

Introducción al lenguaje Lua

Pablo Martínez López
Hugo Prieto Tárrega
Manuel de Castro Caballero

GUI
Grupo Universitario de Informática
Escuela de Ingeniería Informática, Universidad de Valladolid

Hour of Code, 2019/20
Miércoles, 27 de Noviembre de 2019

1 ¿Qué es Lua?

2 Sintaxis y particularidades de Lua

3 Agradecimientos

¿Qué es Lua?

- Lenguaje compacto que le permite ser multiplataforma.
- Lenguaje multiparadigma: Imperativo, funcional, orientado a objetos...
- Lenguaje pensado para apoyar a C en las cosas que hace mal. Pensado para hacer scripting.
- Las variables no tienen tipo. Existen 8 tipos fundamentales.
- Utilizado para hacer scripts en múltiples videojuegos: League of Legends, World of Warcraft...



Figura: Logo de Lua

¿Por qué Lua existiendo Python?

- Más fácil de aprender que Python. Menos restricciones y más visual.
- Al contrario que Python, Lua es **compilado**, no interpretado.
- Ambos enfocados a scripting, pero cada uno en un ámbito distinto.
- Estructuras de datos y objetos son lo mismo en Lua, al contrario que Python.
- Prototipos.



Figura: Python vs Lua

nil	Non value
boolean	true
number	10
string	"Hello World"
userdata	io.stderr
function	print
thread	thread
table	{ }

1 ¿Qué es Lua?

2 Sintaxis y particularidades de Lua

- Diferencias con otros lenguajes
- Orientación a objetos en Lua

3 Agradecimientos

- Por defecto, **se empieza en 1**, no en 0.
- Las estructuras de control de flujo son de la forma *if...then...end*; *for...do...end*. También se aplica a funciones.
- Todas las variables se asumen como globales a no ser que se especifique lo contrario con **local** antes de la variable.
- No se niega con \neq , sino con \neq .
- La estructura de datos básica no es el array, es la tabla.



Figura: Salvo en este caso, Lua mola.

- While...do
- Repeat...until
- For...do
- Iteradores (índice, valor)
 - pairs: sólo aporta orden a claves numéricas.
 - ipairs: sólo claves numéricas. Para al encontrar un **nil**.
- Todas acaban en **end**. También aplicable a funciones.

- Estructura básica de datos en Lua. Llevado a lo conocido sería un array multitipado el cual es tratado como un objeto (al igual que un array convencional).
- Son accesibles por índices.
- Pueden ser creados vacíos o a partir de valores.
- También pueden ser creadas como un mapa: $a = \{x = 10, y = 20\}$
- En general, una combinación de varias estructuras de datos.
- La longitud se puede expresar con el operador `#`. (`#a`)

- Los metamétodos redefinen métodos comunes a las tablas.
- Las metatablas se basan en prototipos que definen los valores por defecto de las tablas.
- Al crear una tabla, se no se declaran ciertos valores, se comprueba el **__index** de la tabla y se asignan los valores por defecto.
- **__index** puede ser declarado como función o como tabla (más compacto y sencillo).

- No existe el concepto de clase. Creamos un objeto prototipo y crearemos el resto de objetos a partir de él.
- Declaramos `__index` en el constructor haciendo que una tabla objeto sea su propia metatabla.
- En el constructor pasamos como argumento una variable para referenciar a la metatabla. Devolvemos esa misma variable transformada en tabla.
- Para referirse a uno mismo utilizamos **self**.
- **__index** refiere a los atributos (prototipo) y **setmetatable** a los métodos.
- Para referirse a parámetros del objeto utilizamos `'.'`.

1 ¿Qué es Lua?

2 Sintaxis y particularidades de Lua

3 Agradecimientos

Esperamos que el taller haya sido ameno, que hayáis aprendido y sobre todo que os hayamos despertado las ganas de investigar un poco más por vuestra cuenta.

Los programas comentados realizados en el taller así como las diapositivas las podéis encontrar en:

<https://github.com/HylianPablo/LuaInicialHoC>

Contacto: @HylianPablo @Erizo_Da_Funk @bomilk

Podéis pasaros por el GUI en cualquier momento a preguntar (Puerta enfrente nada más subir las escaleras al primer piso por el ala de informática). Redes sociales:

- Twitter: @GUI_UVa
- Instagram: @gui_uva