



Introdução à Engenharia

ENG1000

Aula 04 – Introdução a Linguagem Lua 2016.1



Prof. Augusto Baffa
<abaffa@inf.puc-rio.br>



Linguagem Lua

- Lua é uma **linguagem de programação** projetada para dar suporte à programação procedimental em geral.
- Oferece um bom suporte para programação orientada a objetos, programação funcional e programação orientada a dados.
- É utilizada em diversos ramos da programação, como no desenvolvimento de jogos, controle de robôs, processamento de texto, etc.



PONTIFÍCIA
UNIVERSIDADE
CATÓLICA
DO RIO DE JANEIRO

Linguagem Lua

- Exemplos de **empresas que desenvolvem jogos** usando a linguagem Lua:
 - LucasArts, Blizzard, Microsoft, BioWare...
- Lua é inteiramente projetada, implementada e desenvolvida na **PUC-Rio**.
 - Nasceu e cresceu no **Tecgraf**, o Grupo de Tecnologia em Computação Gráfica da PUC-Rio.
 - Atualmente é desenvolvida no laboratório **Lablua**.



PONTIFÍCIA
UNIVERSIDADE
CATÓLICA
DO RIO DE JANEIRO

NIT

Núcleo de Inovação Tecnológica



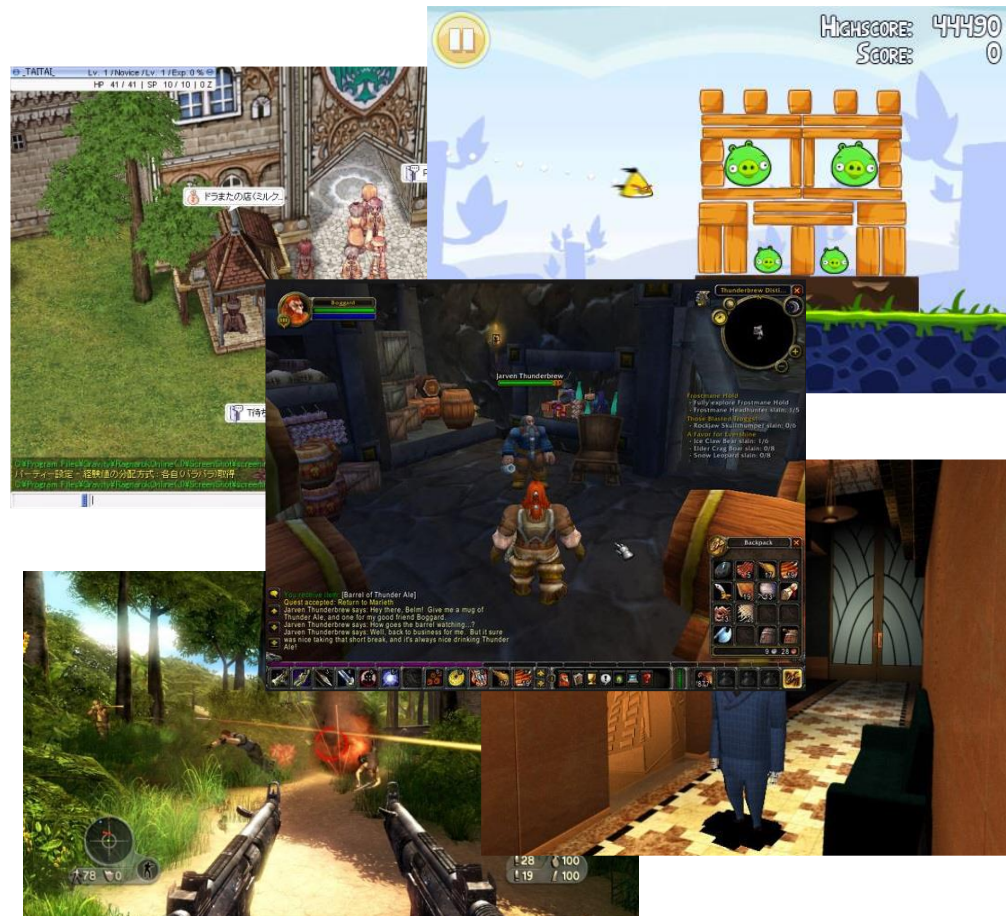
Instituto de Tecnologias de Software



Linguagem Lua

- Exemplos de jogos que utilizam Lua:

- Angry Birds
- Civilization V
- Far Cry
- Grim Fandango
- Ragnarok
- Tibia
- World of Warcraft



Por que escolher Lua?

- **É uma linguagem estabelecida e robusta:**
 - É usada em muitas aplicações industriais (e.g., Adobe's Photoshop Lightroom), com ênfase em sistemas embutidos (e.g., o middleware Ginga para TV digital) e jogos (e.g., World of Warcraft e Angry Birds).
 - Lua tem um sólido manual de referência e existem vários livros sobre a linguagem.
- **É rápida:**
 - Outras linguagens de script aspiram ser "tão rápidas quanto Lua".
 - Vários benchmarks mostram Lua como a linguagem mais rápida dentre as linguagens de script interpretadas.

Por que escolher Lua?

- **É portátil:**
 - Lua roda em todos os tipos de Unix e Windows, e também em dispositivos móveis (Android, iOS, BREW, Symbian, Windows Phone) e em microprocessadores embutidos (como ARM e Rabbit, para aplicações como Lego MindStorms).
- **É embutível:**
 - Lua tem uma API simples e bem documentada que permite uma integração forte com código escrito em outras linguagens.
- **É livre:**
 - Lua é software livre de código aberto, distribuída sob licença MIT.
 - Lua pode ser usada para quaisquer propósitos, incluindo propósitos comerciais, sem qualquer custo ou burocracia.

Exemplo de Programa em Lua

- O que faz este programa?

```
local cels
local fahr

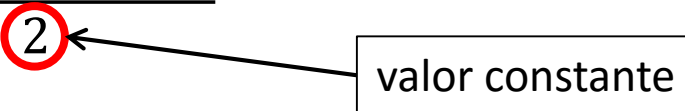
io.write("Digite a temperatura em Celsius: ")
cels = io.read()

fahr = 1.8 * cels + 32

io.write("Temperatura em Fahrenheit: ", fahr, "\n");
```


Variáveis e Constantes

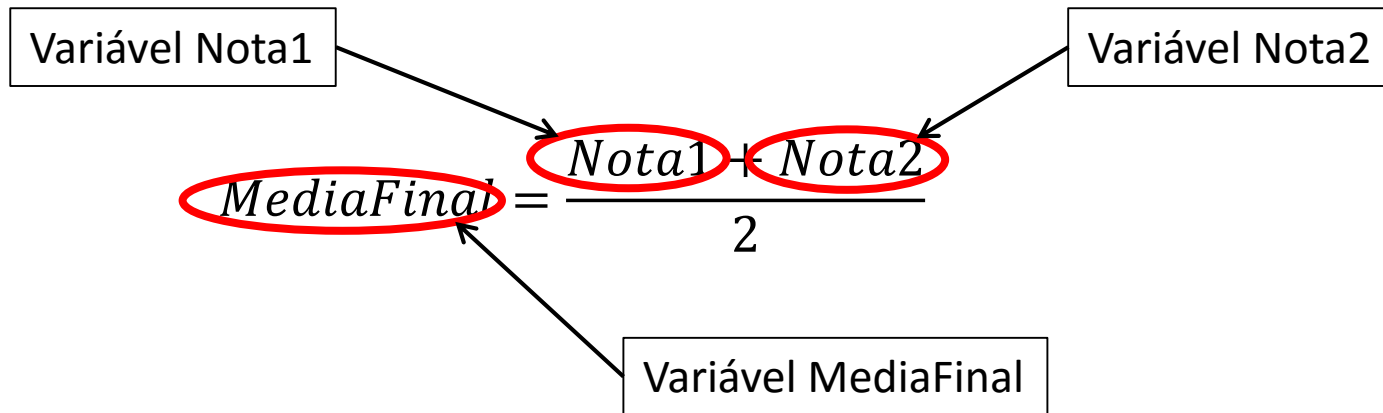
- **Variáveis** e **constantes** são os elementos básicos manipulados por um programa.
- **Constante** é um valor fixo que não se modifica ao longo da execução de um programa.

$$MediaFinal = \frac{Nota1 + Nota2}{2}$$


valor constante

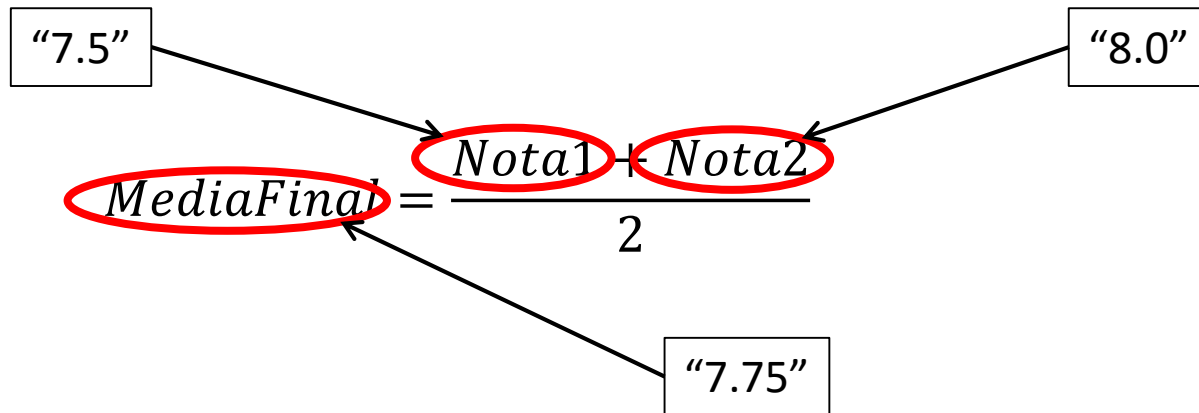
Variáveis

- **Variável** é um espaço reservado na memória do computador para armazenar um determinado tipo de dado.
- Variáveis recebem **nomes** para serem referenciadas e modificadas quando necessário.



Variáveis

- O **conteúdo de uma variável** pode se modificado ao longo da execução do programa.
- Embora uma variável possa assumir diferentes valores, ela só pode armazenar **um valor a cada instante**.



Variáveis em Lua

- **Variável** é um espaço reservado na memória do computador para armazenar um tipo de dado.
- Devem receber **nomes** para poderem ser referenciadas e modificadas quando necessário.
- Toda variável tem:
 - um nome
 - um tipo de dado
 - um valor
- **Restrição para nomes:** não é permitido começar o nome com um algarismo (0-9), alguns caracteres não são válidos (*, -, /, +, ...), e palavras reservadas não podem ser utilizadas (main, if, while, ...).

Variáveis em Lua

- Lua é uma linguagem **dinamicamente tipada**. Isto significa que variáveis não possuem tipos; somente valores possuem tipos.
 - Não existe definição de tipos na linguagem.
 - Todos os valores carregam o seu próprio tipo.
- Todos os valores em Lua são valores de primeira classe. Isto significa que todos os valores podem ser armazenados em variáveis, passados como argumentos para outras funções e retornados como resultados.

Tipos de Variáveis da Linguagem Lua

Tipo	Exemplos de Valores
<code>number</code>	0, 1, 2.3, -2.3
<code>string</code>	“oi”, “olá mundo”, “teste 123”, “”
<code>Boolean</code>	true, false
<code>function</code>	0x1234567
<code>table</code>	0x2345678
<code>Thread</code>	0x3456789
<code>Userdata</code>	0x4567890
<code>nil</code>	nil

Declaração de Variáveis em Lua

- Variáveis devem ser explicitamente declaradas.
- Variáveis podem ser declaradas em conjunto.
- Variáveis podem ser utilizadas sem serem declaradas (globais)

Exemplos:

```
local a      -- declara uma variável local chamada a
local b      -- declara uma variável local chamada b
local d, e   -- declara duas variáveis locais
local d = 5  -- declaração e inicialização da variável
```


Operadores Aritméticos

- **Operadores aritméticos** são usados para se realizar operações aritméticas com as variáveis e constantes.

Operação	Símbolo
Adição	+
Subtração	-
Multiplicação	*
Divisão	/
Resto da Divisão	%
Exponenciação	^

Exemplos:

operador de atribuição

total = preco * quantidade

media = (nota1 + nota2)/2

resultado = 3 * (1 - 2) + 4 * 2

resto = 4 % 2

res_exp = b ^2

Funções de Entrada e Saída em Lua

- **Função “write” da biblioteca “io”:** Permite a saída de dados, ou seja, a escrita de dados na tela.

```
io.write(constantes/variáveis/expressões...);
```

```
io.write(33)
```

tem como resultado a impressão da linha:

```
33
```

```
io.write("Valor = ", 33, " Total = ", 33 + 40);
```

saída:

```
Valor = 33    Total = 73
```

Funções de Entrada e Saída em Lua

- Impressão de texto:

```
io.write("Curso de Programação\n de Jogos");
```

exibe na tela a mensagem:

Curso de Programação

de Jogos

Funções de Entrada e Saída em Lua

- **Função “read” da biblioteca “io”:** Permite a entrada de dados, ou seja, a captura de valores fornecidos via teclado.

```
io.read()
```

```
local n  
n = io.read()
```

valor digitado pelo usuário é armazenado na variável n

- **Importante:** o valor lido é sempre tratado como um **texto**. Em alguns casos é necessário convertê-lo para um número com o comando **tonumber**:

```
local n  
n = tonumber(io.read())
```

Exemplo 01

- Escreva um programa que leia dois números inteiros e retorne a soma deles.

```
local numero1, numero2, resultado


io.write("Digite o primeiro numero: ")
numero1 = io.read()

io.write("Digite o segundo numero: ")
numero2 = io.read()

resultado = numero1 + numero2

io.write("Resultado da soma é ", resultado)
```

Nesses caso não convertemos os valores para números porque os operadores aritmético forçam a conversão automaticamente.



Exemplo 1 – Execução Passo-a-Passo

- Comando:

<u>numero1</u>	<u>numero2</u>	<u>resultado</u>	<u>saída</u>
????	????	????	

Exemplo 1 – Execução Passo-a-Passo

- Comando:

```
io.write("Digite o primeiro numero: ")
```

<u>numero1</u>	<u>numero2</u>	<u>resultado</u>	<u>saída</u>
----------------	----------------	------------------	--------------

????

????

????

Digite o primeiro numero:

Exemplo 1 – Execução Passo-a-Passo

- Comando:

```
numero1 = io.read()
```

<u>numero1</u>	<u>numero2</u>	<u>resultado</u>	<u>saída</u>
----------------	----------------	------------------	--------------

????	????	????	
------	------	------	--

Digite o primeiro numero:

15	????	????	
----	------	------	--

Exemplo 1 – Execução Passo-a-Passo

- Comando:

```
io.write("Digite o segundo numero: ")
```

<u>numero1</u>	<u>numero2</u>	<u>resultado</u>	<u>saída</u>
????	????	????	Digite o primeiro numero:
15	????	????	Digite o segundo numero:

Exemplo 1 – Execução Passo-a-Passo

- Comando:

```
numero2 = io.read()
```

<u>numero1</u>	<u>numero2</u>	<u>resultado</u>	<u>saída</u>
????	????	????	Digite o primeiro numero:
15	????	????	Digite o segundo numero:
15	3	????	

Exemplo 1 – Execução Passo-a-Passo

- Comando:

```
resultado = numero1 + numero2
```

<u>numero1</u>	<u>numero2</u>	<u>resultado</u>	<u>saída</u>
????	????	????	Digite o primeiro numero:
15	????	????	Digite o segundo numero:
15	3	18	

Exemplo 1 – Execução Passo-a-Passo

- Comando:

```
io.write("Resultado da soma é ", resultado)
```

<u>numero1</u>	<u>numero2</u>	<u>resultado</u>	<u>saída</u>
????	????	????	Digite o primeiro numero:
15	????	????	Digite o segundo numero:
15	3	18	Resultado da soma é 18

Programando em C - Exemplo

- **Comentários:**

```
-- Programa para converter temperatura de Celsius em Fahrenheit

local cels          -- armazena temperatura em oC
local fahr          -- armazena temperatura em oF

io.write("Digite a temperatura em Celsius: ")
cels = io.read()    -- captura valor fornecido via teclado

fahr = 1.8 * cels + 32  -- faz a conversão

-- exibe resultado na tela
io.write("Temperatura em Fahrenheit: ", fahr, "\n");
```

Exercícios

Lista de Exercícios 01 - Algoritmos e Variáveis

<http://www.inf.puc-rio.br/~abaffa/eng1000/>