

Forum:[Hack Forums](#)

Thread: [CSV to javascript array](#)

newt0n - November 9, 2019, 10:09 pm

I have a .CSV file that has 3 columns file that needs to be converted to 3 arrays(one for each column(obviously))

The .CSV is sever side and not client side, if that matters

MavenGradle - November 10, 2019, 5:03 am

CSV files are comma separated. It's literally in the name. Do note that sometimes the values between the separators are enclosed in quotation marks in order to prevent the separator from being broken. For more information, there is an entire standard in regards to CSV files that you have to take into account if you want to do this properly:

<https://tools.ietf.org/html/rfc4180>

Since you're using JavaScript do to this, you can very easily create a function that, according to the CSV standard, takes out all the values and pushes the values into an array:

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/push

I don't really know why you need 3 arrays for this? It makes no sense to me. You could very easily push the values into a single JSON object, much like this:

```
Code {  
  "0": [  
    "row0",  
    "row1"  
  ],  
  "1": [  
    "row0",  
    "row1"  
  ],  
  "2": [  
    "row0",  
    "row1"  
  ]  
}
```

newt0n - November 10, 2019, 7:19 am

(Nov 10, 2019 06:03 AM)MavenGradle Wrote: [\[CITE\]](#) CSV files are comma separated. It's literally in the name. Do note that sometimes the values between the separators are enclosed in quotation marks in order to prevent the separator from being broken. For more information, there is an entire standard in regards to CSV files that you have to take into account if you want to do this properly:

[RFC 4180 - Common Format and MIME Type for Comma-Separated Values \(CSV\) Files](#)

Since you're using JavaScript do to this, you can very easily create a function that, according to the CSV standard, takes out all the values and pushes the values into an array:

[Array.prototype.push\(\) - JavaScript | MDN](#)

I don't really know why you need 3 arrays for this? It makes no sense to me. You could very easily push the values into a single JSON object, much like this:

```
Code {  
  "0": [  
    "row0",  
    "row1"  
  ],  
  "1": [  
    "row0",  
    "row1"  
  ],  
  "2": [  
    "row0",  
    "row1"  
  ]  
}
```

alright, thank you. I needed the data to be printed out into a graph/chart, I was able to find a solution. Thanks anyways, much appreciated. ? ?

ModernJS - November 22, 2019, 4:24 am

(Nov 10, 2019 06:03 AM)MavenGradle Wrote: [\[CITE\]](#) CSV files are comma separated. It's literally in the name. Do note that sometimes the values between the separators are enclosed in quotation marks in order to prevent the separator from being broken. For more information, there is an entire standard in regards to CSV files that you have to take into account if you want to do this properly:

[RFC 4180 - Common Format and MIME Type for Comma-Separated Values \(CSV\) Files](#)

Since you're using JavaScript do to this, you can very easily create a function that, according to the CSV standard, takes out all the values and pushes the values into an array:

[Array.prototype.push\(\) - JavaScript | MDN](#)

I don't really know why you need 3 arrays for this? It makes no sense to me. You could very easily push the values into a single JSON object, much like this:

```
Code {  
  "0": [  
    "row0",  
    "row1"  
  ],  
  "1": [  
    "row0",  
    "row1"  
  ],  
  "2": [  
    "row0",  
    "row1"  
  ]  
}
```

Well we can also Map the items out into Array instead of pushing:

Maybe say like this?

PHP Code

```
<?php
export default {
  data() {
    return {
      articles: [],
      users: [], //ignore
      category: []
    };
  },
  created() {
    let uri = "/api/v1/articles"; //can also be a csv file

    axios.get(uri).then(response => {
      this.articles = response.data;

      console.log(this.articles);

      response.data.map(element => {
        console.log(element);
      });
    });
  }
};
```