1.Lua将值false和nil视为“假”，而将除此之外的其他值视为“真”，将数字零和空字符串也都视为“真”。

2.Lua的字符串是不可变的值，不能像在C语言中那样直接修改字符串的某个字符，而是应该根据修改要求来创建一个新的字符串。

a="one string"

b=string.gsub(a,"one","another");

3.字面字符串需要以一对匹配的单引号或双引号来界定：

4.在一个字符串上应用算术操作时，Lua会尝试将这个字符串转换成一个数字。

5.在 lua中期望一个字符串但却得到一个数字时，它也会将数字转换成字符串：

print（10 .. 20） -->1020

6.但最好不要依赖lua的自动的强制转换，例如比较运算10==“10”总为false。如果需要显式地将一个字符串转换成数字，可以使用函数tonumber。当这个字符串的内容不能表示一个正确的数字时，tonumber将返回nil。

line=io.read() --读取一行

n=tonumber(line) --尝试将它转换为一个数字

if n==nil then

error(line . . "is not a valid number")

else

print(n\*2)

end

7.要将一个数字转换成字符串，可以调用函数tostring，或者将该数字与一个空字符串相连接：

print（tostring(10) =="10") -->true

print （10 .. "" =="10"）-->true

8.在lua5.1中，可以在字符串前放置操作符“#”来获得该字符串的长度：

a="hello"

print(#a) -->5

print(#"good\0bye") -->8

9.table类型实现了“关联数组”。“关联数组”是一种具有特殊索引方式的数组。不仅可以通过整数来索引它，还可以使用字符串或其他类型的值（除了nil）来索引它。此外，table没有固定的大小，可以动态地添加任意数量的元素到一个table中.

a={}

k="x"

a[k]=10;

print(a["x"]) -->10

print(a.x) -->10

10.虽然可以用任何值作为一个table的索引，也可以用任何数字作为数组索引的起始值。就Lua的习惯而言，数组通常以1作为索引的起始值。并且还有不少机制依赖于这个惯例。

a={}

for i=1,10 do

a[i]=io.read()

end

11.在lua5.1中，长度操作符“#”用于返回一个数组或线性表的最后一个索引值（或为其大小）。例如，可以像下面这样打印上例中读取到的每行内容：

--打印所有的行

for i=1 #a do

print(a[i])

end

12.获得table的长度

a={}

a[1000]=1

print(table.maxn(a)) -->1000

13.对于table，userdata和函数，lua是作引用比较的。也就是说，只有当它们引用同一个对象时，才认为它们相等。例如，一下代码：

a={};a.x=1;a.y=0

b={};b.x=1;b.y=0

c=a

其结果是a==c,但a~=b.

14.and和or都使用“短路求值”，也就是说，它们只会在需要时采取评估第二个操作数。

有一种常用的lua习惯写法“x=x or v”，它等价于：if not x then x=v end

它可用在没有设置x的时候（即对x的求值结果为假时），将其设为一个默认值v。

另外，还有一种习惯写法是“（a and b）or c”，这类似于C语言中的表达式a？b：c，但前提是b不为假。例如，为了选出数字x和y中的较大者，可以使用一下语句：

ma=（x>y）and x or y

操作符not永远只返回true或false

print（not nil） -->true

print （not false） -->true

print （not 0） -->false

print (not not nil) -->false

15.table 构造式

days ={"Sunday","Monday","Tuesday","Wednesday"}

w={x=0,y=0,label="console"}

x={math.sin(0),math.sin(1),math.sin(2)}

w[1]="another field"

x.f=w

print(w["x"])

print(w[1]) -->another field

print(x.f[1]) -->another field

w.x=nil --删除字段"x"

16.赋值。

Lua允许"多重赋值"，也就是一下子将多个值赋予多个变量。每个值或每个变量之间以逗号分隔。例如：

a,b=10,2\*x

赋值后，变量a为10，b为2\*x。

在多重赋值中，Lua先对等号右边的所有元素求值，然后才执行赋值。这样便可以用一句多重赋值来交互两个变量了，如下所示：

x,y=y,x --交换x与y

a[i],a[j]=a[j],a[i] --交换a[i]与a[j]

多重赋值一般用在一个函数允许返回多个值，在这种情况一次函数返回的结果就需要有多个变量来接收。例如，a,b=f(),函数f将返回两个值，a接收了第一个，b接收了第二个。