SUB: Design Pattern And Framework

TOPIC: JAVASCRIPT

Java Script - Events

What is an Event?

JavaScript's interaction with HTML is handled through events that occur when the user or the browser manipulates a page.

When the page loads, it is called an event. When the user clicks a button, that click too is an event. Other examples include events like pressing any key, closing a window, resizing a window, etc.

Developers can use these events to execute JavaScript coded responses, which cause buttons to close windows, messages to be displayed to users, data to be validated, and virtually any other type of response imaginable.

Events are a part of the Document Object Model (DOM) Level 3 and every HTML element contains a set of events which can trigger JavaScript Code.

Event Object	Description
AnimationEvent	For CSS animations
ClipboardEvent	For modification of the clipboard
DragEvent	For drag and drop interaction
FocusEvent	For focus-related events
HashChangeEvent	For changes in the anchor part of the URL
InputEvent	For user input
KeyboardEvent	For keyboard interaction
MouseEvent	For mouse interaction
PageTransitionEvent	For navigating to, and away from, web pages
PopStateEvent	For changes in the history entry
ProgressEvent	For the progress of loading external resources

StorageEvent	For changes in the window's storage area.
TouchEvent	For touch interaction
TransitionEvent	For CSS transitions
UiEvent	For user interface interaction
WheelEvent	For mousewheel interaction

MouseEvent

Event	**Description**
onclick	The event occurs when the user clicks on an element
oncontextmenu	The event occurs when the user right-clicks on an element to open a context menu
ondblclick	The event occurs when the user double-clicks on an element
onmousedown	The event occurs when the user presses a mouse button over an element
onmouseenter	The event occurs when the pointer is moved onto an element
onmouseleave	The event occurs when the pointer is moved out of an element
onmousemove	The event occurs when the pointer is moving while it is over an element
onmouseout	The event occurs when a user moves the mouse pointer out of an element, or out of one of its children
onmouseover	The event occurs when the pointer is moved onto an element, or onto one of its children
onmouseup	The event occurs when a user releases a mouse button over an element

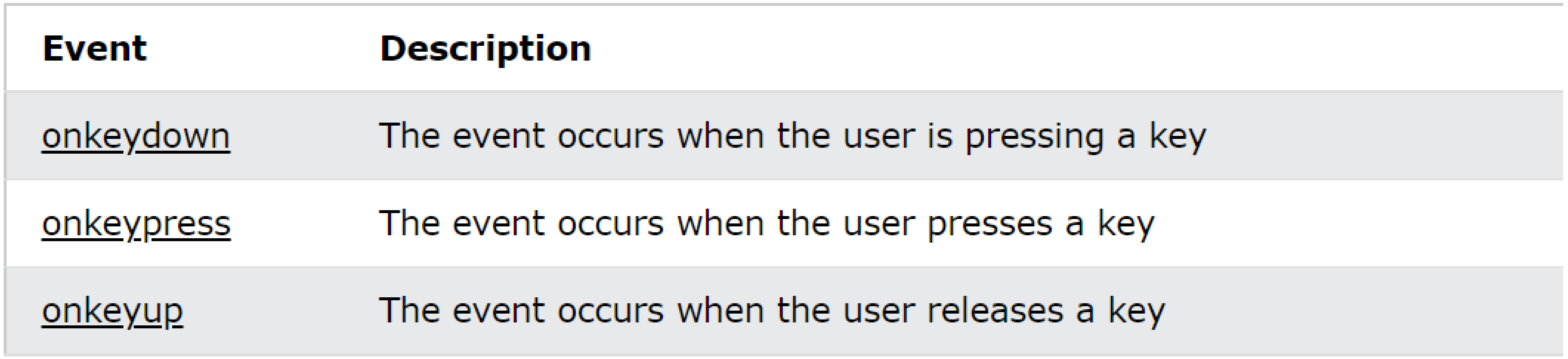
Java Script - Events

```
<html>
<head>
<script type="text/javascript">
<!--
function over() {
   document.write ("Mouse Over");
}
function out() {
   document.write ("Mouse Out");
}
//-->
</script>
</head>
```

Java Script Events

```
<body>
<p>Bring your mouse inside the division to see the result:</p>
<div onmouseover="over()" onmouseout="out()">
<h2> This is inside the division </h2>
</div>
</body>
</html>
```

Keyboard Event



Event	Description
onkeydown	The event occurs when the user is pressing a key
onkeypress	The event occurs when the user presses a key
onkeyup	The event occurs when the user releases a key

Java Script – Dialog Box

Alert Dialog Box

An alert dialog box is mostly used to give a warning message to the users. For example, if one input field requires to enter some text but the user does not provide any input, then as a part of validation, you can use an alert box to give a warning message. Nonetheless, an alert box can still be used for friendlier messages.

Alert box gives only one button "OK" to select and proceed.

Confirmation Dialog Box

A confirmation dialog box is mostly used to take user's consent on any option. It displays a dialog box with two buttons: **OK** and **Cancel**.

If the user clicks on the OK button, the window method **confirm()** will return true. If the user clicks on the Cancel button, then **confirm()** returns false. You can use a confirmation dialog box as follows.

Java Script – Dialog Box

```
<html>
<head>
<script type="text/javascript">
<!--
function getConfirmation(){
   var retVal = confirm("Do you want to continue ?");
   if( retVal == true ){
      document.write ("User wants to continue!");
       return true;
   }else{
      Document.write ("User does not want to continue!");
       return false;
   }
}
//-->
</script>
</head>
```

Java Script – Dialog Box

```
 <body>
 <p>Click the following button to see the result: </p>
 <form>
<input type="button" value="Click Me" onclick="getConfirmation();" />
</form>
</body>
</html>
```

Java Script – Dialog Box

Prompt Dialog Box

The prompt dialog box is very useful when you want to pop-up a text box to get user input. Thus, it enables you to interact with the user. The user needs to fill in the field and then click OK.

This dialog box is displayed using a method called **prompt()** which takes two parameters: (i) a label which you want to display in the text box and (ii) a default string to display in the text box.

This dialog box has two buttons: **OK** and **Cancel**. If the user clicks the OK button, the window method **prompt()** will return the entered value from the text box. If the user clicks the Cancel button, the window method **prompt()** returns **null**.

Java Script – Dialog Box

```
<html>

<head>

<script type="text/javascript">

<!--

function getValue(){

   var retVal = prompt("Enter your name : ", "your name here");


   document.write("You have entered : " +  retVal);

}

//-->
```

Java Script – Dialog Box

```
</script>
</head>
<body>
<p>Click the following button to see the result: </p>
<form>
<input type="button" value="Click Me" onclick="getValue();" />
</form>
</body>
</html>
```

Java Script - Objects

```
<html>
<head>
<title>User-defined objects</title>
<script type="text/javascript">
    var book = new Object();   // Create the object
    book.subject = "Perl"; // Assign properties to the object
    book.author  = "Mohtashim";
</script>
</head>
<body>
<script type="text/javascript">
   document.write("Book name is : " + book.subject + "<br>");
   document.write("Book author is : " + book.author + "<br>");
</script>
</body>
```

Java Script - Objects

```
<html>
<head>
<title>User-defined objects</title>
<script type="text/javascript">

// Define a function which will work as a method
function addPrice(amount){
    this.price = amount;
}
function book(title, author){
    this.title = title;
    this.author  = author;
    this.addPrice = addPrice; // Assign that method as property.
}
</script>
```

Java Script - Objects

```
<body>
<script type="text/javascript">
   var myBook = new book("Perl", "Mohtashim");
   myBook.addPrice(100);
   document.write("Book title is : " + myBook.title + "<br>");
   document.write("Book author is : " + myBook.author + "<br>");
   document.write("Book price is : " + myBook.price + "<br>");
</script>
</body>
</html>
```

Java Script String Data Type and Method

The **String** object lets you work with a series of characters; it wraps Javascript's string primitive data type with a number of helper methods.

As JavaScript automatically converts between string primitives and String objects, you can call any of the helper methods of the String object on a string primitive.

There are many string manipulation functions and properties are available. Some of them are listed below,

Java Script String Data Type and Method

Properties	Functions
length	indexOf()
	lastIndexOf()
	Search(), indexOf();lastIndexOf(); search();slice(start, end); substring(start, end);substr(start, length); replace(); toUpperCase(); toLowerCase(); concat(); trim(); charAt(position); charCodeAt(position); split(); and many more

Java Script String Data Type and Method

Some of the example code is written here more example you can explore on your own:

```
<script>
var str = "Please locate where 'locate' occurs!";
var pos = str.search("locate");
document.write(pos);
</script>
```

JavaScript Arrays

- var cars = [“Maruti", “Mahindra", “Tata"];
- JavaScript arrays are used to store multiple values in a single variable.
- Using an array literal is the easiest way to create a JavaScript Array.
- Syntax:
- var *array_name* = [*item1*, *item2*, ...];

JavaScript Arrays

- Other way of defining an array is,
- var cars = new Array(“Maruti", “Mahindra", “Tata“);

- The way of accessing an element of an array is ,

- var cars = [“Maruti", “Mahindra", “Tata“];
  document.getElementById("demo").innerHTML =
  cars[0];

JavaScript Arrays Methods and Properties

Properties	Methods
Length	Concate()
	sort()
	reverse()
	reduce()
	Filter() and many more