

# Using Event Parameters

In this lesson, we will learn how to use event parameters in event handlers.  
Let's begin!

## WE'LL COVER THE FOLLOWING



- Listing 6-15: Catching event parameters

Most events have parameters that you can use in event handlers.

For example, when you catch the `onmousedown` event, you may utilize the mouse pointer coordinates. When you pass the event object to the handler method, you can access the parameters of an event.

Listing 6-15 shows an example.

## Listing 6-15: Catching event parameters #

```
<!DOCTYPE html>
<html>
<head>
  <title>Catching event parameters</title>
  <style>
    #rectangle {
      position: absolute;
      left: 24px;
      top: 24px;
      width: 200px;
      height: 100px;
      background-color: navy;
      color: white;
    }

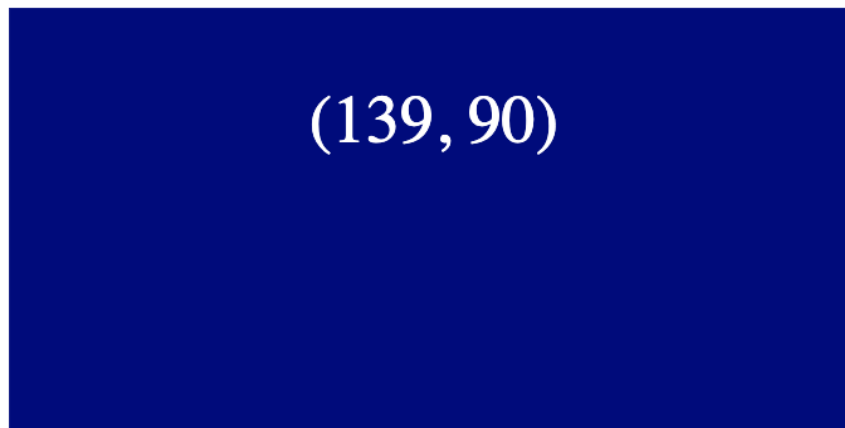
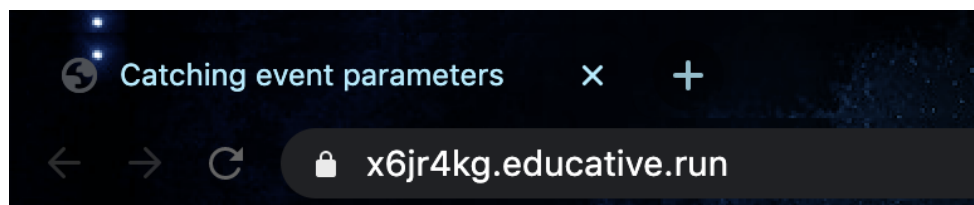
    #pos {
      text-align: center;
    }
  </style>
</head>
<body>
  <div id="rectangle" onmousedown="handler(event)">
    <p id="pos">(?, ?)</p>
  </div>
```

```
<script>
function handler(event) {
  var pos = document.getElementById('pos');


  pos.textContent = '(' + event.x + ', ' + event.y + ')';
}
</script>
</body>
</html>
```

Here, the `onmousedown` attribute contains the `“handler(event)”` value, so when you click the mouse button while the pointer is over the rectangle represented by the `<div>` tag, the event parameters are passed to the `handler()` method. In the body of `handler()` you can access the properties of the event.

Among the others, `x` indicates the horizontal and `y` the vertical coordinate of the mouse pointer. The current coordinates are displayed in the paragraph nested into the rectangle, as shown below:



The event parameters can be caught

 **NOTE:** Later in this course you will learn more about events and their parameters.

In the next lesson, we will summarize what we have learned in this chapter.

See you there!