

JavaScript Funcional

Marco Antonio Agüero Hidalgo
Matías Arturo Adauy Castro
Kevin Cespedes Arancibia

rowsToColumns & columnsToRows



```
1 // Transpone el CSV (cambia filas por columnas o al revés).  
2 const rowsToColumns = flow(parseCSV, transpose, toCSV);  
3 const columnsToRows = flow(parseCSV, transpose, toCSV);
```

_.flow

Arguments: [funcs] The *functions to invoke*.

Returns: (Function) Returns the new *composite function*. $\Rightarrow f(g(x))$

```
function square(n) {  
  return n * n;  
}
```

```
var addSquare = _.flow([_.add, square]);  
addSquare(1, 2);  
// => 9
```

rowsToColumns & columnsToRows



```
1 // Transpone el CSV (cambia filas por columnas o al revés).  
2 const rowsToColumns = flow(parseCSV, transpose, toCSV);  
3 const columnsToRows = flow(parseCSV, transpose, toCSV);
```



```
1 const parseRow = flow(split(','), map(trim));  
2 // Convierte un CSV en una matriz (array de arrays).  
3 const parseCSV = flow(trim, split('\n'), map(parseRow));  
4 // Convierte una matriz (array de arrays) a un CSV.  
5 const toCSV = flow(map(join(',')), join('\n'));
```

every & _.zip



```
1 // Transposición de una matriz mediante zip
2 const transpose = (matrix) => {
3   const rowLength = matrix[0].length;
4   const hasSameLength = row => row.length === rowLength;
5   if (matrix.every(hasSameLength)) {
6     return _.zip(...matrix);
7   }
8   else {
9     throw new Error('Las filas o columnas no tienen la misma longitud');
10  }
11 };
```

```
_.zip(['a', 'b'], [1, 2], [true, false]);
// => [['a', 1, true], ['b', 2, false]]
```

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \quad A^T = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$$

swap



```
1 // Intercambia de columnas n y m en el CSV.
2 const swap = (csv, n, m) => flow(
3   parseCSV,
4   transpose,
5   cols => swapColumns(cols, n - 1, m - 1),
6   transpose,
7   toCSV
8 )(csv);
```