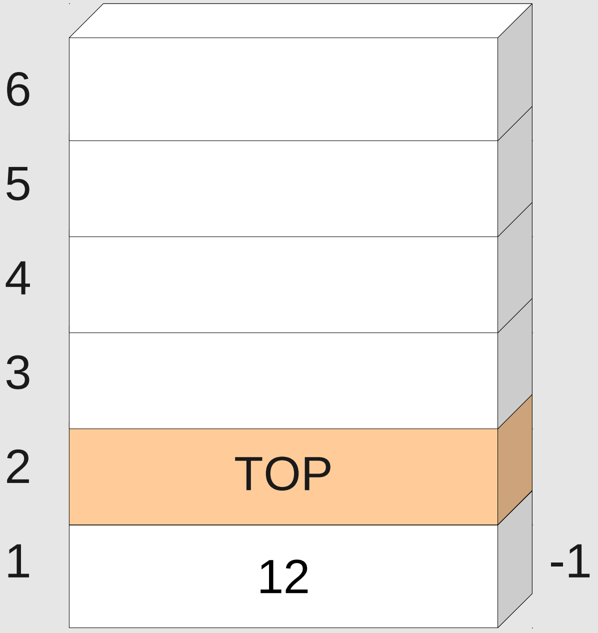
main.cpp

```
lua_getglobal(L, ”add”);
lua_pushnumber(L, 3.5);
lua_pushnumber(L, 8.5);
lua_call(L, 2, 1);

float result =
   lua_tonumber(L, -1);
```

script.lua

```
foo = ”Hello”
bar = 6.28

function add(a,b)
   return a+b
end
```

Stack



Globals

foo	”Hello”
bar	6.28
add	<function>

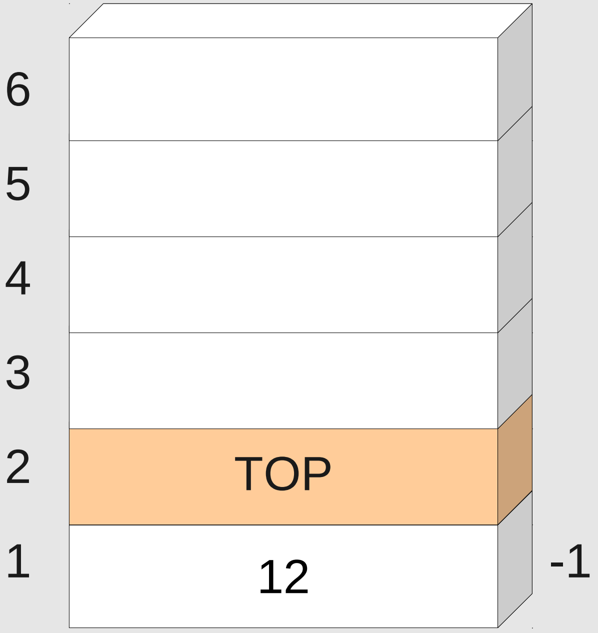
main.cpp

```cpp
lua_pop(L, 1);
```

script.lua

```lua
foo = ”Hello”
bar = 6.28

function add(a,b)
   return a+b
end
```

Stack



Globals

foo	”Hello”
bar	6.28
add	<function>

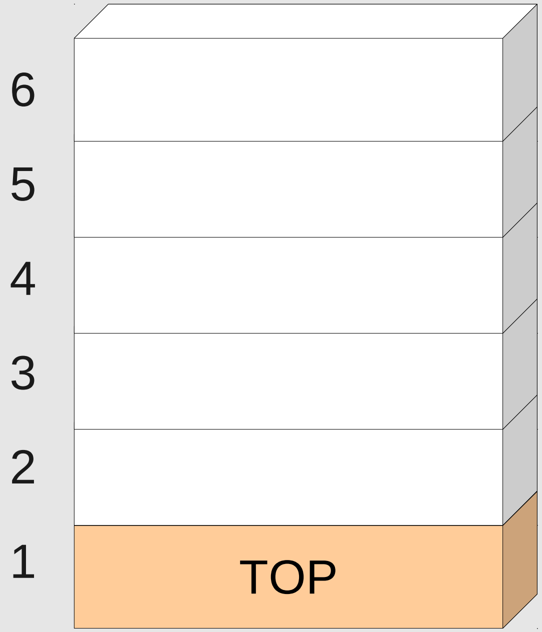
main.cpp

```
lua_pop(L, 1);
```

script.lua

```
foo = ”Hello”
bar = 6.28

function add(a,b)
   return a+b
end
```

Stack



Globals

foo	”Hello”
bar	6.28
add	<function>

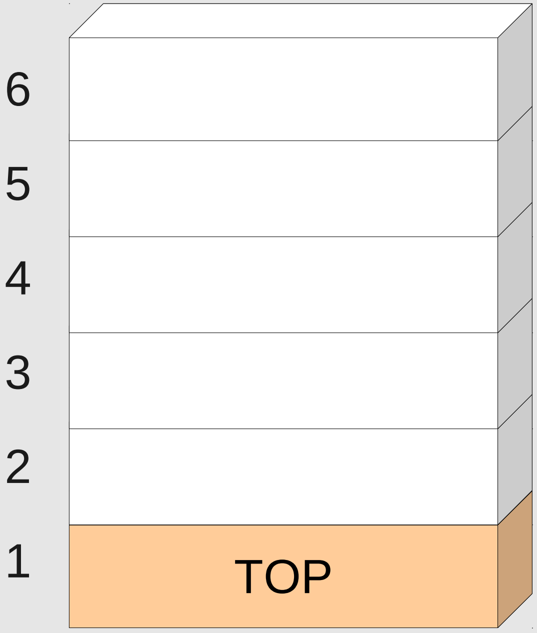
main.cpp

```
lua_pushstring(L, ”World”);

lua_setglobal(L, ”baz”);
```

script.lua

```
foo = ”Hello”
bar = 6.28

function add(a,b)
   return a+b
end
```

Stack



Globals

foo	”Hello”
bar	6.28
add	<function>

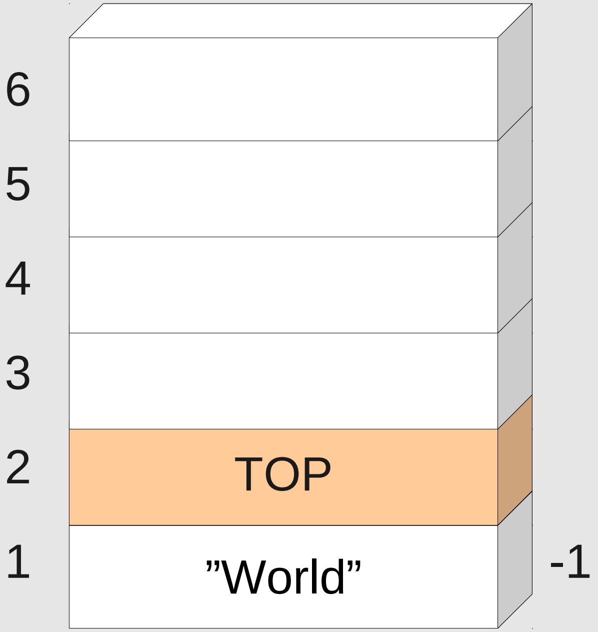
main.cpp

```
lua_pushstring(L, ”World”);

lua_setglobal(L, ”baz”);
```

script.lua

```
foo = ”Hello”
bar = 6.28

function add(a,b)
   return a+b
end
```

Stack



Globals

foo	"Hello"
bar	6.28
add	<function>

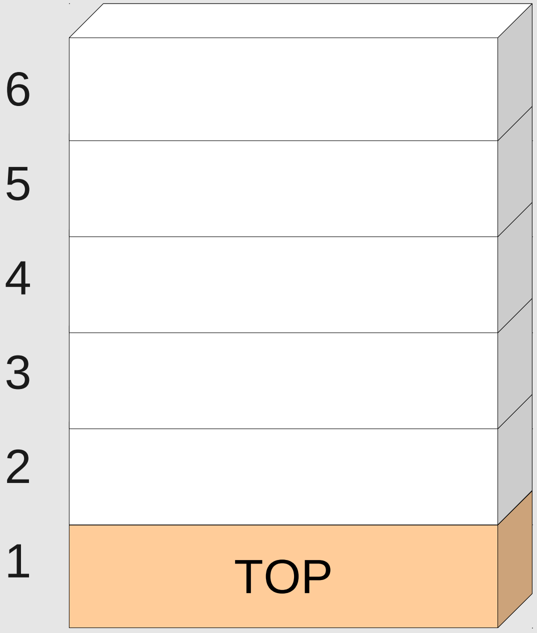
main.cpp

```
lua_pushstring(L, ”World”);

lua_setglobal(L, ”baz”);
```

script.lua

```
foo = ”Hello”
bar = 6.28

function add(a,b)
    return a+b
end
```

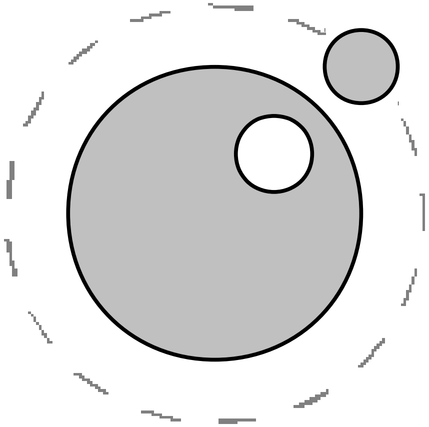
Stack



Globals

foo	”Hello”
bar	6.28
add	<function>
baz	”World”

Demo



Scripting Alternatives [1]

- Tcl
- Boost::Python

- Squirrel Language
- Pawn
- GameMonkey

- AngelScript

[1] http://codeplea.com/game-scripting-languages