* **NEW REQUEST**

1. What is JSON?

Ans :- JSON stands for **J**ava**S**cript **O**bject **N**otation

JSON is a lightweight format for storing and transporting data

JSON is often used when data is sent from a server to a web page

scribing" and easy to understand

1. What is promises?

Ans : The **Promise** object represents the eventual completion (or failure) of an asynchronous operation and its resulting value.

A Promise is in one of these states:

* *pending*: initial state, neither fulfilled nor rejected.
* *fulfilled*: meaning that the operation was completed successfully.
* *rejected*: meaning that the operation failed.
* **JavaScript Essentials**

1. What is JavaScript Output method?

Ans :- Writing into an HTML element, using innerHTML.

Writing into the HTML output using document. write().

Writing into an alert box, using window. alert().

Writing into the browser console, using console. log().

1. How to used JavaScript Events to do all examples?

* **Ans :-** Onload:  When your page loads, it performs accordingly.
* Onclick: When a user clicks on a button or inputs it occurs.
* Onmouseover: When a user mouses over on the button.
* Onfocus:  Certain scenarios when a user keeps the cursor in a form field.
* Onblur: If a particular form field leaves within it.

Below are the types of JavaScripts with examples of code syntax.

### Onclick Events and Syntax

We can define it as a mouse event that stimulates as per the code logic you determine in your code. Here is the code snippet we can use.

Code:

<!doctype html>

<html>

  <head>

    <script>

      function Greet() {

        alert('Hello World!');

      }

    </script>

  </head>

  <body>

    <button type="button" onclick="Greet()">Please click here! </button>

  </body>

</html>

#### Output

### Onmouseover Event and Syntax

We can use this event for hovering the mouse pointer when we put the cursor and it performs as per the logic of the element which is connected to and its child's elements. We can use the below code snippet.

Code:

<!DOCTYPE html>

<html>

<body>

<h1 id="demo">Test Mouse over me</h1>

<script>

document.getElementById("demo").onmouseover = function() {mouseOver()};

function mouseOver() {

  document.getElementById("demo").style.color = "Purple";

}

</script>

</body>

</html>

#### Output

### Onmouseout Event

When you leave the mouse cursor it moves to the element that controls a mouseout event;, a function associated with it is executed. The below code snippet can help you to understand the logic.

 Code

<!DOCTYPE html>

<html>

<body>

<h1 id="demo">Test Mouse over me</h1>

<script>

document.getElementById("demo").onmouseout = function() {mouseOut()};

function mouseOut() {

  document.getElementById("demo").style.color = "Red";

}

</script>

</body>

</html>

#### Output

### Onchange Event and Syntax

This event identifies the variance in the value of any element listing to this event. The best example of this is when text and dropdown list change events. The below code snippet can help you understand the logic as to how it converts the input name with the upper case when text changes.

Code:

<!DOCTYPE html>

<html>

<body>

Please Enter name: <input type="text" id="Firstname">

<script>

document.getElementById("Firstname").onchange = function() {myFunction()};

function myFunction() {

  var x = document.getElementById("Firstname");

  x.value = x.value.toUpperCase();

}

</script>

</body>

</html>

#### Output

After changing the event:

### Onload Event and Syntax

The JavaScript onload event can be utilized when we have a specific requirement to execute a specific function once the page is represented fully. The below code snippet can help you understand the logic.

Code:

<!DOCTYPE html>

<html>

<body onload="checkyourCookies()">

<p id="OnloadTest"></p>

<script>

function checkyourCookies() {

  var text = "";

  if (navigator.cookieEnabled == true) {

    text = "your web page Cookies are active.";

  } else {

    text = "your web page Cookies are not active.";

  }

  document.getElementById("OnloadTest").innerHTML = text;

}

</script>

</body>

</html>

#### Output

### Onfocus Event and Syntax

This Javascript function performs when the given instruction receives the focus as per the change or click event. The below code snippet can help you understand the logic.

Code:

<!DOCTYPE html>

<html>

<body>

<p>This is the best scenario to uses the addEventListener() function to attach a "focus" event to an input element box.</p>

Enter your First name: <input type="text" id="Firstname">

<script>

document.getElementById("Firstname").addEventListener("focus", myFunction);

function myFunction() {

  document.getElementById("Firstname").style.backgroundColor = "DarkBlue";

}

</script>

</body>

</html>

#### Output

Input text focused

### Onblur Event and Syntax

This Javascript Onblur event triggers when a certain object loses focus. We can execute the below code to understand how to implement it.

Code

<!DOCTYPE html>

<html>

<body>

<p>This code snippet uses the addEventListener() method and performs a "blur" event to an input element.</p>

<p>please write something and see the result (blur).</p>

<input type="text" id="fname">

<script>

document.getElementById("fname").addEventListener("blur", myFunction);

function myFunction() {

  alert("your Input element lost focus.");

}

</script>

</body>

</html>

#### Output