# **TYPES OF ERRORS**

## Uncaught TypeError: Cannot read property

Initialize state with reasonable default values in the constructor.

## TypeError: ‘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. You can test this very easily in the Safari Developer Console. This is essentially the same as the above error for Chrome, but Safari uses a different error message.

## TypeError: null is not an object

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.

## (unknown): Script error

To resolve, do the following,

* + **Send the Access-Control-Allow-Origin header**
  + **Set crossorigin="anonymous" on the script tag.**

## TypeError: Object doesn’t support property

This is equivalent to the error "TypeError: ‘undefined’ is not a function" in Chrome.

## Uncaught RangeError

This is an error that occurs in Chrome under a couple of circumstances. One is when you call a recursive function that does not terminate. You can test this in the Chrome Developer Console.

## TypeError: Cannot read property ‘length’

You have two ways to resolve your issue:

* + Remove parameters in the function declaration statement (it turns out you want to access those variables that are declared outside of the function, so you don’t need parameters for your function).
  + Invoke the function passing it the array that we declared.

## 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. In that case, an application will throw “Uncaught TypeError cannot set property of undefined”.

## ReferenceError: event is not defined

If you’re getting this error when using the event handling system, make sure you use the event object passed in as a parameter. Older browsers like IE offer a global variable event, and Chrome automatically attaches the event variable to the handler. Firefox will not automatically add it. Libraries like jQuery attempt to normalize this behavior. Nevertheless, it’s best practice to use the one passed into your event handler function.