Splitting a String into an Array of Substrings

In this lesson, we will split strings into arrays using the 'split' method.

The Split Method

The `split` method splits a string into an array of substrings. Split expects one argument describing how the split should be made.

For instance, in the programming world, we often process CSV files. CSV stands for Comma Separated Values. Let's create an array from a CSV line:

```
const line = '19,65,9,17,4,1,2,6';
console.log(line.split( ',' ));//console.log for printing
//["19", "65", "9", "17", "4", "1", "2", "6"]
```

If we want to create an array containing each character in the string, we can pass an empty string to the `split` method:

```
const line = '19,65,9,17,4,1,2,6';
console.log(line.split( '' ));
//["1", "9", ",", "6", "5", ",", "9", ",", "1", "7", ",", "4", ",", "1", ",", "2", ",", "6"]
```

As the contents of arrays can be changed, we can easily change the digit `4` inside the `digits` sequence:

```
let digitsArray = '0123456789'.split( '' );
digitsArray[4] = 'X';
console.log(digitsArray)
//["0", "1", "2", "3", "X", "5", "6", "7", "8", "9"]
```
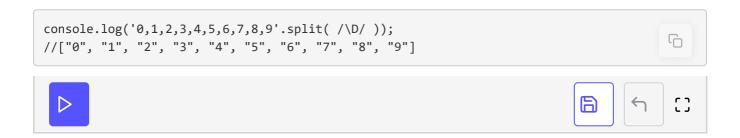
Splitting using Regular Expression

Split also works with a regular expression argument. In this case, the regex describes the pattern used for splitting strings. For instance, `/\D/` matches every character that is not a digit. Splitting based on this regular expression works as follows:

```
console.log('0,1,2,3,4,5,6,7,8,9'.split( /\D/ ));
//["0", "1", "2", "3", "4", "5", "6", "7", "8", "9"]
```

If you want to learn more about regular expressions, check out my JavaScript Regex Udemy video course. Use the coupon `JSREGEX` to get it at minimum price.