

Além de jogar, eu faço jogs!

Uma parceria de sucesso entre o CPDI e o Instituto Cooperforte



Professora Juliana Oliveira



Pauta – Aula 2

O que vamos aprender hoje

01

Ínicio do módulo jogos: conhecendo Lua



Conhecendo o Sublime Text

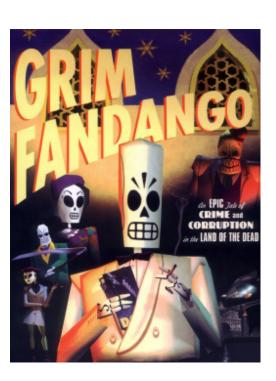


Conhecendo o Solar 2d/Corona SDK



Linguagem utilizada

- Projetada pela PUC-Rio em 1993.
- Primeiro jogo Grim Fandango (1997)



Jogos Famosos

Jogos que utilizam Lua em seu processo de programação

- Angry Birds
- Counter-StrikeOnline
- Roblox
- Street Fighter IV



• Linguagem de programação de extensão projetada para suportar facilidade na descrição de dados.

 Os identificadores podem ser qualquer sequência de letras ou dígitos, exceto números. Algumas palavras são reservadas e não podem ser usadas como váriaveis.

and	break	do	else	elseif	end	false
for	function	if	in	local	nil	not
or	repeat	return	then	true	until	while

- Há diferenciação entre maiúsculas e minúsculas.
- Linguagem tipada dinâmicamente.
- Todos os valores são de primeira classe.
- Mecanismo de estruturação por tabelas.

Variáveis

São lugares que armazenam valores. Existem três tipos de variáveis em Lua: **global**, **local** e **table fields(campos da tabela)** (propriedades). Qualquer variável não inicializada nil é por padrão.

Globais

Variáveis globais não precisam de declarações. Você simplesmente atribui um valor a um para criá-lo.

Local

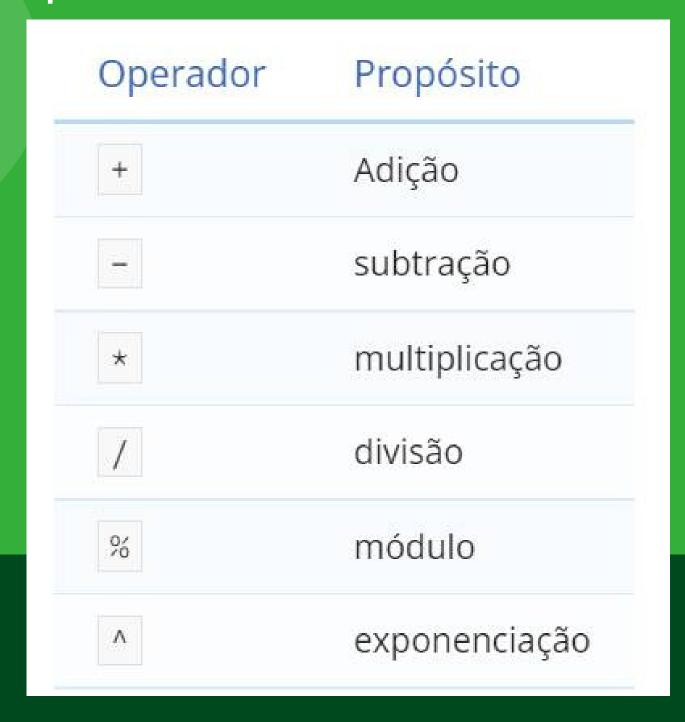
As variáveis locais são declaradas usando a instrução <u>local.</u>

Campos da tabela

Os campos da tabela são apenas os elementos da própria tabela. Você indexa na matriz para atribuir os valores a um campo.

Operadores aritméticos

Lua suporta os operadores aritméticos binários usuais:



Operadores relacionais

Os operadores relacionais em Lua são:

Operador	Propósito
==	igual a
~=	não igual a
<	Menor que
>	Maior que
<=	menos que ou igual a
>=	Melhor que ou igual a

Operadores lógicos

Os operadores lógicos em Lua são <u>and</u>, <u>or</u> e <u>not.</u> Todos consideram falsos <u>false</u> e <u>nil</u> e qualquer outra coisa como verdadeiro.

- and se o valor do primeiro argumento for <u>igual a false ou nil</u> ele retorna o mesmo, caso contrário retorna o segundo argumento.
- or se o primeiro argumento for <u>diferente de false ou nil</u> ele retorna o primeiro, caso contrário retorna o segundo.
- not sempre retorna false ou true.

Exemplos:

```
10 or 20
                --> 10
10 or error()
            --> 10
nil or "a"
                --> "a"
              --> nil
nil and 10
false and error() --> false
false and nil --> false
false or nil
           --> nil
10 and 20
            --> 20
```