Rest Parameters

introduction to rest parameters, calling functions with a variable number of arguments and its potential errors

The Spread operator and Rest parameters are two related features in ES2015 that are worth learning. You can do cool things with them, and they often make your code more compact than the equivalent ES5 code.

Rest parameters

In some cases, you might want to deal with processing a variable number of arguments. In ES5, it was possible to use the `arguments` array inside a function to access them as an array:

```
(
  function()
  {
   console.log( arguments );
  }
) ( 1, 'Second', 3 );
```

In ES2015, the last argument of a function can be preceded by `...`. This argument collects all the remaining arguments of the function in an array. The name for this construct is *rest parameters*, because it contains the rest of the parameters passed to a function.

Let's rewrite the above function in ES2015:

```
( (...args) => // using rest parameters
  {
    console.log( args );
  }
) ( 1, 'Second', 3 );
```

Note that the argument list containing the rest parameter is placed in parentheses. This is mandatory, as `...args` is equivalent to `arg1, arg2, arg3`.

The rest parameter has to be the last argument of a function. As a consequence, there can only be one rest parameter in a function. If the rest parameter is not the last argument of the argument list of a function, an error is thrown.

In the next lesson, we'll discuss the spread operator.