**Fix for the Errors**

1. **Uncaught Type Error: cannot read property:**

This error occurs when we try access the data that are not initialized or not defined. So to prevent this from occurrence use

**Fix:**

If data != undefined then { access the data }

1. **Type Error: Undefined is not an object**

This is an error that occurs in Safari when you read a property or call a method on an undefined object.

**Fix:**

If ( data != undefined && data.property) then { access the data }

1. **Type Error: null is not an object**

This is also an error that occurs in Safari when you read a property or call a method on a null object. This error will occur if we try to access DOM before loading

**Fix:**

We can resolve the issue by adding an event listener that will notify us when the page is ready. Once the addEventListener is fired, the init() method can make use of the DOM elements.

1. **Script error**

This is what browsers send to the onerror callback when an error originates from a JavaScript file served from a different origin (different domain, port, or protocol). Even though there’s an error occurring, you don’t know what the error is, nor from which code it’s originating.

**Fix:** Set crossorigin = "anonymous" and

Set Access-Control-Allow-Origin: "\*"

Once both of these are set, any errors triggered by the script will report to window.onerror just like any regular same-domain script. So instead of “Script error”, then we can know what kind of error it is

1. **TypeError: Object doesn’t support property**

This is an error that occurs in IE when you call an undefined method. You can test this in the IE Developer Console, which is same as "is not afunction error" in Chrome.

**Fix:**

check this.function != undefined

1. **TypeError: ‘undefined’ is not a function**

This is an error that occurs in Chrome when you call an undefined function.

For Example:

function clearBoard(){

  alert("Cleared");

}

document.addEventListener("click", function(){

  this.clearBoard();    // This will cause an error ?

});

Error occurs because "this" refers to the function is executed in the context of document where clearBoard is defined on window.

**Fix:**

var self=this;  // save reference to 'this', while it's still this!

document.addEventListener("click", function(){

  self.clearBoard();

});

1. **Uncaught RangeError: Maximum call stack**

This kind of error occurs if we call a function recursively that does not terminated or if we pass value to a function that is out of range

For Example:

Number.toExponential(digits) and Number.toFixed(digits) accept digits from 0 to 100, and Number.toPrecision(digits) accepts digits from 1 to 100.

**Fix:** Use numbers that are in range for a function.

1. **TypeError: Cannot read property length**

This is an error occur because of reading length property for an undefined variable.

**Fix:** Check whether the variable is initialized and have the length property

1. **Uncaught TypeError: Cannot set property**

When we try to access an undefined variable it always returns undefined and we cannot get or set any property of undefined.

For Example:

var a = undefined;

a.value = 10 // will throw error

1. **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.

**Fix:** Provide event to the function

document.addEventListener("mousemove", function (event) {

  console.log(event);

})