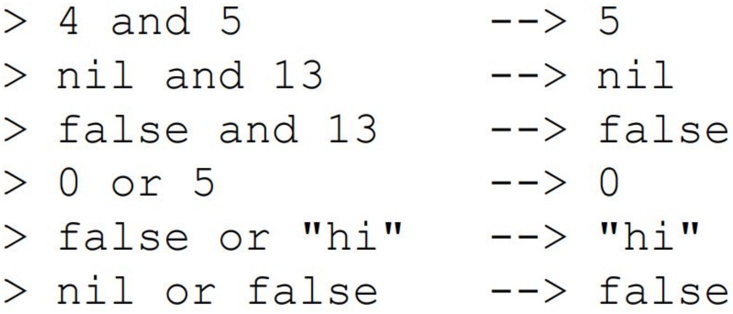
Booleans and Logical Operators

Booleans and Logical Operators

- The Boolean type has two values, `false` and `true` , which represent the traditional Boolean values.

Booleans and Logical Operators

- Lua supports a conventional set of logical operators: **`and`**, **`or`**, and **`not`**
  - The result of the **`and`** operator is its first operand if that operand is false; otherwise, the result is its second operand.
  - The result of the **`or`** operator is its first operand if it is not false; otherwise, the result is its second operand.



```
> 4 and 5          --> 5
> nil and 13       --> nil
> false and 13     --> false
> 0 or 5           --> 0
> false or "hi"    --> "hi"
> nil or false     --> false
```

Booleans and Logical Operators

- A useful Lua idiom is `x = x or v`, which is equivalent to

```
if not x then x = v end
```

Booleans and Logical Operators

- A useful Lua idiom is `x = x or v`, which is equivalent to

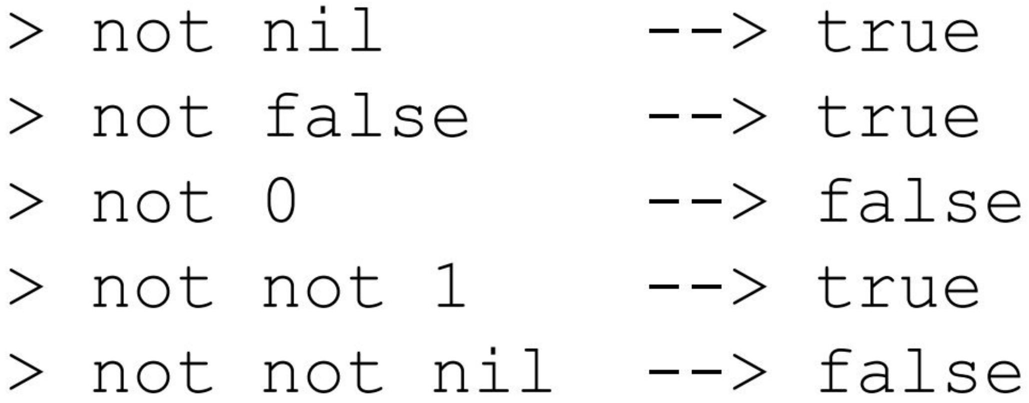
  ```
  if not x then x = v end
  ```

- Another useful idiom is `(a and b or c)`. It is equivalent to the C expression `a ? b : c`

  For instance, we can select the maximum of two numbers x and y with the expression `(x > y) and x or y`. When x > y, the first expression of the **and** is true, so the **and** results in its second operand `(x)`, which is always true (because it is a number), and then the **or** expression results in the value of its first operand, `x`. When x > y is false, the **and** expression is false and so the **or** results in its second operand, y.

Booleans and Logical Operators

- The **not** operator always gives a Boolean value:



```
> not nil        --> true
> not false      --> true
> not 0          --> false
> not not 1      --> true
> not not nil    --> false
```