Linguagem de Programação: Lua

1. Elementos Sintáticos

1.1 Comandos

Lua oferece um conjunto de comandos convencionais, semelhantes às linguagens Pascal e C, que incluem:

- Atribuições
- Estruturas de Controle
- Chamadas de Função
- Declarações de Variáveis

1.2 Trechos

Um trecho é definido como uma sequência de comandos executados sequencialmente, podendo ser seguido por um ponto-e-vírgula (;).

Definição Sintática:

trecho -> comando ";"

1.3 Blocos

Blocos são listas de comandos delimitadas que formam um único comando.

Definição Sintática:

bloco -> trecho
| comando -> do bloco end

Notas:

- Blocos controlam o **escopo de variáveis.**
- Permitem a **adição de comandos** como **return** ou **break** no código.

1.4 Atribuições

Lua suporta atribuições múltiplas, onde uma lista de variáveis recebe valores de uma lista de expressões, separadas por vírgulas.

Definição Sintática:

comando -> listavar "=" listaexp
listavar -> var {"," var}
listaexp -> exp {"," exp}

Antes da atribuição ser realizada, a lista de valores é *ajustada* para o comprimento da lista de variáveis. Se há mais valores do que o necessário, os valores em excesso são descartados.

1.5 Estruturas de Controle

As estruturas de controle **if**, **while** e **repeat** possuem o significado usual e a sintaxe familiar.

Definição Sintática:

comando -> **while** exp do bloco end
| **repeat** bloco until exp
| **if** exp then bloco {**elseif** exp then bloco} [else bloco] end

1.5.1 While

Inicia-se com **while**, seguido de uma expressão e um bloco delimitado por do e end.

1.5.2 Repeat

Inicia-se com a palavra reservada **repeat.**

1.5.3 If

Inicia-se com **if** e uma expressão, com um bloco delimitado por then e finaliza com end. Pode incluir elseif e else.

1.6 Comando For

O comando **for** possui duas variações: uma numérica e outra genérica.

1.6.1 Variação Numérica

Inicia com for, seguido por:

- Nome da variável
- Palavra reservada =
- Expressões inicial, final e, opcionalmente, passo.

Bloco delimitado por do e end.

Definição Sintática:

comando -> **for** nome `=´ exp `**,**´ exp [`**,**´ exp] **do** bloco **end**

1.6.2 Variação Genérica

Inicia com for, seguido por:

- Lista de nomes de variáveis separadas por vírgulas.
- Palavra reservada in.
- Lista de expressões.

Definição Sintática:

comando -> **for** listadenomes **in** listaexp **do** bloco **end**
| listadenomes ::= Nome {`**,**´ Nome}

1.7 Expressões

As expressões em Lua abrangem diversos padrões sintáticos.

Definição Sintática:

exp -> expprefixo
| **nil | false | true**
| Numero
| Cadeia
| funcao
| construtortabela
| "..."
| exp opbin exp
| opunaria exp
expprefixo -> var | chamadadefuncao | "(" exp ")"

Tipos de Expressões:

- **Primitivas**:nil,false,true.
- **Numéricas**: números.
- **Textuais**: cadeias de caracteres.
- **Funções**: padrão funcao.
- **Tabelas**: construtores conforme construtortabela.
- **Operações Unárias**: padrão opunaria exp.
- **Operações Binárias**: padrão exp opbin exp.
- **Prefixos**: variáveis, chamadas de função ou expressões entre parênteses.

1.7.1 – Operadores Aritméticos

Lua possui os operadores aritméticos padrão: +, -, *, /, %, ^ e o unário -.
Aceita números e strings numéricas.
Aceita qualquer expoente (ex: x^(-0.5) → inverso da raiz de x).
O módulo é definido por a % b == a - math.floor(a/b)*b, ou seja, resto da divisão arredondada para menos infinito.

1.7.2 – Operadores Relacionais

São: ==, ~=, <, >, <=, >=.

Retornam sempre **true** ou **false**.

A igualdade compara tipos antes dos valores — tipos diferentes sempre resultam em **false**.
Operadores de ordem (<, >, <=, >=) comparam números entre si e strings entre si; caso contrário, chamam metamétodos lt e le.

1.7.3 – Operadores Lógicos

São **and, or e not.**
Somente **false e nil** são considerados falsos.

- **not** retorna sempre true ou false.
- **and** retorna o primeiro operando falso ou o segundo se o primeiro for verdadeiro.
- **or** retorna o primeiro valor verdadeiro ou o segundo se o primeiro for falso.
  Usam avaliação de curto-circuito.

1.7.4 – Concatenação

O operador de concatenação é ...
Se ambos os operandos são números ou strings, são convertidos para string.

1.7.5 – O Operador de Comprimento

Denotado por #.
Para strings, retorna o **número de bytes.**
Para tabelas, o comprimento é o maior índice inteiro n tal que t[n] existe e t[n+1] é nil.

1.7.6 – Precedência

Ordem da menor para a maior:

or and < > <= >= ~= == .. + - * / % not # - (unário) ^

Pode-se usar parênteses para alterar a ordem.
Os operadores .. e ^ são associativos à direita; os demais, à esquerda.

1.7.7 – Construtores de Tabelas

Criam novas tabelas e inicializam campos.

Definição Sintática:

construtortabela -> `**{**´ [listadecampos] `**}**´
| listadecampos ::= campo {separadordecampos campo} [separadordecampos]
| campo ::= `**[**´ exp `**]**´ `**=**´ exp | Nome `**=**´ exp | exp
separadordecampos ::= `**,**´ | `**;**´

1.7.8 – Chamadas de Função

Em uma chamada de função, primeiro expprefixo e args são avaliados. Se o valor de expprefixo possui tipo function, então esta função é chamada com os argumentos fornecidos

Definição Sintática:

chamadadefuncao -> expprefixo args

Pode ser usada para chamar "métodos". Uma chamada v:nome(args) é um açúcar sintático
para v.nome(v,args), com a diferença de que v é avaliado somente uma vez.

Argumentos possuem a seguinte sintaxe:

args -> `**(**´ [listaexp] `**)**´
| construtordetabela | Cadeia

1.7.9 – Definições de Funções

Uma definição de função é uma expressão executável, cujo valor tem tipo *function*.

Definição Sintática:

funcao -> **function** corpodafuncao
|`**(**´ [listapar] `**)**´ bloco **end**

Uma definição de função é uma expressão executável, cujo valor tem tipo function. Quando Lua pré-compila um trecho, todos os corpos das funções do trecho são pré-compilados também. Então, sempre que Lua executa a definição de uma função, a função é instanciada (ou fechada). Esta instância da função (ou fecho) é o valor final da expressão. Instâncias diferentes da mesma função podem se referir a diferentes variáveis locais externas e podem ter diferentes tabelas de ambiente.

Referências

- **Manual de Referência de Lua 5.1**. Disponível em: <https://www.lua.org/manual/5.1/pt/manual.html#2.4>. Acesso em: 19 jan. 2026.