Cheatsheets / **Learn Lua**

Functions

lua function arguments

When a function is called, it can contain a comma-separated list of values between parenthesis, these are **arguments**. They are assigned to parameters based on the order in the function declaration.

```lua
function printNumbers(x, y)
  print("Your first number is: " .. x)
  print("Your second number is: " .. y)
end

-- 3 and 25 are our arguments
printNumbers(3, 25)
-- Prints:
-- Your first number is 3
-- Your second number is 25
```

lua function declarations

Functions are declared using their name, which can be used to call the function later. Function declarations are built from:

- The `function` keyword
- The function’s name
- An optional list of comma-separated parameters, enclosed by `()`
- The `end` keyword as the last line of the function

```lua
function printFavoriteColor(color)
  print("Your favorite color is " ..
color)
end
```

Functions

A function is a set of statements that are executed together when called using the function’s name.

```lua
function getTallestMountain()
  print("The tallest mountain is Mount
Everest")
end
```

lua function return

A function can **return**, or pass back, values using the `return` keyword. Return ends function execution and returns the specified value to where the function was called.

```
function double(x)
  return 2 * x
end

doubledX = double(2)
print(doubledX) -- Prints: 4
```

lua function documentation

Lua's built-in functions are all documented on Lua's official website. For additional examples of these functions, the user-maintained documentation is also useful.

lua function parameters

When a function is declared it can contain **parameters** as inputs. These act as variables inside the function. These parameters are only assigned a value when the function is called.

```
function printNumber(number)
  print("Your number is: " .. number)
end
```

lua function calls

Functions can be called after they have been declared using the function's name followed by parenthesis. When called, all the code inside the function's declaration is run.

```
function printSquare(num)
   print(num * num)
end

printSquare(4) --Prints: 16
printSquare(2) --Prints: 4
printSquare(16) --Prints: 256
```

lua functions nil parameters

If a function is called without passing in enough arguments, the `nil` value is assigned to the remaining parameters.

```lua
function printAge(age)
  print("You are " .. age .. " years old")
end

printAge() -- Prints: You are nil years old
```

lua function built-ins

Lua has several built-in functions. Most are automatically included and can be called with no additional code. Some built-in functions include the `print()` and `type()` functions.

Print Share ▼