DOM
Document Object Model

JAVASCRIPT DOM –

JavaScript can access and change all the elements of an HTML document.

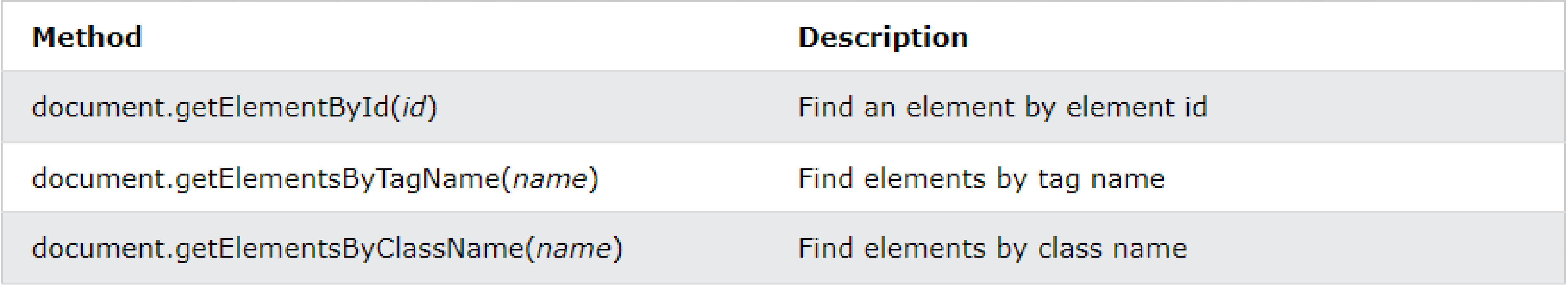
With the object model, JavaScript gets all the power it needs to create dynamic HTML.

JavaScript can change all the HTML elements in the page
JavaScript can change all the HTML attributes in the page
JavaScript can change all the CSS styles in the page
JavaScript can remove existing HTML elements and attributes
JavaScript can add new HTML elements and attributes
JavaScript can react to all existing HTML events in the page
JavaScript can create new HTML events in the page

THE HTML DOM DOCUMENT OBJECT -:

- ~ The document object represents your web page.
- ~ If you want to access any element in an HTML page, you always start with accessing the document object.

- ~ Below are some examples of how you can use the document object to access and manipulate HTML.



Method	Description
document.getElementById(*id*)	Find an element by element id
document.getElementsByTagName(*name*)	Find elements by tag name
document.getElementsByClassName(*name*)	Find elements by class name

1. FINDING HTML ELEMENTS BY ID:

~ The easiest way to find an HTML element in the DOM, is by using the element id.

```
<body>

        <p id=“demo” style=“color:green;”></p>

        <script>
                var A = get.ElementById(“demo”);
                A.innerHTML = “ Hello World ”;
        </script>

</body>
```

Output : Hello World

2. FINDING HTML ELEMENTS BY CLASS NAME :

~ The way to find an HTML element in the DOM, is by using the element ClassName .

```
<body>

        <p class=“demo” style=“color:green;”></p>

        <script>
                var A = get.ElementByClassName(“demo”);
                A.innerHTML = “ Hello World ”;
        </script>

</body>
```

Output : Hello World

3. FINDING HTML ELEMENTS BY TAG NAME :

~ The way to find an HTML element in the DOM, is by using the element TagName.

```
<body>

        <p style=“color:green;”></p>

        <script>
                var A = get.ElementByTagName(“p”);
                A.innerHTML = “ Hello World ”;
        </script>

</body>
```

Output : Hello World

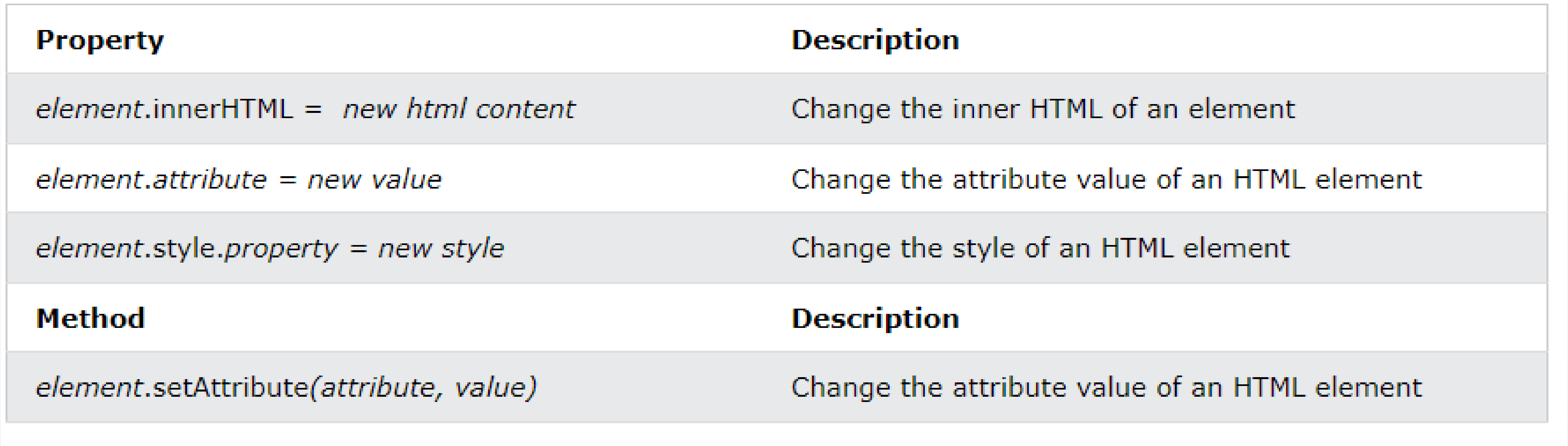
4. FINDING HTML ELEMENTS BY CSS SELECTOR:

~ The easiest way to find an HTML element in the DOM, is by using the element id.

```
<body>

        <p class=“demo” style=“color:green;”></p>

        <script>
                var A = document
                getElementByQuarySelector(“p.class”);
                A.innerHTML = “ Hello World ”;

        </script>

</body>
```

Output : Hello World

CHANGING HTML ELEMENT -:



Property	**Description**
element.innerHTML = *new html content*	Change the inner HTML of an element
element.attribute = *new value*	Change the attribute value of an HTML element
element.style.*property* = *new style*	Change the style of an HTML element
Method	**Description**
element.setAttribute*(attribute, value)*	Change the attribute value of an HTML element

1. CHANGING THE VALUE OF AN INNERHTML :

~ The easiest way to modify the content of an HTML element is by using the innerHTML property.

```
<body>

        <p class=“demo” style=“color:green;”> Hello world</p>

         <script>
                document.getElementById("p1").innerHTML = "New text!";
        </script>

</body>
```

Output : New text

2. CHANGING THE VALUE OF AN ATTRIBUTE :

~ To change the value of an HTML attribute, use this syntax :

```
<img id="image" src="smiley.gif" width="160" height="120">

          <script>
                    document.getElementById("image").src = "landscape.jpg";
          </script>

</body>
```

Output : src=“landscape.jpg”

3. CHANGING HTML STYLE :

~ To change the style of an HTML element, use this syntax :

syntax - : document.getElementById(id).style.property = new style

```
<body>
        <p id="p2">Hello World!</p>

                <script>
                        document.getElementById("p2").style.color = "blue";
                        document.getElementById("p2").style.fontFamily = "Arial";
                        document.getElementById("p2").style.fontSize = "larger";
                </script>
</body>
```

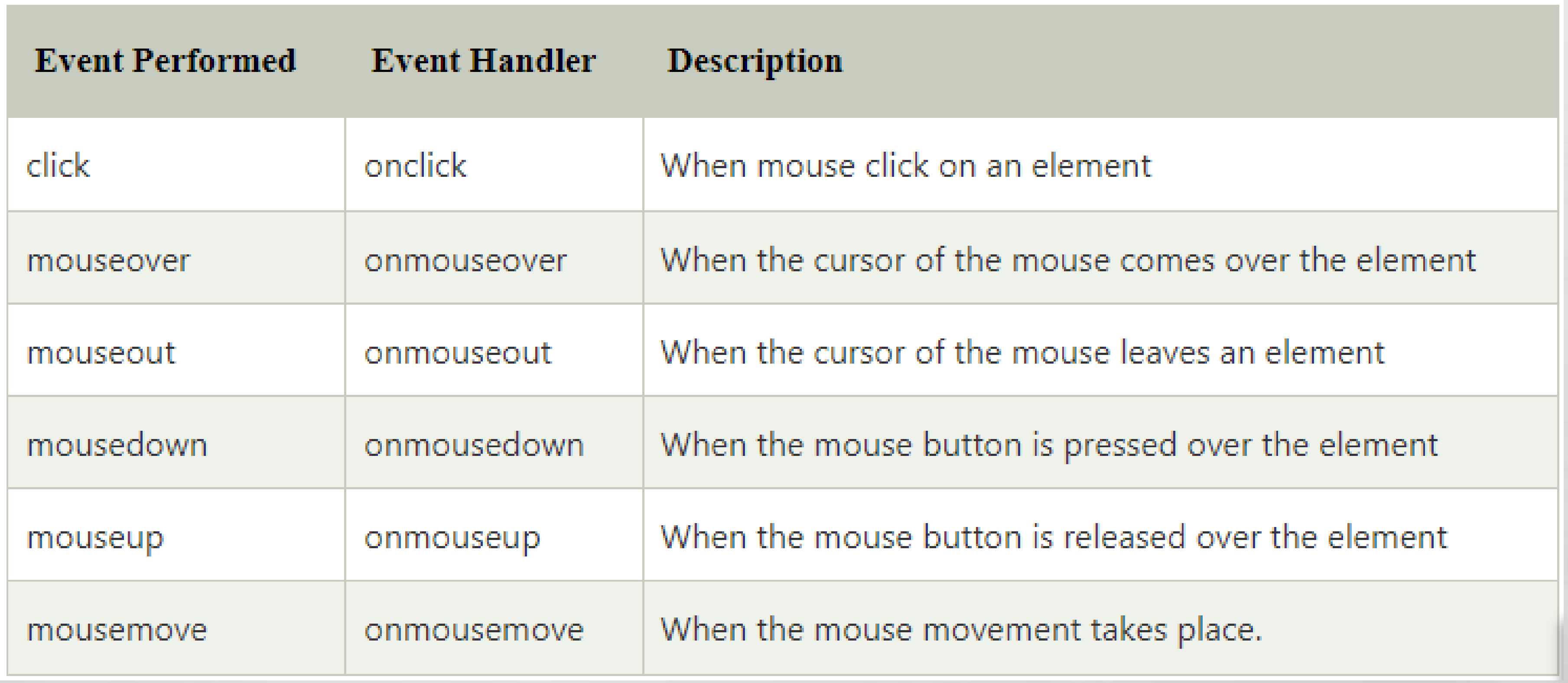
Output : hello world

DOM EVENT

THE JAVASCRIPT EVENTS -:

~ The change in the state of an object is known as an Event. In html, there are various events which represents that some activity is performed by the user or by the browser. When javascript code is included in HTML, js react over these events and allow the execution.

Mouse events :



Event Performed	Event Handler	Description
click	onclick	When mouse click on an element
mouseover	onmouseover	When the cursor of the mouse comes over the element
mouseout	onmouseout	When the cursor of the mouse leaves an element
mousedown	onmousedown	When the mouse button is pressed over the element
mouseup	onmouseup	When the mouse button is released over the element
mousemove	onmousemove	When the mouse movement takes place.

1. ONCLICK: When mouse click on an element.

```
<body>

        <input type="button" onclick="clickevent()" value="Who's this ?"/>

         <script>
                function clickevent()
                        {
                           alert(“this is ajay”);
                        }
        </script>

</body>
```

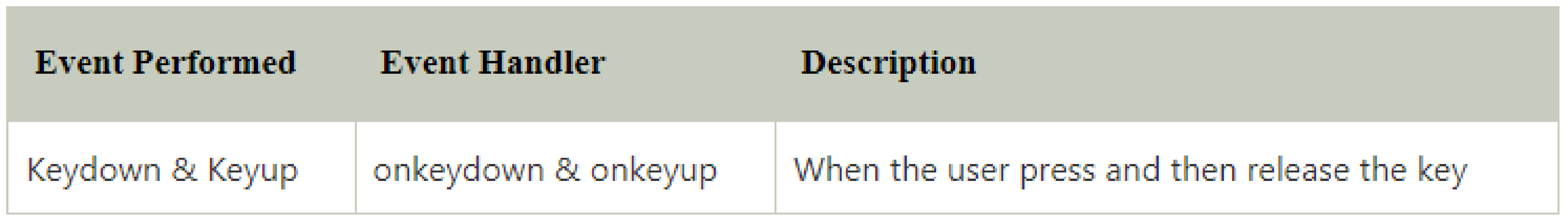
2. ONMOUSEOVER & ONMOUSEOUT : When the cursor of the mouse comes over and out over the element.

```
<body>

        <input type="button" onmouseover="overevent()" onmouseout="outevent" value="Who's this
?"/>

         <script>
                function overevent()
                        {
                           alert("this is ajay");
                        }

                function outevent()
                        {
                       alert("thankyou");
                        }
        </script>

</body>
```

3. ONMOUSEUP & ONMOUSEDOWN : When the mouse button is pressed over the element and mouse button is released over the element

```
<body>

<input type="button" onmouseup=“upevent()“ onmousedown=“downevent” value="Who's this ?"/>

            <script>
                      function upevent()
                                 {
                                    alert(“this is ajay”);
                                 }

                      function downevent()
                                 {
                                alert(“thankyou”);
                                 }
           </script>

</body>
```

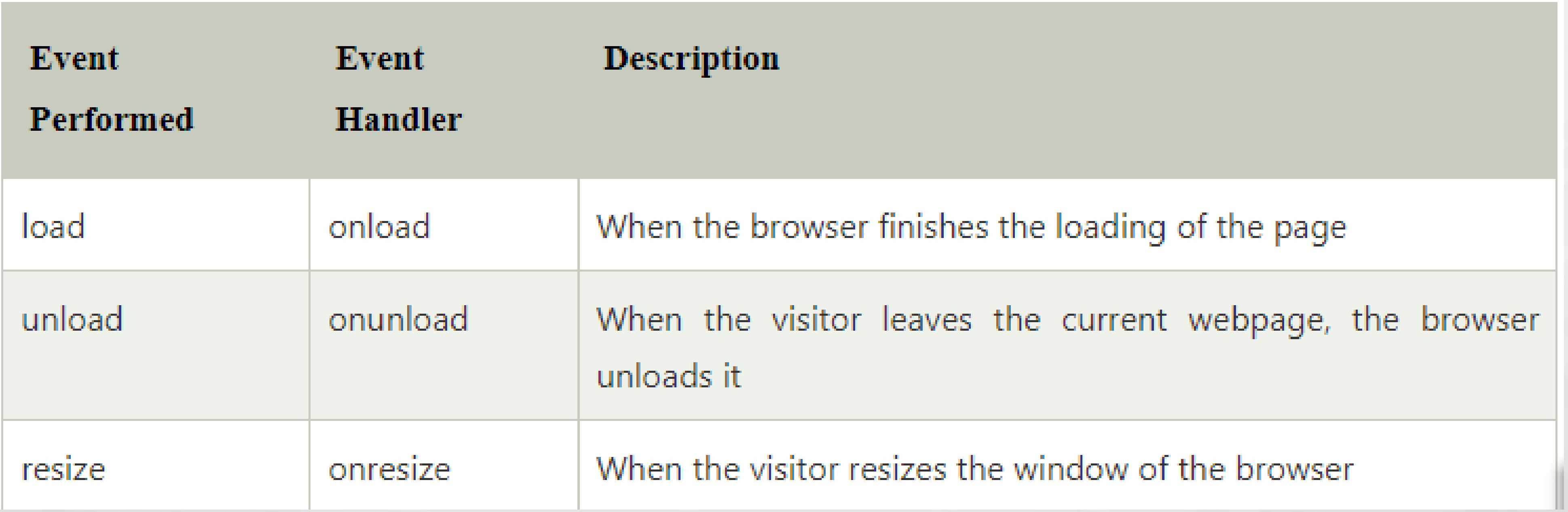
Keyboard events:



Event Performed	Event Handler	Description
Keydown & Keyup	onkeydown & onkeyup	When the user press and then release the key

```
<body>
        <h2> Enter something here</h2>
        <input type="text" id="input1" onkeydown="keydownevent()"/>
                <script>
                        function keydownevent()
                                {
                                 document.getElementById("input1");
                                alert("Pressed a key");
                                }
                </script>
</body>
```

Window/Document events :

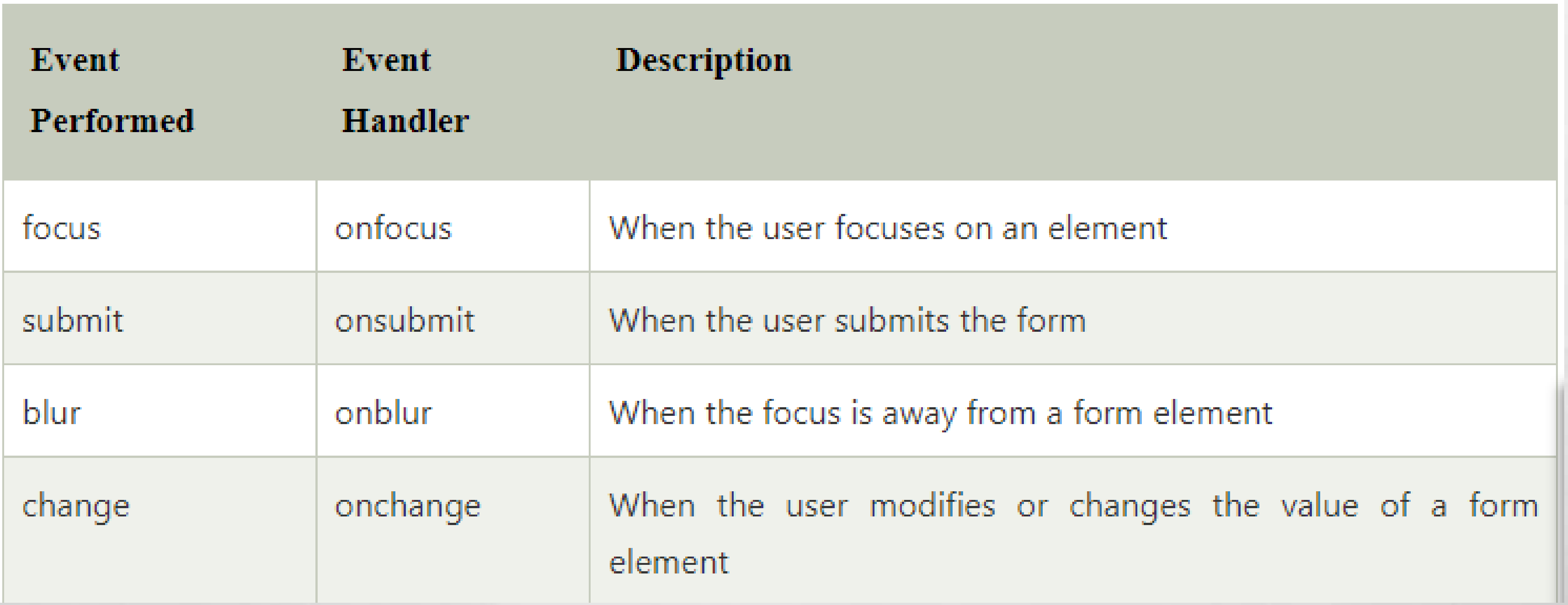


Event Performed	Event Handler	Description
load	onload	When the browser finishes the loading of the page
unload	onunload	When the visitor leaves the current webpage, the browser unloads it
resize	onresize	When the visitor resizes the window of the browser

Onload events:

```
        // <body onload="window.alert('Page successfully loaded');">  //


<body onload=“loadevent()”>

        <script>
          function loadevent(){
                window.alert("The page is loaded successfully");
                }
        </script>


</body>
```

Form events :



Event Performed	Event Handler	Description
focus	onfocus	When the user focuses on an element
submit	onsubmit	When the user submits the form
blur	onblur	When the focus is away from a form element
change	onchange	When the user modifies or changes the value of a form element

ONFOCUS: When the user focuses on the element ,

```
<body>

<input type="“text" id="input1" onfocus="focusevent()"/>

        <script>
                function focusevent()
                {
                 document.getElementById("input1").style.background=" aqua";
                 }
        </script>
</body>
```

JAVASCRIPT ADDEVENT LISTENER()

THE JAVASCRIPT EVENTS -:

~ The addEventListener() method is used to attach an event handler to a particular element. It does not override the existing event handlers. Events are said to be an essential part of the JavaScript.

The addEventListener() method is an inbuilt function of JavaScript. We can add multiple event handlers to a particular element without overwriting the existing event handlers.

Syntax

```
element.addEventListener(event, function);
```

EXAMPLE :

```
<body>
        <button id = "btn"> Click me </button>

        <p id = "para"></p>

                <script>
                        document.getElementById("btn").addEventListener("click", fun);

                        function fun() {

                                document.getElementById("para").innerHTML = "Hello World“;

                                        }
                </script>
</body>
```

onclick event

function