

Pràctica 0

> Lua Interpreter

En tractar-se d'un llenguatge interpretat, l'execució de scripts i programes en Lua presenta diferències respecte els llenguatges compilats clàssics. Els scripts es llegeixen mitjançant un software específic, anomenat intèrpret, que tradueix i executa les ordres en temps real. Aquesta singularitat permet també l'execució de l'intèrpret de forma interactiva (stand-alone) mitjançant una consola, on la introducció i execució d'ordres es fa de forma immediata.

En aquesta pràctica, introduïrem l'intèrpret de Lua (versió actual 5.4.0), i practicarem en l'execució de scripts senzills de forma interactiva i interpretada.

1. Objectius

Els objectius d'aquesta pràctica són els següents:

- Introduir el concepte de llenguatge interpretat, i les seves diferències principals respecte els llenguatges compilats.
- Introduir l'intèrpret de Lua (versió actual 5.4.0), el seu funcionament bàsic i les seves opcions d'execució.
- Introduir l'execució de *scripts* en mode interactiu mitjançant consola de comandes i l'intèrpret de Lua
- Introduir les primeres nocions del llenguatge Lua, i les seves característiques bàsiques i algunes funcions senzilles.
- Realitzar els primers *scripts* en Lua, tan en mode interpretat com interactiu

2. Software i eines necessàries

- Intèrpret de Lua (versió actual 5.4.3)
<http://lua-users.org/wiki/LuaBinaries>
- Consola de comandes del sistema operatiu (MS Windows: cmd)

3. Entrega i avaluació

La pràctica es desenvoluparà de forma **individual**.

El codi ha de ser clar, entenedor, compacte i fàcilment depurable. És recomanable utilitzar comentaris per explicitar el funcionament de determinades parts del codi, sobretot aquelles en què la funcionalitat no s'aprecia a simple vista. Cal assegurar-se d'haver eliminat codi de prova si al final no s'executa i no forma part de la solució final.

Cada entrega consistirà en el fitxer o fitxers de codi font (.lua), a través del campus virtual, a més de la resta de fitxers que es

demani en cada cas. **Cada fitxer de codi font i/o text inclourà explícitament (com a comentari) el nom del/s autor/s de la pràctica.**

A banda de l'avaluació dels fitxers de codi font entregats, es podrà realitzar una avaluació del treball realitzat durant les sessions de pràctiques.

En aquesta avaluació, el professor podrà realitzar preguntes, consultes i comprovacions sobre el desenvolupament en la resolució de la pràctica al llarg de les diverses sessions.

També es podrà realitzar consultes o demanar una breu presentació oral de la practica, un cop realitzada l'entrega. En aquesta presentació caldrà demostrar la funcionalitat del programa (encara que no estigui acabat), exposar la solució desenvolupada, les característiques principals del programa, les dificultats sorgides i la manera de resoldre-les.

Totes les actuacions anteriors formaran part del conjunt l'avaluació de la pràctica.

L'avaluació de la pràctica **és individual**, i en cas de les pràctiques que es realitzen en grup, pot ser diferent per a cada membre del grup, en funció de l'avaluació del treball realitzat en les sessions pràctiques i la presentació oral.

4. Bibliografia i referències

- Lua Official website:
<https://www.lua.org/>
- Lua 5.4 – Reference Manual
<https://www.lua.org/manual/5.4/>
- Ierusalimsky, R. *Programming in Lua*. (Third edition). Rio de Janeiro, 2013.
- List of DOS commands
https://en.wikipedia.org/wiki/List_of_DOS_commands
- Introduction to the Windows Command Prompt
<https://www.bleepingcomputer.com/tutorials/windows-command-prompt-introduction/>

5. Desenvolupament de la pràctica

5.1. Lua Interpreter – Mode Interactiu

5.1.1. Instruccions

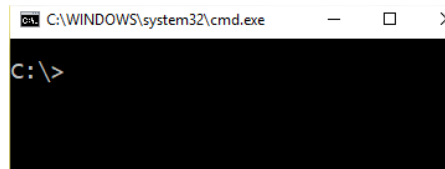
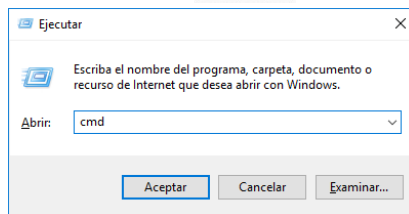
1. Creeu una carpeta de treball al vostre ordinador (Per exemple, *Prac0*).

2. Descarregueu l'interpret de Lua (versió actual 5.4.3), de <http://lua-users.org/wiki/LuaBinaries>, per a Windows la versió més actual està dins l'enllaç "Lua - joedf's Builds" (<https://joedf.ahkscript.org/LuaBuilds/>).

Copieu el contingut del zip a la carpeta de treball. Només cal els fitxers lua.exe i lua54.dll.

3. Executeu la consola de comandes del sistema operatiu

MS Windows:  +  + 'cmd' + Enter



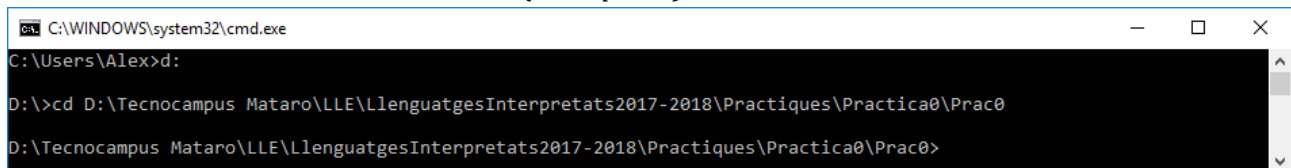
Us apareixerà la consola de comandes 'clàssica' del sistema operatiu (la ubicació per defecte pot ser diferent en cada usuari)

4. Desplaceu-vos a la carpeta de treball (*Prac0*) en la consola de comandes.

MS Windows: 'cd' + path

Si teniu el fitxers en una altra unitat que no sigui C:, primer s'ha de canviar la unitat.

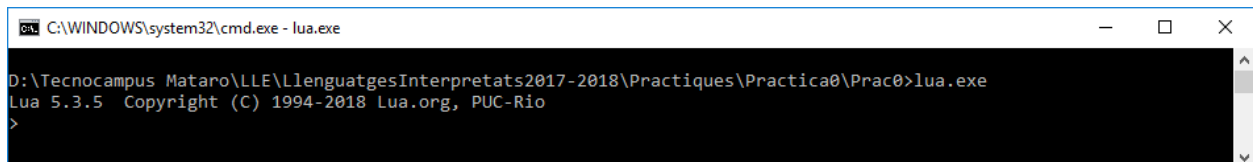
MS Windows: LLETRA_UNITAT: (exemple d:)



(Consulteu bibliografia per conèixer més comandes i opcions de la consola)

5. Executeu l'interpret de Lua situat en la mateixa carpeta, en la consola de comandes.

MS Windows: 'lua.exe'



Heu entrat en mode interactiu, ja podeu començar a executar ordres directament en llenguatge Lua. (Per tancar l'interpret de Lua i sortir del mode interactiu: *CTRL + Z + Enter*)

5.1.2. - Exercicis

- a) Escriviu i executeu el primer *script* "Hello World!" directament des de la consola de comandes. Naturalment, el programa ha d'imprimir per pantalla "Hello World!". Busqueu la sintaxi de la funció `print()` a la bibliografia.

```
> print("Hello World!")  
Hello World!
```

- b) Proveu d'executar l'*script* "Hello World!", modificant la sintaxi de la funció `print`: elimineu els parèntesis . Funciona correctament ? Perquè ? Quines precaucions cal tenir vist el resultat ? Busqueu la resposta a la bibliografia

```
> print "Hello World!"  
Hello World!
```

- Funciona correctamente por que el interprete de lua acepta una string como parámetro de la función `print` sin la necesidad de pasarlo entre paréntesis.
 - Las precauciones a tomar serían indicar con claridad el tipo de valor que se le pasa en el caso de no usar paréntesis, en este caso con las `""` para indicar que es una string.
- c) Modifiqueu ara el text a aparèixer pantalla. Mostreu: "HelloWorld" (sense espais ni signe de exclamació), mitjançant la funció `print`. Amb aquest nou text, elimineu les cometes (") i executeu l'*script*. Què apareix en pantalla ? Quin és el motiu ? Quin és el significat del resultat ? Busqueu ajuda a la bibliografia i raoneu la resposta.

```
> print(HelloWorld)  
nil
```

- Por pantalla se muestra como resultado «nil», significa nulo, el motivo de «nil» es debido a que HelloWorld es interpretado como una variable
 - El resultado significa que la variable HelloWorld tiene un valor nulo debido a no haber sido asignado antes de usarse
- d) Escriviu i executeu "Hello World!", modificant la sintaxi de la funció `print`: Separeu les paraules en dos textos ("Hello " i "World!") i utilitzeu l'operador de concatenació, en una única sentència i una única crida a `print`. Consulteu la bibliografia.

```
> print("Hello " .. "World!")  
Hello World!
```

- e) Busqueu a la bibliografia una funció per esborrar la pantalla de la consola de comandes. Ajuda: Es tracta d'executar una ordre del sistema operatiu. Feu el mateix amb una funció per sortir de l'interpret de Lua.

- Para limpiar la pantalla

```
> os.execute("cls")
```

- Para cerrar el interprete

```
> os.exit()
```

- f) Assigneu a una variable el text 'Hello World!' com a valor. Imprimiu el text per pantalla. Com és la declaració de variables en Lua ? Com s'assigna el tipus de dades que contindrà ? Trobeu una manera per esborrar el valor de la variable, i comproveu de nou el resultat per pantalla.

```
> mVar = "Hello World!"  
> print(mVar)  
Hello World!  
> mVar = 3  
> print(mVar)  
3
```

- El tipo de la variable va implícito en el valor, no en el contenedor, de tal forma que «mVar» no tiene un tipo de valor fijo para contener valores.
- g) Assigneu a una variable el text original, i imprimiu el valor per pantalla. A continuació, assigneu-li un valor numèric, i imprimiu novament el resultat per pantalla. Ha aparegut algun error d'incoherència de tipus de dades ? Expliqueu el motiu.

```
> mVar = "Hello World!"  
> print(mVar)  
Hello World!  
> mVar = 3  
> print(mVar)  
3
```

- No aparece ningún error de incoherencia ya que la bibliografía de lua declara lo siguiente: "Lua is a dynamically typed language. This means that variables do not have types; only values do. There are no type definitions in the language. All values carry their own type."
- h) Busqueu a la bibliografia una funció que permeti determinar el tipus de dades d'una variable, i mostrar-lo per pantalla. Comproveu-ne el funcionament.

```
> type(mVar)  
number
```

- La función type(v) según la bibliografía: "Returns the type of its only argument, coded as a string"

- i) Escriuiu en mode interactiu un *script* que assigni valors numèrics a dues variables, i retorni la suma, la resta i el producte dels dos nombres. Ajuda: es pot fer en tres línies de codi.

```
> function SumRestMulti(a,b)
>> return a+b,a-b,a*b
>> end
> print(SumRestMulti(2,2))
4      0      4
```

- j) Escriuiu en mode interactiu un script que demani la introducció de dos valors numèrics pel teclat, els assigni a dues variables, i retorni la suma dels dos nombres. Us caldrà buscar una funció de lectura de teclat.

```
> function InputSum()
>> print("Type a value and press Enter")
>> a = io.read()
>> print("Type a value and press Enter")
>> b = io.read()
>> return a+b
>> end
> print(InputSum())
Type a value and press Enter
10
Type a value and press Enter
25
35
```

A la mateixa carpeta de treball on teniu l'interpret de Lua, creeu un fitxer de text on escriureu un *script* extern (per exemple, *prac0.lua*). No és necessari que tingui extensió *.lua*, però aquesta extensió identifica clarament la tipologia del fitxer. Creeu un fitxer diferent per cada un dels *scripts* a desenvolupar. Per desenvolupar els scripts podeu utilitzar qualsevol editor de text (*MS Windows*: Notepad)

ractiques > Practica0 > Prac0				Buscar en Prac0	
Nombre	Fecha de modifica...	Tipo	Tamaño		
lua.exe	17/09/2018 10:09	Aplicación	68 KB		
lua53.dll	17/09/2018 10:11	Extensión de la apl...	593 KB		
prac0.lua	17/09/2018 12:09	Archivo LUA	0 KB		

- o) Desenvolpeu els *scripts* 'Hello World!' i de suma de nombres en un fitxer de text independent, i executeu-los des de l'interpret en mode interactiu. Busqueu una funció a la bibliografia per executar *scripts* externs.

```
prac0.o.lua
1  print("Hello World!")
```

```
prac0.j.lua
1  print("Type a value and press Enter")
2  a = io.read()
3  print("Type a value and press Enter")
4  b = io.read()
5  return a+b
```

- Los ficheros *.lua se ejecutan con la función «dofile(t)».

```
> dofile("prac0.o.lua")
Hello World!
> dofile("prac0.j.lua")
Type a value and press Enter
1
Type a value and press Enter
2
3
```

- p) Repetiu l'operació anterior amb la funció `loadfile()`. Quina diferència hi ha amb la funció anterior?

```
> loadfile("prac0.o.lua")
function: 00babf68
```

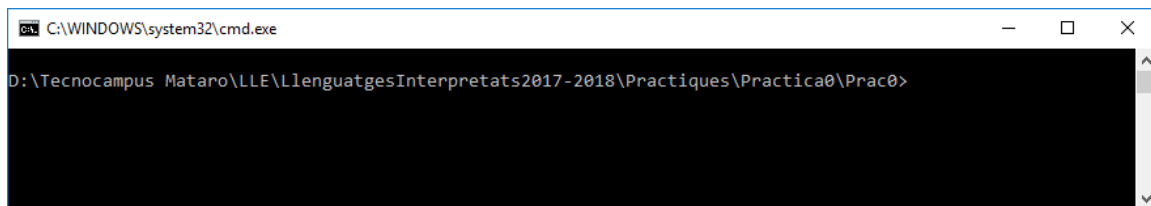
- Carga el chunk de código y permite guardarlo en un contenedor para ejecutarlo con la función `load()` o añadiendo `()`
- q) Quins tipus d'estructures de control hi han a Lua? Com funcionen? Indica un exemple de cada una.
- Bucle básico, puede permitir ninguna ejecución
 - `stat ::= while exp do block end`
 - Bucle básico, permite siempre una ejecución
 - `stat ::= repeat block until exp`
 - Condicional básico
 - `stat ::= if exp then block {elseif exp then block} [else block] end`
 - Bucle complejo, permite recorrer una cantidad X de valores
 - `stat ::= for Name '=' exp [, exp] [, exp] do block end`

- Salto básico, permite saltar a otra línea de código para continuar la ejecución desde ese punto
 - `stat ::= goto Name`
 - `stat ::= label`
 - `label ::= '::' Name '::'`
- Interrupción, permite detener la ejecución de un código dentro de su pipe
 - `stat ::= break`
- Retorno básico, termina una pipe devolviendo un valor X
 - `stat ::= return [explist] [';']`

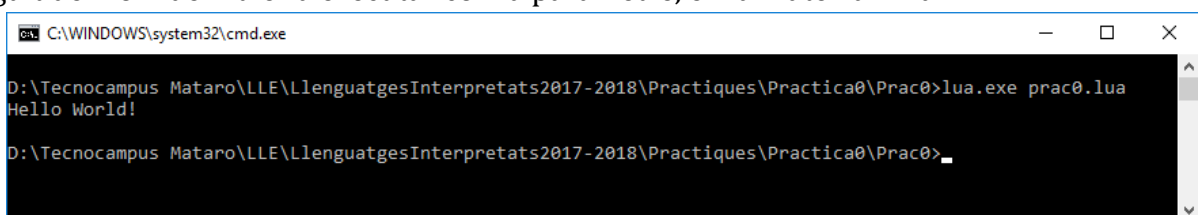
5.2. Lua Interpreter – Mode Interpretat

5.2.1. Instruccions

1. Un cop desenvolupats els *scripts*, per executar-los amb l'interpret de forma interpretada cal que obriu de nou la consola de comandes del sistema operatiu, i us situeu a la carpeta de treball.



2. Per *executar* un script extern, cal escriure una comanda on s'executa l'interpret de Lua, seguit del nom del fitxer a executar com a paràmetre, en la mateixa línia.



Observeu com en aquest cas, un cop s'ha executat l'script, per defecte es surt de l'interpret de Lua i es retorna a la consola de comandes del sistema.

5.2.2. Exercicis

- a) Determineu amb l'ajuda de la bibliografia, com passar-li a l'script arguments en la crida al fitxer extern. Com s'utilitzen aquest arguments dins el codi Lua? Quina peculiaritat té el primer argument?

```
.\lua.exe .\prac0.args.lua a b c
```


- Dentro del código, se accede a los argumentos pasados en ejecutan mediante el uso de la tabla `args[]`, desde la posición 1 hasta la posición "n", donde "n" es el número de argumentos.
- La peculiaridad del primer argumento, según la bibliografía es que es el nombre del script y los argumentos empiezan desde la posición 1 de la tabla `args[]`.
- De tal forma, también indica la bibliografía, que los parámetros de ejecución de lua y sus opciones se encuentran en las posiciones -2 y -1 de la tabla `args[]`

- b) Desenvolueu un script que mostri per pantalla «Hello World! I'm X!» on X és el vostre nom. Utilitzeu la variable de sistema que permet recuperar els arguments d'entrada, per tal que el nom a mostrar per pantalla sigui un paràmetre d'entrada en la línia d'execució del programa. Exemple crida: `>lua.exe prac0.lua Alex`


- Ejecución

```
.\lua.exe .\prac0.args.lua Ricard  
Hello World! Im Ricard
```


- Código

```
Practiques > P0 >  prac0.args.lua  
1 print("Hello World! Im " .. arg[1])
```

- c) Desenvolpeu un script que mostri per pantalla el factorial d'un nombre determinat, Utilitzeu la variable de sistema que permet recuperar els arguments d'entrada, per tal que el nombre a utilitzar sigui un paràmetre d'entrada en la línia d'execució del programa. Exemple crida: >lua.exe factorial.lua 4

```
Practiques > P0 >  prac0.factorial.lua
1  function factorial(n)
2      if (n == 1) then
3          return 1
4      end
5      return (n * factorial(n - 1))
6  end
7
8  b = tonumber(arg[1])
9
10 if (string.find(type(b), "number") ~= nil) then
11     print("Factorial de " .. arg[1] .. ": " .. factorial(b))
12 else
13     print("Introduce un número válido.")
14 end
```

- d) Desenvolpeu un script que mostri per pantalla els N primers termes de la successió de Fibonacci, on N és un nombre que es demana a l'usuari. Busqueu en la bibliografia les eines necessàries. Exemple crida: >lua.exe fibonacci.lua 5

```
Practiques > P0 >  prac0.fibonacci.lua
1  function reFibonacci(pos, table, exit)
2      pos = pos + 1
3      if (pos < 3) then
4          table[pos] = 1
5          table = reFibonacci(pos, table, exit)
6      elseif (pos ~= exit) then
7          table[pos] = table[pos-1] + table[pos-2]
8          table = reFibonacci(pos, table, exit)
9      end
10     return table
11 end
12
13 function fibonacci(b)
14     return reFibonacci(0, {}, b)
15 end
16
17 b = tonumber(arg[1])
18
19 if ((string.find(type(b), "number") ~= nil)) then
20     str = "Fibonacci de " .. arg[1] .. ": "
21     for i,v in ipairs(fibonacci(b)) do
22         str = str .. tostring(v) .. ", "
23     end
24     str = string.sub(str, 1, #str - 2)
25     print(str)
26 else
27     print("Introduce un número válido.")
28 end
29
30 |
```