

# Manipulación de tablas

Lua implementa funciones que permiten la manipulación de las tablas facilitando la realización de operaciones con ellas. Todas ellas están definidas en el módulo `table` de la librería estándar de Lua. Vamos a ver a continuación más en detalle estas funciones:

- [table.concat\(\)](#)
- [table.insert\(\)](#)
- [table.remove\(\)](#)
- [table.move\(\)](#)
- [table.sort\(\)](#)
- [table.unpack\(\)](#)
- [table.pack\(\)](#)

---

## Función `table.concat`

La función `concat` permite la concatenación de elementos de una tabla. Todos los elementos de la tabla deben ser cadenas o números. Veamos la signatura del método:

```
table.concat(lista, [, sep [, i [, j]])
```

Detalle de los argumentos:

**lista.** Lista que se quiere concatenar, todos sus elementos deben ser cadenas o números.

**sep.** Separador, opcional. Es el caracter que se usará como separador. Si no se especifica se usa por defecto la cadena vacía.

**i.** Índice de inicio, opcional. Permite definir el índice del primer elemento que se concatenará. Por defecto es 1.

**j.** Índice de fin, opcional. Permite definir el índice del último elemento que se concatenará. Por defecto es `#lista`.

Veamos ahora un ejemplo del uso de la función `concat`:

```
paises = {"Suiza", "Espana", "Colombia", "Italia", "Francia"}

-- Concatenación de la lista con los parámetros por defecto
print ("Concatenacion con parametros por defecto: ", table.concat(paises))
```

```
-- Concatenación con un caracter separador
print ("Concatenacion con un separador: ", table.concat(paises, "; "))

-- Concatenación de una sublista
print ("Concatenacion de una sublista: ", table.concat(paises, "; ", 2, 4))
```

Al ejecutar el código anterior, obtenemos el siguiente resultado:

```
Concatenacion con parametros por defecto:      SuizaEspanaColombiaItaliaFrancia
Concatenacion con un separador:              Suiza; Espana; Colombia; Italia; Francia
Concatenacion de una sublista:  Espana; Colombia; Italia
```

En la primera línea, los elementos de la lista se han concatenado sin separación, ya que por defecto, se usa una cadena vacía. En la segunda, se ha definido un separador. En la tercera se han definido, además del separador, el elemento de inicio y fin para la concatenación.

## Función `table.insert`

Permite insertar un elemento en la tabla en la posición especificada, desplazando los demás elementos si es necesario. La signatura de este método es como sigue:

```
table.insert(lista, [ pos,] valor)
```

Los argumentos son:

**lista.** Lista a la cual queremos insertarle un elemento.

**pos.** Posición, opcional. Define la posición donde queremos insertar el elemento dentro de la lista. Por defecto los elementos se insertan al final de la lista.

**valor.** Valor a insertar en la lista.

Veamos a continuación un ejemplo de uso de la función `insert`:

```
paises = {"Suiza", "Espana", "Colombia"}

-- Inserción de un elemento al final de la lista
table.insert (paises, "Cuba")
print ("Se agrego un elemento al final de la lista:", table.concat (paises, ", "))

-- Inserción de un elemento en el índice 3 de la lista
table.insert (paises, 3, "Venezuela")
print ("Se agrego un elemento en el indice 3 de la lista:", table.concat (paises, ", "))
```

Al ejecutar el código anterior obtenemos el siguiente resultado:

```
Se agrego un elemento al final de la lista:      Suiza, Espana, Colombia, Cuba
Se agrego un elemento en el indice 3 de la lista:  Suiza, Espana, Venezuela, Colombia,
Cuba
```

En la primera línea, insertamos el elemento "Cuba" al final de la lista. En la segunda línea, insertamos el elemento "Venezuela" en la posición 3, como podemos ver, elemento que estaba en la posición 3, "Colombia", ha sido desplazado a la posición 4 y a su vez el elemento en la posición 4 ha sido desplazado a la posición 5.

## Función `table.remove`

La función `table.remove` permite eliminar un elemento de la lista. Veamos la signatura del método:

```
table.remove (lista [, pos])
```

El detalle de los argumentos es:

**lista.** Lista de la cual eliminaremos el elemento.

**pos.** Posición, opcional. Posición que ocupa el elemento que queremos eliminar. Si no se especifica, se eliminará el último elemento de la lista.

**Retorno:** esta función retorna el elemento eliminado.

Veamos un ejemplo del uso de la función `table.remove`:

```
paises = {"Suiza", "Espana", "Colombia", "Italia", "Francia"}

-- Eliminamos el cuarto elemento de la lista
pais = table.remove (paises, 4)
print ("El pais eliminado de la lista es:", pais)
print ("Contenido de la lista:", table.concat (paises, ", "))

-- Vaciamos la lista
contenido = table.remove(paises)
print ("Paises eliminados de la lista:", contenido)
print ("Contenido de la lista:", table.concat(paises, ", "))
```

Al ejecutar el código anterior obtenemos el siguiente resultado:

```
El pais eliminado de la lista es:      Italia
Contenido de la lista:  Suiza, Espana, Colombia, Francia
Paises eliminados de la lista:  Francia
Contenido de la lista:  Suiza, Espana, Colombia
```

En la primera línea se ha eliminado "Italia" cuya posición es 4 dentro de la lista. En la segunda línea vemos los elementos restantes en la lista después de eliminar uno. En la tercera línea, no hemos especificado ningún elemento, por lo que se eliminará el último elemento de la lista, que es "Francia".

## Función `table.move`

La función `table.move` nos permite copiar elementos de una tabla a otra. Podemos copiar los elementos en una nueva tabla o bien copiarlos en la misma tabla. Esta función está disponible a partir de Lua 5.3. Esta es la signatura del método:

```
table.move (lista1, desde, hasta, insercion, [, lista2])
```

Veamos el detalle de los argumentos:

**lista1.** Lista desde donde copiaremos los elementos.

**desde.** Índice desde el cual comenzarán a copiarse elementos de la lista1.

**hasta.** Índice hasta el cual se copiarán elementos de la lista1.

**insercion.** Índice de la lista2 donde se insertarán los elementos copiados.

**lista2.** Opcional, lista donde se insertarán los elementos copiados. Si no se especifica una lista, el destino será lista1.

**Retorno:** devuelve lista2.

Vamos a ver un ejemplo de la función `table.move`:

```
paises = {"Suiza", "Espana", "Colombia", "Italia", "Francia"}
otros = {"Alemania"}

table.move (paises, 2, 3, 2, otros)

print ("Paises en tabla otros:", table.concat(otros, ", "))
print ("Paises en tabla paises:", table.concat(paises, ", "))
```

El resultado que obtendremos es el siguiente:

```
Países en tabla otros: Alemania, España, Colombia
```

```
Países en tabla países: Suiza, España, Colombia, Italia, Francia
```

Como puedes observar hemos copiado los elementos "España" y "Colombia" dentro de la nueva lista a continuación del elemento "Alemania", cuyo resultado es una lista con tres elementos. Por otro lado, vemos que la lista original conserva todos sus elementos.

## Función `table.sort`

La función `table.sort`, permite ordenar los elementos de una lista. La signatura de la función es como sigue:

```
table.sort (lista [, comparador])
```

Veamos el detalle de los argumentos:

**lista.** Lista que queremos ordenar.

**comparador.** Opcional. Función que acepta dos elementos de la lista y que devuelve `true` si el primer elemento debe ir delante del segundo elemento. Si no se especifica, se usará el operador `<` de Lua.

Veamos un ejemplo:

```
países = {"Suiza", "España", "Colombia", "Italia", "Francia"}

-- Ordenar los países por orden alfabético
table.sort (países)
print ("Países ordenados por orden alfabético:", table.concat(países, ", "))

-- Ordenar los países por orden alfabético inverso
table.sort (países, function (e1, e2) return e1 > e2 end)
print ("Países ordenados por orden alfabético inverso:", table.concat(países, ", "))

function reordena (elementoA, elementoB)
  if elementoA < elementoB then
    return true
  else
    return false
  end
end
```

```
-- Ordenar las cifras usando una función externa
cifras = {23, 56, 11, 9, 32, 60, 3}

table.sort (cifras, reordena)
print ("Cifras ordenadas por funcion:", table.concat(cifras, ", "))

function inversa (elementoA, elementoB)
    if elementoA > elementoB then
        return true
    else
        return false
    end
end

-- Ordenar las cifras usando una función externa inversa
table.sort (cifras, inversa)
print ("Cifras ordenadas por funcion inversa:", table.concat(cifras, ", "))
```

Al ejecutar el ejemplo anterior, obtendremos el siguiente resultado:

```
Países ordenados por orden alfabetico:  Colombia, Espana, Francia, Italia, Suiza
Países ordenados por orden alfabetico inverso:  Suiza, Italia, Francia, Espana, Colombia
Cifras ordenadas por funcion:    3, 9, 11, 23, 32, 56, 60
Cifras ordenadas por funcion inversa:    60, 56, 32, 23, 11, 9, 3
```

## Función table.unpack

La función `table.unpack()` devuelve los elementos de una lista. Su signatura es:

```
table.unpack (lista [, i [, j]])
```

Veamos el detalle de los argumentos:

**lista.** Lista de la cual se extraerán los elementos.

**i.** Opcional. Índice a partir del cual comenzar a extraer elementos. Su valor por defecto es 1.

**j.** Opcional. Índice del último elemento a extraer. Su valor por defecto es el último elemento de la lista (`#lista`)

**Retorna:** los elementos extraídos.

Veamos a continuación un ejemplo de la función `table.unpack`:

```
países = {"Suiza", "España", "Colombia", "Italia", "Francia"}

espana, colombia = table.unpack (países, 2, 3)

print ("Países extraídos:", espana, colombia)
```

Cuyo resultado es:

```
Países extraídos:      España Colombia
```

## Función table.pack

La función `table.pack` crea una nueva tabla a partir de un número arbitrario de elementos pasados como argumentos. Su signatura corresponde con:

```
table.pack (elm1 [, elm2 [, elm_n]])
```

**elm1, elm2 ... elm\_n.** Elementos que serán insertados en la nueva tabla.

**Retorna:** una nueva tabla con los elementos pasados.

Veamos un ejemplo de `table.pack`:

```
espana = "España"
colombia = "Colombia"
peru = "Peru"
argentina = "Argentina"
mexico = "Mexico"

países = table.pack (espana, colombia, peru, argentina, mexico)

print ("Lista de países:", table.concat(países, ", "))
```

Si ejecutamos el código anterior obtendremos el siguiente resultado:

```
Lista de países:      España, Colombia, Peru, Argentina, Mexico
```

---

Revisión #1

Creado 19 febrero 2024 07:17:31 por Guillermo

Actualizado 19 febrero 2024 14:16:51 por Guillermo