

SPLIT in

JS

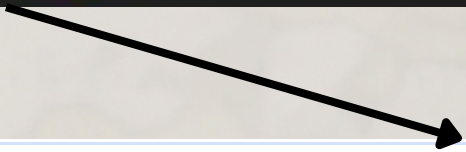


Sahith Bodla



We know that **Split** method splits a string into an array of substrings.

```
const brands = "audi,mercedes,tata";  
const cars = brands.split(",");  
console.log({ cars });
```



```
▼ {cars: Array(3)} i  
  ► cars: (3) ['audi', 'mercedes', 'tata']  
  ► [[Prototype]]: Object
```

But what if we have different separators other than comma?

```
const brands = "audi;mercedes,tata;toyota";
```



If you use a regular string inside **Split**,
result will be like

```
▼ {cars: Array(2)} ⓘ  
  ► cars: (2) ['audi;mercedes', 'tata;toyota']  
  ► [[Prototype]]: Object
```


which is not what we desire



The solution is to use **RegExp** inside the **Split** function to get the desired one!

like

```
const brands = "audi;mercedes,tata;toyota";  
const cars = brands.split(/[,;]/); // used RegExp here  
console.log({ cars });
```



```
▼ {cars: Array(4)} ⓘ  
  ► cars: (4) ['audi', 'mercedes', 'tata', 'toyota']  
  ► [[Prototype]]: Object
```

and all set!



This will be helpful in cases where we are handling data from a **CSV (Comma Seperated Values)**, but some fields may have commas within them, and you want to split the string into individual fields while correctly handling those commas within quotes.

```
"John Doe","123 Main St, Apt 3B","New York, NY",10001
```

CSV data



Sahith Bodla



```
const csv_string = '"John Doe","123 Main St, Apt 3B","New York, NY",10001';  
  
const fields = csv_string.split(/,(?=(?:[^\"]*"|"[^"]*"|'[^']*'|'[^\']*')*$)/);  
  
console.log({ fields });
```

[VM676:5](#)

```
▼ {fields: Array(4)} ⓘ  
  ► fields: (4) ['"John Doe"', '"123 Main St, Apt 3B"', '"New York, NY"', '10001']  
  ► [[Prototype]]: Object
```

Here `/,(?=(?:[^\"]*"|"[^"]*"|'[^']*'|'[^\']*')*$)/` is the
RegExp for split by commas, but
ignores commas within double quotes



LIKE & FOLLOW

for more



Sahith Bodla