

I want you to become a Grandmaster.

So, let me explain something to you...

The `Event` object is automatically passed to all event handlers. An event handler is just a piece of code, or function, that executes code when an event happens. In our example, the `loadSecret()` function is our event handler.

That is why we could pass in the Event object when defining our `onsubmit` event handler, as follows:

```
onsubmit="loadSecret(event)";
```


Important: This is the recommended way of passing the event object into the `loadSecret()` function.

Now, in the video, I also mentioned that you can directly use the `event` keyword and access the Event object directly inside of the `loadSecret()` function. This is only possible because one can access the current event through `window.event`. Just using `event` is implicitly accessing `window.event`.

What is the problem with this approach?

Answer: It is currently being phased out. Although it works in Chrome, it may not work properly in other browsers. Here is a snapshot from MDN's website:

Window.event

 **Deprecated:** This feature is no longer recommended. Though some browsers might still support it, it may have already been removed from the relevant web standards, may be in the process of being dropped, or may only be kept for compatibility purposes. Avoid using it, and update existing code if possible; see the [compatibility table](#) at the bottom of this page to guide your decision. Be aware that this feature may cease to work at any time.

The read-only `Window` property `event` returns the `Event` which is currently being handled by the site's code. Outside the context of an event handler, the value is always `undefined`.

You should avoid using this property in new code, and should instead use the `Event` passed into the

You *should* avoid using this property in new code, and should instead use the [Event](#) passed into the event handler function. This property is not universally supported and even when supported introduces potential fragility to your code.

Hope this makes sense.

Let's march on.

