

05

## [JavaScript] O poder do `split`

Recebemos de um cliente um arquivo que, segundo ele, era para ser um CSV (valores separados por vírgula). Contudo, como nada é perfeito, logo na primeira linha há evidencia que foram utilizados vírgula, hífen e ponto e vírgula. Com certeza, nada foi padronizado!

Declare a seguinte string que nos dá um exemplo da situação atual:

```
var arquivo = '100,200-150,200;20';
```

Precisamos em JavaScript extrair todos os valores dessa string para no final somá-los. Para tarefas como essa, há o método `split` que todo objeto String possui. Nela, podemos passar um separador que ele se encarregará de criar um array com cada item considerando o separador utilizado. Contudo, temos um problema com essa estrutura precária. Vejamos o resultado do `split`:

```
var valores = arquivo.split(',');
```

Quando imprimimos os `valores` no console temos:

```
["100", "200-150", "200;20"]
```

Com certeza não é isso que queremos, queremos cada valor em uma posição do array. E agora? A boa notícia é que a função `split` aceita receber como critério uma expressão regular. Vamos criar uma que consiga o que desejamos:

```
var exp = /[,;-]/;
```

Agora, vamos passar nossa expressão para `split`:

```
var exp = /[,;-]/;
var valores = arquivo.split(exp);
```

Quando imprimimos no console os `valores` temos:

```
["100", "200", "150", "200", "20"]
```

Excelente. Agora o desenvolvedor pode iterar pela lista, convertendo os valores de `string` para `number` e totalizar.

