We were given the task of explaining the differences between the Undefined and Null datatypes in Javascript, which is what this post will do.

Undefined vs Null in Javascript

In JavaScript, both **null** and **undefined** represent the absence of a meaningful value, but they are used in slightly different contexts and have some nuanced differences. Both **null** and **undefined** are primitive value in JavaScript.

Many times we often get confused on what the difference between UNDEFINED and NULL is. Simply put, undefined means a variable has been declared but has not yet been assigned a value. undefined is a type by itself (undefined). Unassigned variables are initialized by JavaScript with a default value of undefined.

On the other hand, null is an object. It can be assigned to a variable as a representation of no value. JavaScript never sets a value to null. That must be done programmatically

Undefined vs null - the differences

Data types:

The data type of undefined is undefined whereas that of null is object. We can find the datatypes of both undefined and null using the typeof operator.

In arithmetic operations:

When used in arithmetic operations, undefined will result in NaN (not a number), whereas null will be converted to 0 behind the screens.

Undefined and null are falsy:

When used in conditional logic both undefined and null will return false.

In summary, "undefined" usually indicates a variable that hasn't been assigned a value, while "null" indicates a deliberate absence of a value. They serve different purposes and are used in different contexts depending on the programming language and the specific requirements of your code.