

Common JS Errors

1. **Uncaught TypeError**: Cannot read property of undefined In JavaScript

'Undefined' is the property of the global object. If you don't assign any value to a variable is of type 'undefined'. A code also return the undefined value when the evaluated variable doesn't have any assigned value.

Debug: If you get undefined error, you need to make sure that which ever variables throws undefined error, is assigned a value to it.

2. The **"TypeError: 'undefined' is not an object"** error occurs when a property is accessed or a method is called on an undefined object. So define it
3. **Typeerror null is not an object**: If you try using a DOM element in your JavaScript before the element is loaded. You will get this error if the DOM elements have **not** been created before loading the script.
4. **unknown script error** : This happens when your JS code is hosted on a different domain (i.e. a CDN). To fix, set Access-Control-Allow-Origin: * on the JS file, and crossorigin="anonymous" on the <script> tag
5. The **"object doesn't support this property or method"** error occurs when you try to use a method or property that the specified object does not support. The ActiveDocument property returns a Document object. There is no Copy method or property available for the Document object, therefore an error occurs. So make sure to use the methods only available for that object
6. **Undefined is not a function**: This is a common JavaScript error that happens when you try to call a **function** before it is **defined**. You get this error when you try to execute a **function** that is uninitialized or improperly initialized . It means that the expression did **not** return a **function** object.
7. **Uncaught Range Error: Maximum call stack**: It **means** that somewhere in your code, you are **calling** a function which in turn **calls** another function and so forth, until you hit the **call stack limit**. This is almost always because of a recursive function with a base case that isn't being met.

8. **TypeError: Cannot read property 'length'**: This error indicates that our code expects to have an object with a **length property**, but that object was not present. **Length** is commonly used on string and array , but a custom object could also have this **property**. This is a blocking error, and execution will stop.
9. **Cannot set property**: If a variable has been declared, but has not been assigned a value, is automatically assigned the value undefined . Therefore, if you try to access the value of such variable, it will throw Uncaught TypeError **cannot set property** of undefined .
10. **Reference: event is not defined: ReferenceError: event is not defined**. This error is thrown when you try to access a variable that is **undefined** or is outside the current scope. You can test it very easily in Chrome browser. If you're getting this error when using the **event** handling system, make sure you use the **event** object passed in as a parameter