شركـــة تحالـــف الإمـــارات للحــلـــول الـتـقـنـيـــة ذ.م.م.
TAHALUF AL EMARAT TECHNICAL SOLUTIONS L.L.C.

JavaScript

Tahaluf Training Center 2021

شركـــة تحالـــف الإمـــارات للحــلـــول الـتـقـنـيـــة ذ.م.م.
TAHALUF AL EMARAT TECHNICAL SOLUTIONS L.L.C.

شركة تحالف الإمارات للحلول التقنية ذ.م.م.
TAHALUF AL EMARAT TECHNICAL SOLUTIONS L.L.C.

Day 5

'use strict'

The directive looks like a string: "use strict" or 'use strict’.

When it is located at the top of a script, the whole script works the “modern” way.

'use strict'

The purpose of "use strict" is to indicate that the code should be executed in "strict mode".
With strict mode, you can not, for example, use undeclared variables.

You can use strict mode in all your programs. It helps you to write **cleaner code**, like preventing you from using undeclared variables.

'use strict'

```
"use strict";

x = 3.14; // This will cause an error
because x is not declared
```

'use strict'

شركة تحالف الإمارات للحلول التقنية ذ.م.م.
TAHALUF AL EMARAT TECHNICAL SOLUTIONS L.L.C.

```
myFunction();
function myFunction() {
"use strict";
  y = 3.14;   // This will also cause an
error because y is not declared
}
```

'use strict'

Strict mode makes it easier to write "secure" JavaScript.

Strict mode changes previously accepted "bad syntax" into real errors.

As an example, in normal JavaScript, mistyping a variable name creates a new global variable. In strict mode, this will throw an error, making it impossible to accidentally create a global variable.

'use strict'

In normal JavaScript, a developer will not receive any error feedback assigning values to non-writable properties.

In strict mode, any assignment to a non-writable property, a getter-only property, a non-existing property, a non-existing variable, or a non-existing object, will throw an error.

Form Validation

شركة تحالف الإمارات للحلول التقنية ذ.م.م.
TAHALUF AL EMARAT TECHNICAL SOLUTIONS L.L.C.

Form validation normally used to occur at the server, after the client had entered all the necessary data and then pressed the Submit button.

If the data entered by a client was incorrect or was simply missing, the server would have to send all the data back to the client and request that the form be resubmitted with correct information.
This was really a lengthy process which used to put a lot of burden on the server.

Form Validation

JavaScript provides a way to validate form's data on the client's computer before sending it to the web server. Form validation generally performs two functions.

•**Basic Validation** − First of all, the form must be checked to make sure all the mandatory fields are filled in. It would require just a loop through each field in the form and check for data.

•**Data Format Validation** – Secondly, the data that is entered must be checked for correct form and value. Your code must include appropriate logic to test correctness of data.

Form Validation

شركة تحالف الإمارات للحلول التقنية ذ.م.م.
TAHALUF AL EMARAT TECHNICAL SOLUTIONS L.L.C.

Form validation is needed anytime you accept data from a user. This may include:

1. Validating the format of fields such as email address, phone number, zip code, name, password.
2. Validating mandatory fields.
3. Checking the type of data such as string vs number for fields such as social security number.
4. Ensuring that the value entered is a valid value such as country, date, and so on.

Form Validation

If a form field (fname) is empty, this function alerts a message, and returns false, to prevent the form from being submitted.

```
function validateForm() {
  var x =
   document.forms["myForm"]["fname"].value;
  if (x == "") {
    alert("Name must be filled out");
    return false;
  }
}
```

Form Validation

The function can be called when the form is submitted:

```
<form name="myForm" action="/action_page.php" onsubmit=
"return validateForm()" method="post">

    Name: <input type="text" name="fname">
    <input type="submit" value="Submit">

</form>
```

Form Validation

JavaScript is often used to validate numeric input.

```
<p>Please input a number between 1 and 10:</p>
<input id="numb">
<button type="button" onclick="myFunction()">Submit</button>
<p id="demo"></p>

<script>
    function myFunction() {
        var x, text;
        // Get the value of the input field with id="numb"
        x = document.getElementById("numb").value;

        // If x is Not a Number or less than one or greater than 10
        if (isNaN(x) || x < 1 || x > 10) {
            text = "Input not valid";
        } else {
            text = "Input OK";
        }
        document.getElementById("demo").innerHTML = text;
    }
</script>
```

Form Validation

HTML form validation can be performed automatically by the browser:
If a form field (fname) is empty, the required attribute prevents this form from being submitted.

```html
<form action="" method="post">
       <input type="text" name="fname" required>
       <input type="submit" value="Submit">
   </form>

   <p>
       If you click submit, without filling out the text field,
       your browser will display an error message.
   </p>
```

Form Validation

Data Validation

Data validation is the process of ensuring that user input is clean, correct, and useful.

Typical validation tasks are:

- has the user filled in all required fields?
- has the user entered a valid date?
- has the user entered text in a numeric field?

Form Validation

Most often, the purpose of data validation is to ensure correct user input.

Validation can be defined by many different methods, and deployed in many different ways.

Server side validation is performed by a web server, after input has been sent to the server.

Client side validation is performed by a web browser, before input is sent to a web server.

Form Validation Example

```
<div class="wrapper">
        <h2>Contact us</h2>
        <div id="error_message"></div>
        <form action="" id="myform" onsubmit="return validate();">
            <div class="input_field">
                <input type="text" placeholder="Name" id="name">
            </div>
            <div class="input_field">
                <input type="text" placeholder="Phone" id="phone">
            </div>
            <div class="input_field">
                <input type="text" placeholder="Email" id="email">
            </div>
            <div class="input_field">
                <textarea placeholder="Message" id="message"></textarea>
            </div>
            <div class="btn">
                <input type="submit">
            </div>
        </form>
    </div>
```

Form Validation Example

```
<script>
        function validate() {
            var name = document.getElementById("name").value;
            var phone = document.getElementById("phone").value;
            var email = document.getElementById("email").value;
            var message = document.getElementById("message").value;
            var error_message = 
    document.getElementById("error_message");

            error_message.style.padding = "10px";
```

Form Validation Example

```
var text;
           if (name.length < 5) {
               text = "Please Enter valid Name";
               error_message.innerHTML = text;
               return false;
           }
           if (isNaN(phone) || phone.length != 10) {
               text = "Please Enter valid Phone Number";
               error_message.innerHTML = text;
               return false;
           }
           if (email.indexOf("@") == -1 || email.length < 6) {
               text = "Please Enter valid Email";
               error_message.innerHTML = text;
               return false;
           }
           if (message.length <= 140) {
               text = "Please Enter More Than 140 Characters";
               error_message.innerHTML = text;
               return false;
           }
           alert("Form Submitted Successfully!");
           return true;
       }
   </script>
```

شركة تحالف الإمارات للحلول التقنية ذ.م.م.
TAHALUF AL EMARAT TECHNICAL SOLUTIONS L.L.C.

Day Five Task

On the E-Learning Portal