ИСТОРИЯ LUA И ПРЕДНАЗНАЧЕНИЕ

Язык Lua был разработан в Католическом университете Рио-де-Жанейро на базе языка C, первая версия появилась в 1993 году. Этот язык относится к виду скриптовых языков наравне с JavaScript и Python, то есть код такого языка выполняется покомандно.

Язык Lua, как и Python, популярен благодаря простой и краткой записи кода. В первую очередь Lua предназначен для начинающих программистов или пользователей, которых нельзя назвать профессиональными программистами.

Lua — язык с динамической типизацией. Динамическая типизация — это тип работы с переменными. При динамической типизации тип переменной определяется уже в момент выполнения программы. При статической это происходит заранее.

Языки с динамической типизацией: Lua, Python, JavaScript, Perl, PHP… Языки со статической типизацией: C, C++, C#, Java, Delphi. Lua стал широко известен как язык программирования уровней и расширений в компьютерных играх.

ТИПЫ ДАННЫХ И ОПЕРАЦИИ НАД ТИПАМИ ДАННЫХ

ПЕРЕМЕННЫЕ

Первым делом нужно рассмотреть основы любого языка — это типы данных, с которыми он работает.

- **Числа:** целые, числа с плавающей точкой, экспоненциальная форма записи числа.
- **Строки:** текстовые символы.
- **Логический:** имеет два логических значения — true и false.
- **Таблицы:** массивы, списки, словари.
- **Функции:** подпрограмма.
- **Nil:** пустое значение (что-то, что не имеет значения).

Переменные — именованная ячейка памяти. Разные значения разных типов данных мы можем присвоить ячейке памяти с именем и в дальнейшем вызывать ее.

РИС. 2.1. ДВА ТИПА ПЕРЕМЕННЫХ

Глобальная переменная может быть использована в любой части кода программы.

Локальная переменная может использоваться только с начала появления в коде программы.

Если нужно указать, что переменная локальная, то пропиши перед ней оператор `local`.

Какие могут быть переменные?

Переменные могут быть записаны так:

- `a, b, c, d, e, f;`
- `A, B, C, D, E, F;`
- `a1, a2, a3, a4, a5... x1, x2, x3;`
- `Name, long, age, La_Grange.`

Рассмотрим на примерах в Roblox Studio эти типы данных и переменные. Для этого запусти пустое окно локации Baseplate. В окне Explorer напротив слова Workspace нажми на значок + и создай Script.

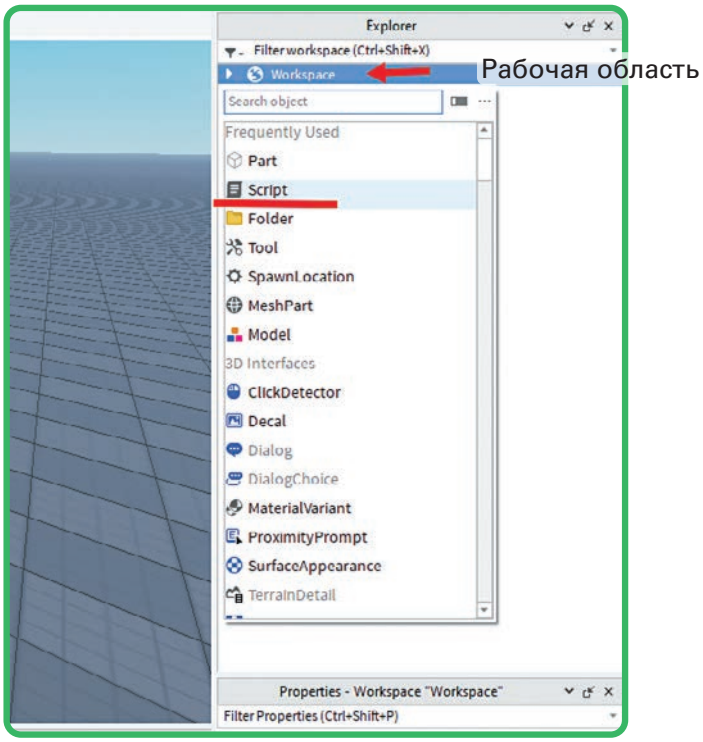

РИС. 2.2. СОЗДАНИЕ СКРИПТА

Script будет содержать код нашей программы, который при запуске игры будет выполняться. Результат выполнения программы в первую очередь будет отображаться в окне Output. После создания скрипта откроется окно Script.

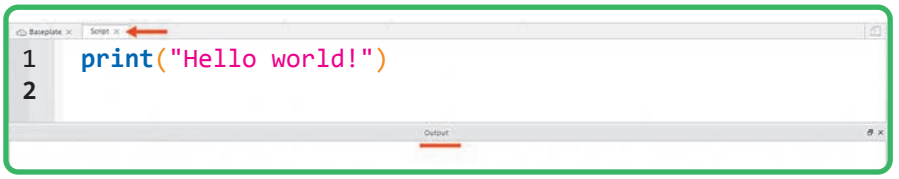

РИС. 2.3. ОКНО SCRIPT

По умолчанию здесь будет функция `print()`. Ее задача — выводить значения, указанные в скобках. Для примера указана строка `"Hello world!"`. Как можно заметить, строка записана в кавычках.

Запусти игру, нажав Play, и ты увидишь, что после загрузки в окне Output появится `Hello world!`, но уже без кавычек.

РИС. 2.4. ЗАПУСК ИГРЫ СО СКРИПТОМ

Выйди из игры, нажав Stop (красный квадрат), и перейди обратно в скрипт.

Рассмотри различные варианты чисел, которые можно использовать в Lua. Для этого нужно создать несколько переменных и присвоить им числовые значения, а затем вывести их в игре.

```lua
a=2
b=0.5
c=2.234564
d=-5
f=34.55e+6

print(a)
print(b)
print(c)
print(d)
print(f)
```

РИС. 2.5. ЧИСЛОВОЙ ТИП ДАННЫХ

Обрати внимание, что в записи дробных чисел целую часть отделяют от дробной с помощью точки. Пробежимся по числам. В переменную `a` записано целое число, в `b` — дробь с точностью до десятых, в `c` — дробь с точностью до шестого знака, в `d` — целое отрицательное число, в `f` — экспоненциальное число.

С какими-то типами чисел ты, наверное, уже знаком, а с какими-то только предстоит познакомиться на уроках математики. Запустим наш пример в Roblox Studio и посмотрим результат.

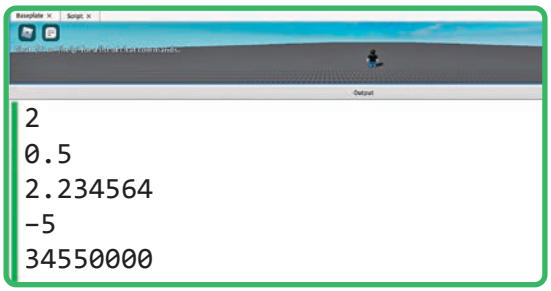

РИС. 2.6. РЕЗУЛЬТАТ РАБОТЫ СКРИПТА

На фото видно, что числа выведены так же, как они были записаны, за исключением последнего варианта. В этом варианте программа преобразовала эту запись в число.

Теперь рассмотрим, что такое строки.

Самые внимательные читатели, скорее всего, заметили, что строки — это любые символы, заключенные в кавычки. Рассмотрим несколько примеров строковых значений. Для этого очистим предыдущий скрипт и перезапишем его в таком виде.

```lua
a= "Name"
b="Вася"
c="356"
d="Lua и Roblox"
f="100 игр Roblox"

print(a)
print(b)
print(c)
print(d)
print(f)
```

РИС. 2.7. СТРОКИ

Для разнообразия используем символы кириллицы, английского алфавита и числа. Запустим игру и посмотрим результат.

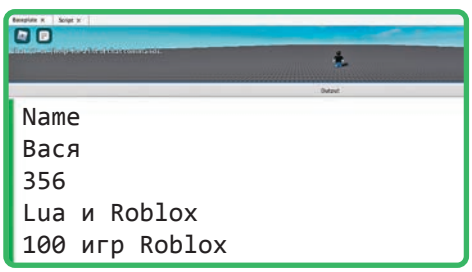

РИС. 2.8. ВЫВОД СТРОК

Ошибок не возникло. Приведем для наглядности пример с самой распространенной ошибкой.

```lua
a=Вася

print(a)

```

```
12:39:29.952 - Baseplate auto-recovery file was created
12:39:30.442 - Workspace.Script:1: Unexpected Unicode
character: U+412. Did you mean 'B'?
```

РИС. 2.9. ОШИБКА В КОДЕ

Здесь, как ты видишь, забыли кавычки. Заметь, что текст об ошибке содержит информацию о скрипте, в котором произошла ошибка, и строку, на которой она произошла.

Рассмотрим теперь логический тип данных.

Логический тип данных имеет два значения — `true` и `false`.

`true` = истина = 1.

`false` = ложь = 0.

Для логического типа данных существует таблица истинности, которая строится на логических операторах: `not`, `and`, `or`.

Эти операторы называются так: «отрицание», «конъюнкция» и «дизъюнкция».

A	B	A and B	A or B	not A
1	1	1	1	0
0	1	0	1	1
1	0	0	1	0
0	0	0	0	1

Ниже представлен пример логической операции из таблицы истинности.

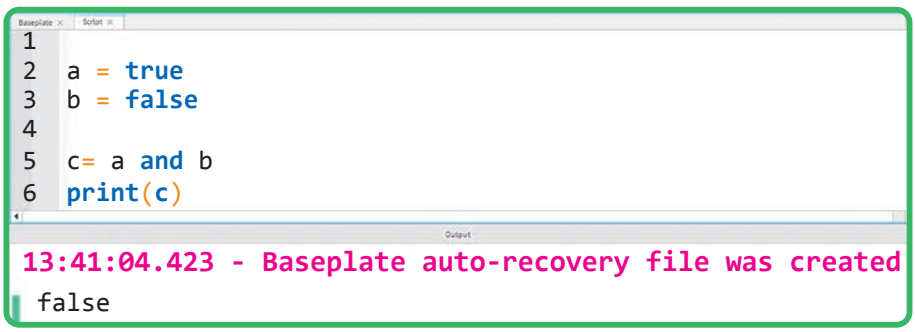

РИС. 2.10. ПРИМЕР ЛОГИЧЕСКОЙ ОПЕРАЦИИ

Как видно из примера, переменная `a` присвоила значение `true` (истина), а переменная `b` — `false` (ложь). При конъюнкции (`a and b`) согласно таблице истинности мы получим `false`. Такой же результат нам дает программа в Roblox.

Логические операции очень важны при создании игр. Любые действия персонажей или объектов постоянно проверяются на истинность или ложность.

Теперь, когда ты познакомился с основными типами данных, посмотрим, как переводить один тип в другой. Речь пойдет о переходе из числа в строку и обратно.

Есть две функции: `tostring()` и `tonumber()`. Первая преобразует числовой тип данных в строковый, а вторая — строковый в числовой. Но есть один нюанс: перевести строковый тип в числовой можно, если строкой является число. Чтобы совсем не запутаться, давай перейдем к примеру.

Возьмем переменную a и присвоим ей значение 10, а затем с помощью функции `tostring()` переведем в строковый тип данных. Чтобы удостовериться в том, что тип данных изменен, нужно получить информацию о типе данных. Это делается с помощью функции `type()`.

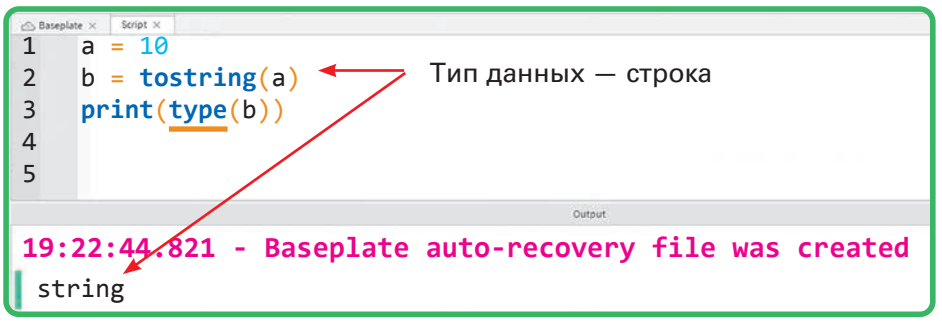

РИС. 2.11. ОПЕРАЦИЯ НАД ТИПОМ ДАННЫХ

Рассмотрим обратную задачу.

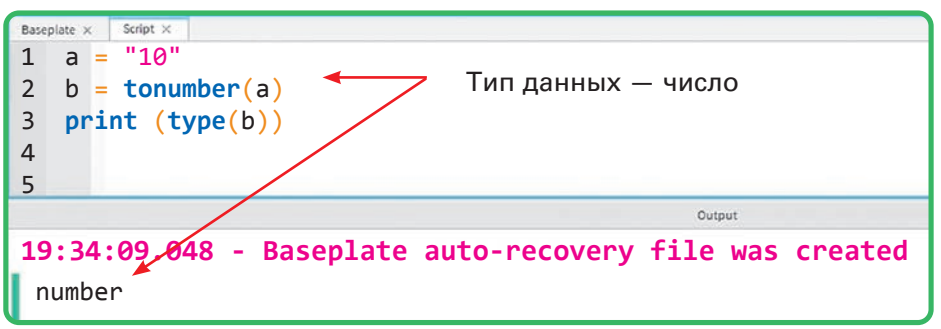

РИС. 2.12. ОПЕРАЦИЯ НАД ТИПОМ ДАННЫХ

Слово `number` указывает, что строка `"10"` была преобразована в число `10`.

В следующей главе эти две операции еще раз проверятся на достоверность. Мы рассмотрели основные типы данных и можем перейти к математическим операциям, которые не менее важны, чем сами данные.