CONSTRUCTOR NEW KEYWORD

We know that a constructor function is used to construct objects

And it is good practice to include the new keyword when executing the constructor function

But what happens if you forget to include the new keyword?

You will end up creating a this object which refers to its parent object (usually the window object)

Thus you will be polluting the global scope / environment 

CONSTRUCTOR NEW KEYWORD

One solution is to use strict mode

JavaScript’s strict mode was introduced in ECMAScript 5, and is a way to opt in to a restricted variant of JavaScript, thereby implicitly opting-out of "sloppy mode"

In strict mode, an exception is thrown when a constructor function is not called with the new keyword

CONSTRUCTOR NEW KEYWORD

This is okay, but not great at providing meaningful feedback

Often you'll want to force usage of the new keyword

This is why JavaScript language introduced the new.target property

The new.target is available in all functions

CONSTRUCTOR NEW KEYWORD

By leveraging the new.target, you can force users of the constructor function to call it with the new keyword

You can throw an error if they don’t do so

If you throw an exception, execution of the current function will stop (the statements after throw won't be executed)

The Error is an inbuilt JavaScript object that allows us to write a custom error message