

# Exercise on String Parsing

Learn to parse strings using the trim and split methods of javascript.

Suppose the following CSV data string is given. Parse the data such that you create an array of lines. Each line is an array of strings containing each data point once.

```
const data = `
Title,Author,Publication Date,Publisher
ES6 in Practice,Zsolt Nagy,2017,Self-published
The Developer's Edge,Zsolt Nagy,2016,Self-published
Regex Quick Syntax Reference,Zsolt Nagy,2018,Apress
The Charismatic Coder,Zsolt Nagy,2018,Self-published
Deep Dive into Functional JavaScript,Zsolt Nagy,2017,PacktPub
Implementing and Testing Applications using Functional JavaScript,Zsolt Nagy,2017,PacktPub
Mastering Functional JavaScript Libraries,Zsolt Nagy,2017,PacktPub
Beginning Modern JavaScript Development with Microservices WebRTC and React,Zsolt Nagy,2017,
Beginning ASP.NET,Zsolt Nagy,2017,PacktPub
Become the CSS Hero of Your Office with CSS Architecture,Zsolt Nagy,2017,SitePoint
Setting Up and Kick Starting TypeScript,Zsolt Nagy,2017,SitePoint
`;
```

## Solution

First, note that there is a newline character before the first line and after the last line. We can get rid of these characters by trimming the string.

We will then split the trimmed string into lines and then split the lines into fields separated by commas. Splitting the string into lines can be done by referencing the newline character `\n`. Note that the 'data' constant has been prepended in a hidden code tab to the following solution to avoid repetition!

```
const result = data.trim().split('\n').map( row => row.split(',') );
console.log(result);
```

Note that there are no commas in the titles. In this specific case, it makes a lot more sense to use another separator character between the fields, as commas may occur in any title. If you encounter such a problem, use a separator that makes the most sense to you.

If you don't want to use a map function, you can also iterate:

```
const result = [];  
const rows = data.trim().split( '\n' );  
for ( let row of rows ) {  
    result.push( row.split( ',' ) );  
}  
  
console.log(result);
```

