

Pd-Ofelia suficiente para começar

Gabriel Haruo

A biblioteca Ofelia permite que scripts Lua sejam executados no Pure Data.

1 Sintaxe básica de Lua para Pure Data

1.1 Comentários

Comentários em Lua são iniciados com `--` e vão até o final da linha.

```
-- Isso é um comentario
```

1.2 Variáveis

Variáveis em Lua são definidas com a palavra-chave `local`.

```
local x = 10
```

1.2.1 Escopo de variáveis

Ao declarar uma variável no topo de um script, ela será global e poderá ser acessada por qualquer função no script.

Ao declarar uma variável dentro de uma função, ela será local e só poderá ser acessada dentro da função.

```
local x = 10
```

```
function minhaFuncao()
  local y = 20
  print('Consigo imprimir a variavel x: ' .. x)
  print('Consigo imprimir a variavel y: ' .. y)
end

print('Consigo imprimir a variavel x: ' .. x)
print('Nao consigo imprimir a variavel y: ' .. y)
```

1.3 Funções

Funções em Lua são definidas com a palavra-chave `function`.

```
function minhaFuncao()
  print('Executei a funcao minhaFuncao!')
end
```

1.3.1 Parâmetros de entrada

Parâmetros de entrada são definidos entre parênteses.

```
function minhaFuncaoComParametro(x)
  print('Executei a funcao minhaFuncaoComParametro')
  print(' com o parametro ' .. x)
end
```

1.4 Listas

Listas em Lua são definidas entre chaves. Para acessar um elemento da lista, use o índice do elemento entre colchetes.

Obs.: As listas em Lua são indexadas a partir de 1.

```
local lista = {10, 20, 30}
print('Primeiro elemento da lista: ' .. lista[1])
```

Listas em Ofelia devem ser inicializadas com a sintaxe `ofTable()`.

```
local lista = ofTable()
lista[1] = 10
lista[2] = 20
lista[3] = 30
print('Primeiro elemento da lista: ' .. lista[1])
```

1.5 Laços

Laços em Lua são definidos com as palavras-chave `for`, `do` e `end`.

```
for i=1, 10 do
  print(i)
end
```

1.6 Estruturas condicionais

Estruturas condicionais em Lua são definidas com as palavras-chave `if`, `else`, `elseif`, `then`, e `end`. Use `and` e `or` para combinar condições.

```
local x = 10

if x > 5 then
  print('x é maior que 5')
elseif x < 5 then
  print('x é menor ou igual a 5')
else
  print('x é igual a 5')
end
```

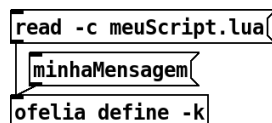
2 Executando scripts .lua no Pure Data

A biblioteca Ofelia registra automaticamente funções nomeadas como `ofelia.*` para serem usadas como **manipuladores de mensagens** no Pure Data.

Vamos exemplificar como isso funciona. Crie um arquivo `meuScript.lua` e defina a seguinte função:

```
function ofelia.minhaMensagem()
  print('Executei a funcao minhaMensagem!')
end
```

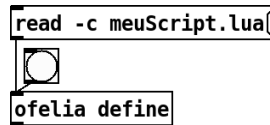
Crie um patch em Pure Data nomeado `meuPatch.pd` no mesmo diretório do script e adicione os seguintes objetos:



Clique na mensagem `read -c meuScript.lua` para carregar o script no Pure Data. Agora, clique na mensagem `minhaMensagem` para **executar a função do script com o nome da mensagem mandada**. A função `ofelia.minhaMensagem()` será executada e a mensagem `Executei a funcao minhaMensagem!` será impressa no console.

2.1 ofelia.bang()

Para executar uma função ao clicar em um objeto **bang**, é necessário adicionar um objeto **bang** ao patch e definir a função **ofelia.bang()** no script.



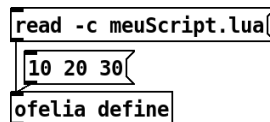
Adicione a seguinte função ao script **meuScript.lua**:

```
function ofelia.bang()  
  print('Executei a funcao ao ativar o bang!')  
end
```

Agora, ao clicar no objeto **bang**, a função **ofelia.bang()** será executada e a mensagem **Executei a funcao ao ativar o bang!** será impressa no console.

2.2 ofelia.list(lista)

Para executar uma função ao enviar uma lista de valores, é necessário adicionar uma mensagem com sua lista ao patch e definir a função **ofelia.list()** no script.



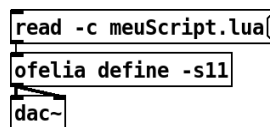
Adicione a seguinte função ao script **meuScript.lua**:

```
function ofelia.list(lista)  
  print('Executei a funcao ao enviar uma lista de valores!')  
  for i=1, #lista do  
    print('Valor ' .. i .. ' da lista: ' .. lista[i])  
  end  
end
```

Agora, ao enviar a mensagem **10 20 30**, a função **ofelia.list()** será executada e as mensagens **Executei a funcao ao enviar uma lista de valores!**, **Valor 1 da lista: 10**, **Valor 2 da lista: 20** e **Valor 3 da lista: 30** serão impressas no console.

2.3 ofelia.perform(bloco)

Para executar uma função a cada ciclo DSP, é necessário definir a função **ofelia.perform()** no script. Em seu patch, adicione um objeto **ofelia define** com a flag **-s11**. Essa flag indica que o objeto possui 1 entrada de sinal e 1 saída de sinal.



A função **ofelia.perform()** recebe um vetor de 64 amostras como parâmetro, que deve ser preenchido com as amostras de saída e retornado. Vamos preencher o bloco de amostras com um ruído branco. Para isso, adicione a seguinte função no script **meuScript.lua**:

```
function ofelia.perform(bloco)  
  for i=1, 64 do  
    bloco[i] = 2*math.random() - 1  
  end  
  return bloco  
end
```

Abaixe o volume! Pois ao conectar o objeto **ofelia define -s11** a um objeto **dac~**, um ruído branco será reproduzido.

2.3.1 Observações

- Evite usar a função `print()` dentro da função `ofelia.perform()`, pois o uso introduz pausas na saída de som.