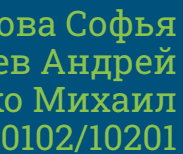




Lua

3. Выражения



Корпусова Софья
Афанасьев Андрей
Лутченко Михаил
5030102/10201

Введение

- Получение значений
- Включают :
 - Числовые константы
 - Строковые литералы
 - Переменные
 - Унарные/бинарные операции
 - Вызовы функций

Арифметические операции



Арифметические операции

- Работают с вещественными числами

pres_3.lua

```
1 local x = 4
2 print(x ^ 0.5)
3 print(x ^ (-1/3))
4 print(4 ^ 2)
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

sofakorpusova@Lenovo:~/lua_scripts\$ lua pres_3.lua

2.0

0.62996052494744

16.0

Операция %

- Работает по правилу $a \% b == a - (\text{math.floor}(a/b)) * b$

```
6  print(5 % 2)
7  print(5 % -2)
8
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

● sofakorpusova@Lenovo:~/lua_scripts\$ lua pres_3.lua

1

-1

Операция %

```
9   local x = math.pi
10  print(x % 1)
11  print(x - x % 1)
12  print(x - x % 0.01)
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

- **sofakorpusova@Lenovo:~/lua_scripts\$** lua pres_3.lua
0.14159265358979
3.0
3.14
- **sofakorpusova@Lenovo:~/lua_scripts\$** █

Операции сравнения

< > <= >= == ~=



Операции сравнения

- Возвращают **true** или **false**
- `==` и `~=` применяются к любым двум значениям
- Разные типы - значения не равны
- `nil` равно только самому себе
- Таблицы и `userdata` сравниваются по ссылке

Операции сравнения

- Таблицы и userdata сравниваются по ссылке

```
14  a = {}; a.x = 1; a.y = 0
15  b = {}; b.x = 1; b.y = 0
16  c = a
17  print(a == b)
18  print(a == c)
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

```
● sofakorpusova@Lenovo:~/lua_scripts$ lua pres_3.lua
false
true
```

Операции сравнения

- Операции порядка применимы только к двум числам или к двум строкам
- Строки сравниваются в алфавитном порядке

```
20  print(2 < 3)
21  print ("abc" > "abd")
--
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

```
● sofakorpusova@Lenovo:~/lua_scripts$ lua pres_3.lua
true
false
```

Операции сравнения

- При сравнении неявное приведение типов не работает
- Lua выбрасывает ошибку при сравнении ($>/<$) строк и чисел

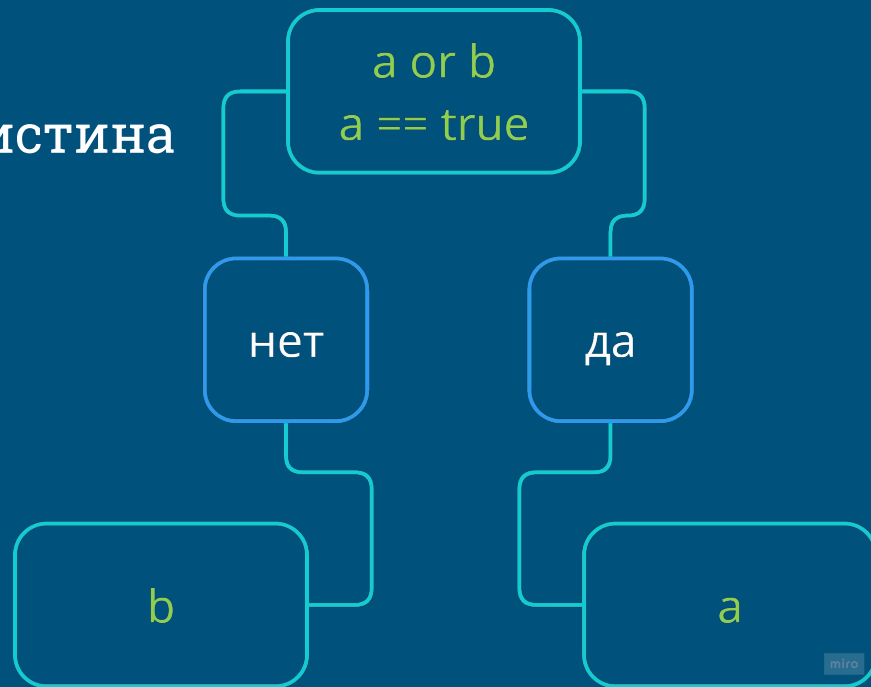
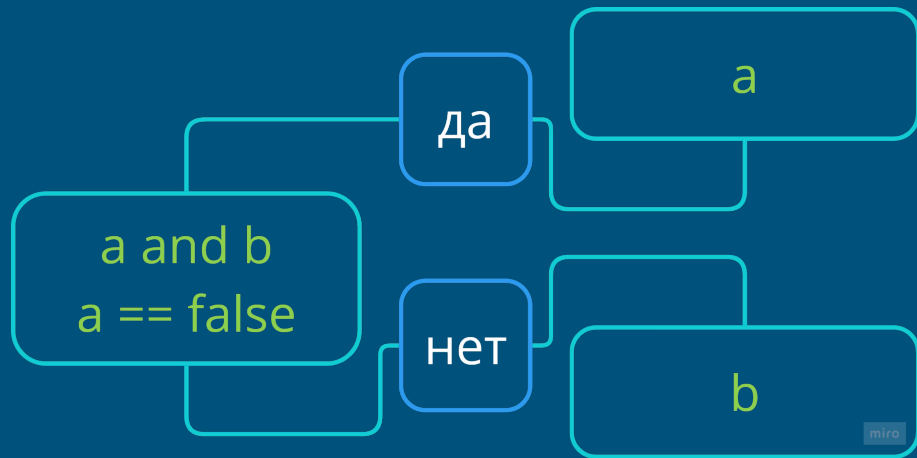
```
23  print("0" == 0)
24  print(2 < 15)
25  print("2" < "15")
26  print(2 < "15")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
⊗ sofakorpusova@Lenovo:~/lua_scripts$ lua pres_3.lua
false
true
false
lua: pres_3.lua:26: attempt to compare number with string
stack traceback:
      pres_3.lua:26: in main chunk
      [C]: in ?
```

Логические операции

- **and**, **or** и **not**
- **false**, **nil** - ложь, остальное истина



Логические операции

- **and** и **or** используют сокращённое вычисление
- **and** и **or** могут возвращать любые значения

```
28  print(4 and 5)
29  print(nil and 13)
30  print(false and 13)
31  print(4 or 5)
32  print(false or 5)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
• sofakorpusova@Lenovo:~/lua_scripts$ lua pres_3.lua
5
nil
false
4
5
```

Логические операции

- `x = x or v` эквивалентно

`if not x then x = v end` (`x` не ложно)

- `a and b or c` эквивалентно

`a : b ? c` в языке C (`b` не ложно)

```
34  local x = 7
35  local y = 4
36  max = (x > y) and x or y
37  print(max)
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

```
● sofakorpusova@Lenovo:~/lua_scripts$ lua pres_3.lua
```

7

Логические операции

- **not** всегда возвращает булево значение

```
39  print(not nil)
40  print(not false)
41  print(not 0)
42  print(not not 1)
43  print(not not nil)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
• sofakorpusova@Lenovo:~/lua_scripts$ lua pres_3.lua
true
true
false
true
false
```

Конкатенация

- ..
- Неявное приведение типов
- Всегда создаёт новую строку не изменяя свои операнды

```
45  print("Hello, " .. "World!")
46  print(0 .. 1)
47  print(type(0 .. 1))
48  local a = "Hello, "
49  print(a .. "World!")
50  print(a)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
• sofakorpusova@Lenovo:~/lua_scripts$ lua pres_3.lua
Hello, World!
01
string
Hello, World!
Hello,
```


Операция длины

- Для строк - количество байт
- Для таблиц - длина представленной последовательности
- Идиомы

`print(a[#a])` - печать последнего значения
последовательности

`a[#a] = nil` - удаление последнего значения

`a[#a + 1] = v` - добавление `v` в конец

Операция длины

```
65  local a = "abcdef"
66  print(#a)
67  print(#{'a', 'b', 'c', 88})
68  print(#{"a" = 1, ["b"] = 2})           --!
69  print(#{[1] = 1, [2] = 2, [3] = "abs"})
70  print(#{[1] = 1, [2] = nil, [3] = 1, [4] = 1}) --!
71  print(#{[1] = 1, [2] = 2, [3] = "abs", [4] = nil, [5] = nil})
--
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

● sofakorpusova@Lenovo:~/lua_scripts\$ lua pres_3.lua

6
4
0
3
1
3

Приоритеты операций

1. ^
2. not # -(унарный)
3. * / %
4. + -
5. ..
6. < > <= >= ~= ==
7. and
8. or

Конструкторы таблиц

- Создают и

инициализируют таблицы



```
73  days = {"Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat"}
74  print(days[1])
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
• sofakorpusova@Lenovo:~/lua_scripts$ lua pres_3.lua
Sun
```

Конструкторы таблиц

- $a = \{x = 10, y = 20\}$ эквивалентно

$a = \{\}; a.x = 10; a.y = 20$

- Всегда можно добавлять и удалять поля
- Можно смешивать стили инициализации

Конструкторы таблиц

```
76  polyline = {  
77      color = "blue",  
78      thickness = 2,  
79      npoints = 4,  
80      {x = 0, y = 0}, -- polyline[1]  
81      {x = -10, y = 0}, -- polyline[2]  
82      {x = -10, y = 1}, -- polyline[3]  
83      {x = 0, y = 1}, -- polyline[4]  
84  }  
85  print(polyline.color)  
86  print(polyline[2].x)  
--
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

● **sofakorpusova@Lenovo:~/lua_scripts\$** lua pres_3.lua

blue

-10

Конструкторы таблиц

```
88  opnames = {["+"] = "add", ["-"] = "sub",
89           |      |      ["*"] = "mul", ["/"] = "div"}
90
91  local i = 20
92  local s = "-"
93  a = {[i+0] = s, [i+1] = s..s, [i+2] = s..s..s}
94
95  print(opnames[s])
96  print(a[22])
---
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

● **sofakorpusova@Lenovo:~/lua_scripts\$** lua pres_3.lua

sub

В следующей презентации

- Операторы присваивания
- Управляющие конструкции
 - if then else
 - while
 - repeat
 - Числовой for
 - Общий for
- break, return и goto

**he was forced to use
lua**



Литература

- Операции в Lua
- Программирование на языке Lua - Роберту Иерузалимски
- Программирование на языках Lua и C