

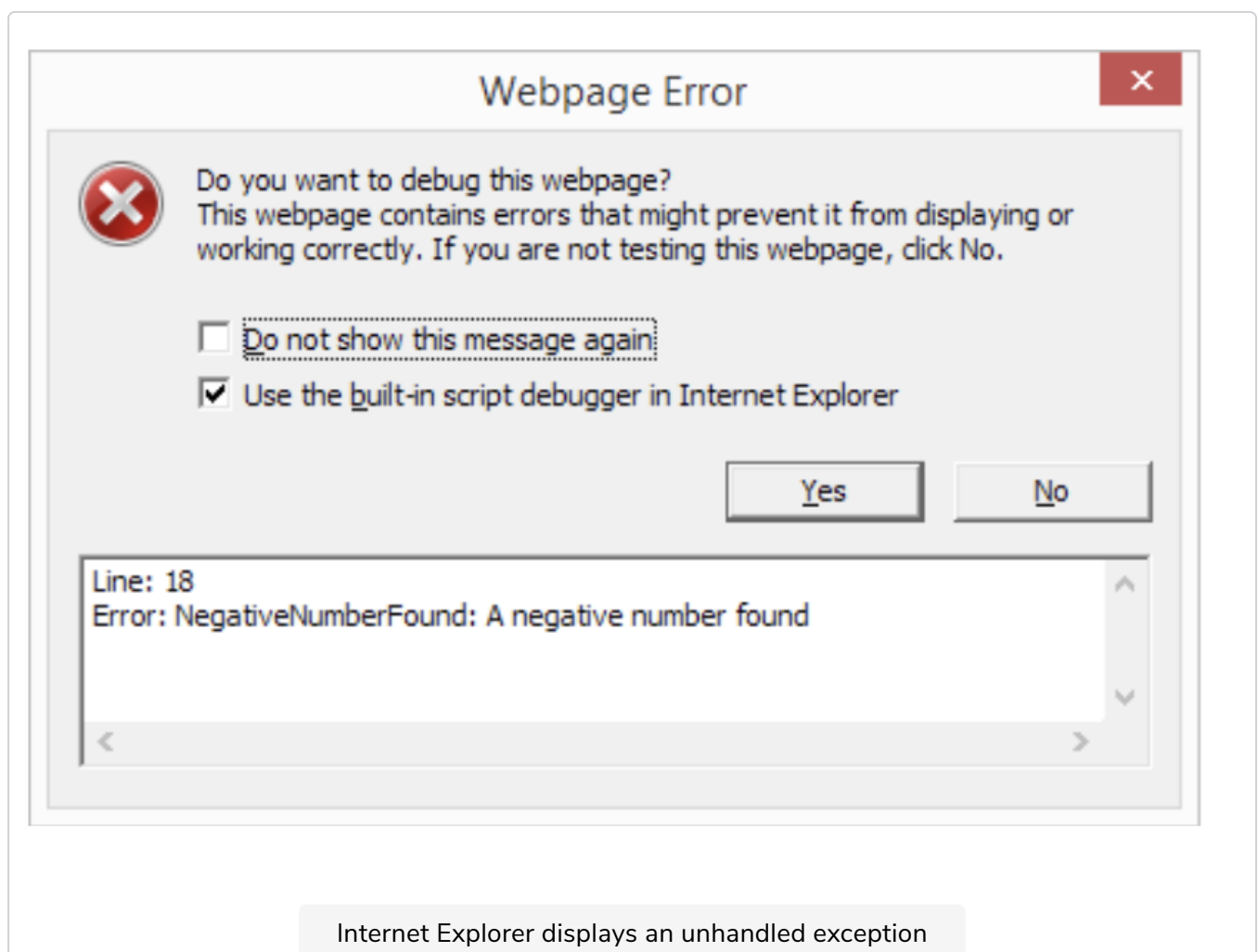
# The onerror Event

In this lesson, we will meet the onerror event in detail.  
Let's begin!

## WE'LL COVER THE FOLLOWING

- Listing 8-23: Exercise-08-24/index.html

Depending on how your browser is configured, unhandled errors (exceptions that have not been handled by the code) are displayed. For example, Internet Explorer displays the `NegativeNumberUsed` exception specified in the previous code snippet as shown below:



The `window` object provides an `onerror` event and other events, as described

in Table 6-5. You can provide an event handler to define a way to handle

exceptions that are otherwise not treated by the page.

The `onerror` event expects an event handler function with three arguments,

- the exception message
- the document URL that causes the unhandled exception
- a line number referring to the code line where the exception originates from

Listing 8-23 demonstrates the usage of the `onerror` event handler method.

## Listing 8-23: Exercise-08-24/index.html #

```
<!DOCTYPE html>
<html>
<head>
  <title>The onerror event</title>
  <script>
    function sometimesWorks(arg) {
      if (arg == 1) return;
      throw "You're unlucky";
    }

    function bet(arg, catchHere) {
      if (catchHere) {
        try {
          sometimesWorks(arg);
        } catch (error) {
          // --- Caught
        }
      } else {
        sometimesWorks(arg);
      }
    }

    function tryThis() {
      onerror =
        function (message, url, line) {
          alert("URL '" + url
            + " caused this message '"
            + message + "' in line "
            + line);
          return false;
        };

      // These work
      bet(1, true);
      bet(1, false);

      // This raises an error
      bet(3, false);
    }
  </script>
</head>
</html>
```

```
    }  
    </script>  
</head>  
<body onload="tryThis()">  
</body>  
</html>
```

The `sometimesWorks()` method raises an exception when invoked with any argument except 1. The `bet()` method accepts a parameter that specifies whether any exception should be caught within the method body or not.

When the page is loaded, `tryThis()` is invoked, and it configures the `onerror` event to display a message when an unhandled exception is caught. In this method, `bet()` is called three times, and the last call raises an exception that is caught by `onerror`.

---

In the *next lesson*, we will learn how to manage errors.