Html5 Forms and Form Fields

What is HTML <form>?

<form> is an HTML element to collect input data containing interactive controls. It provides facilities to input text, numbers, values, email, password, and control fields such as checkboxes, radio buttons, submit buttons, etc., or in other words, form is a container that contains input elements like text, email, number, radio buttons, checkboxes, submit buttons, etc. It also :

Facilitates user input collection through various elements.

Utilizes <form> tags to structure input elements.

Defines actions for data submission upon form completion.

Supports client-side validation for enhanced user experience.

These are the following HTML <form> elements:

<label>: It defines label for <form> elements.

<input>: It is used to get input data from the form in various types such as text, password, email, etc by changing its type.

<button>: It defines a clickable button to control other elements or execute a functionality.

<select>: It is used to create a drop-down list.

<textarea>: It is used to get input long text content.

<fieldset>: It is used to draw a box around other form elements and group the related data.

<legend>: It defines a caption for fieldset elements.

<**datalist>:** It is used to specify pre-defined list options for input controls.

<output>: It displays the output of performed calculations.

<option>: It is used to define options in a drop-down list.

<optgroup>: It is used to define group-related options in a drop-down list.

The <input> Element

One of the most used form elements is the `<input>` element.

The `<input>` element can be displayed in several ways, depending on the `type` attribute.

Example:

```
<label for="fname">First name:</label>

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

Html Input Types

Here are the different input types you can use in HTML:

- `<input type="button">`
- `<input type="checkbox">`
- `<input type="color">`
- `<input type="date">`
- `<input type="datetime-local">`
- `<input type="email">`
- `<input type="file">`
- `<input type="hidden">`
- `<input type="image">`
- `<input type="month">`

- `<input type="number">`
- `<input type="password">`
- `<input type="radio">`
- `<input type="range">`
- `<input type="reset">`
- `<input type="search">`
- `<input type="submit">`
- `<input type="tel">`
- `<input type="text">`
- `<input type="time">`
- `<input type="url">`
- `<input type="week">`

`<input type="text">` defines a single-line text input field

`<input type="password">` defines a password field

`<input type="submit">` defines a button for submitting form data to a form-handler.The form-handler is typically a server page with a script for processing input data.

The form-handler is specified in the form's `action` attribute

`<input type="reset">` defines a reset button that will reset all form values to their default values

`<input type="radio">` defines a radio button.

Radio buttons let a user select ONLY ONE of a limited number of choices

`<input type="checkbox">` defines a checkbox.

Checkboxes let a user select ZERO or MORE options of a limited number of choices.

`<input type="button">` defines a button

The `<input type="color">` is used for input fields that should contain a color.

Depending on browser support, a color picker can show up in the input field.

The `<input type="date">` is used for input fields that should contain a date.

You can also use the `min` and `max` attributes to add restrictions to dates

The `<input type="datetime-local">` specifies a date and time input field, with no time zone.

The `<input type="email">` is used for input fields that should contain an e-mail address.

The `<input type="image">` defines an image as a submit button.

The path to the image is specified in the `src` attribute.

The `<input type="hidden">` defines a hidden input field (not visible to a user).

A hidden field lets web developers include data that cannot be seen or modified by users when a form is submitted.

A hidden field often stores what database record that needs to be updated when the form is submitted.

Note: While the value is not displayed to the user in the page's content, it is visible (and can be edited) using any browser's developer tools or "View Source" functionality. Do not use hidden inputs as a form of security!

The `<input type="month">` allows the user to select a month and year.

The `<input type="number">` defines a numeric input field.

Example:

```
<form>

  <label for="quantity">Quantity:</label>

  <input type="number" id="quantity" name="quantity" min="0"
max="100" step="10" value="30">

</form>
```

The `<input type="range">` defines a control for entering a number whose exact value is not important (like a slider control). Default range is 0 to 100. However, you can set restrictions on what numbers are accepted with the `min`, `max`, and `step` attributes

```
<form>
  <label for="vol">Volume (between 0 and 50):</label>
  <input type="range" id="vol" name="vol" min="0" max="50">
</form>
```

The `<input type="search">` is used for search fields (a search field behaves like a regular text field).

```
<form>
  <label for="gsearch">Search Google:</label>
  <input type="search" id="gsearch" name="gsearch">
</form>
```

The `<input type="tel">` is used for input fields that should contain a telephone number.

The `<input type="time">` allows the user to select a time (no time zone).

```
<form>

  <label for="appt">Select a time:</label>

  <input type="time" id="appt" name="appt">

</form>
```

The `<input type="week">` allows the user to select a week and year.

```
<form>

  <label for="week">Select a week:</label>

  <input type="week" id="week" name="week">

</form>
```

Html Input Attributes

The value Attribute

The input `value` attribute specifies an initial value for an input field:

```
<form>
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Doe">
</form>
```

The readonly Attribute

The input `readonly` attribute specifies that an input field is read-only.

A read-only input field cannot be modified (however, a user can tab to it, highlight it, and copy the text from it).

The value of a read-only input field will be sent when submitting the form!

```
<form>
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John"
readonly><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Doe">
</form>
```

The disabled Attribute

The input `disabled` attribute specifies that an input field should be disabled.

A disabled input field is unusable and un-clickable.

The value of a disabled input field will not be sent when submitting the form!

```
<form>

  <label for="fname">First name:</label><br>

  <input type="text" id="fname" name="fname" value="John"
disabled><br>

  <label for="lname">Last name:</label><br>

  <input type="text" id="lname" name="lname" value="Doe">

</form>
```

The size Attribute

The input `size` attribute specifies the visible width, in characters, of an input field.

The default value for `size` is 20.

Note: The `size` attribute works with the following input types: text, search, tel, url, email, and password.

```
<form>
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" size="50"><br>
  <label for="pin">PIN:</label><br>
  <input type="text" id="pin" name="pin" size="4">
</form>
```

The maxlength Attribute

The input `maxlength` attribute specifies the maximum number of characters allowed in an input field.

Note: When a `maxlength` is set, the input field will not accept more than the specified number of characters. However, this attribute does not provide any feedback. So, if you want to alert the user, you must write JavaScript code.

```html
<form>

  <label for="fname">First name:</label><br>

  <input type="text" id="fname" name="fname" size="50"><br>

  <label for="pin">PIN:</label><br>

  <input type="text" id="pin" name="pin" maxlength="4" size="4">

</form>
```

What is the importance of Navigator Object in Javascript?

The JavaScript navigator object is used for browser detection, it can be used to get browser information such as appName, appCodeName, userAgent, etc. Navigator object is a window property so it can be accessed by the "**window.navigator**" or "**navigator**".

It can also check whether the browser is in online mode or not and whether Java is enabled or not in our system.

There are many properties of navigator objects that return information about the browser following are some of those –

appName – Returns the name
appVersion – Returns the version
appCodeName – Returns the code name
cookieEnabled – Returns true if the cookie is enabled otherwise false
userAgent – Returns the user agent
language – Returns the language. It is supported in Netscape and Firefox only.

userLanguage − Returns the user language. It is supported in IE only.

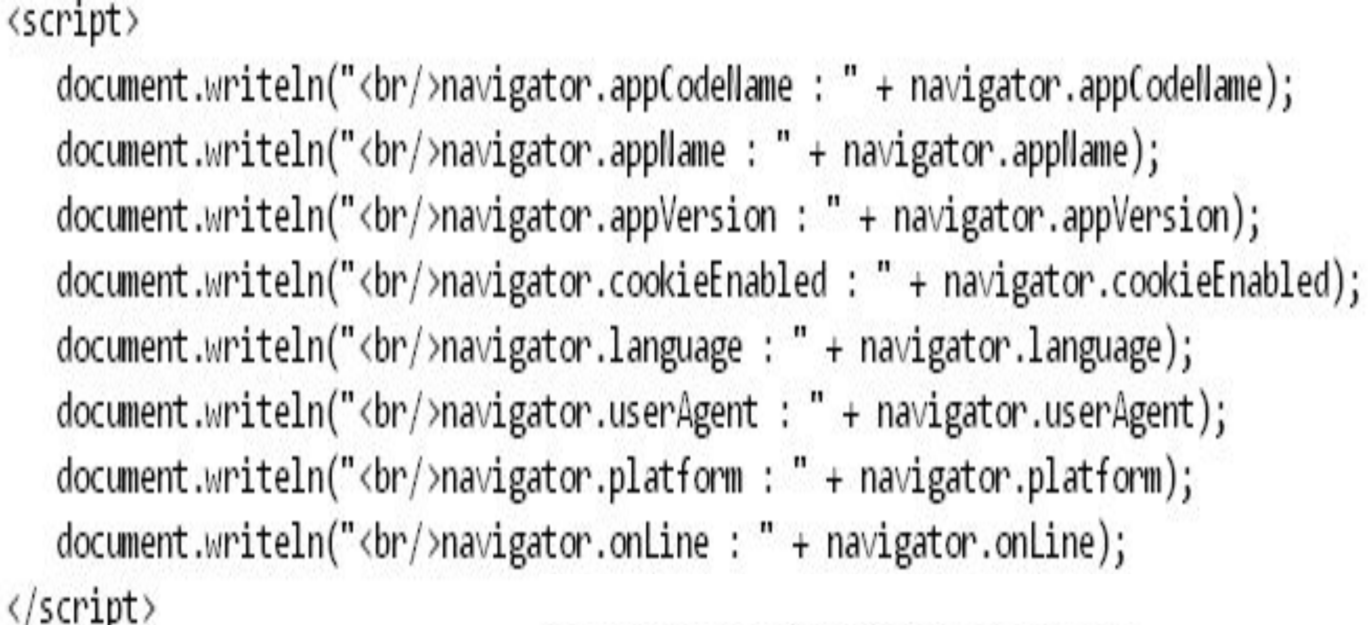
plugins − Returns the plugins. It is supported in Netscape and Firefox only.

systemLanguage − Returns the system language. It is supported in IE only.

mimeTypes[] − Returns the array of mime types. It is supported in Netscape and Firefox only.

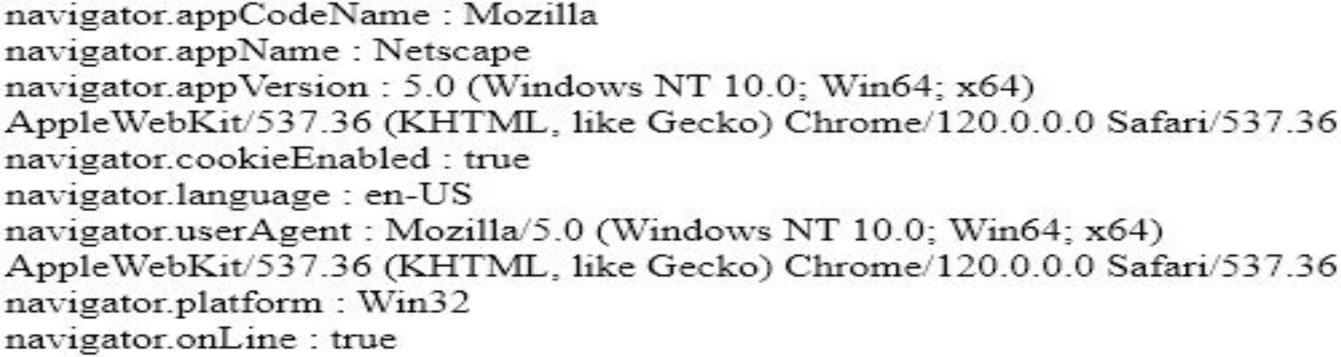
platform − Returns the platform e.g. Win32.

online − Returns true if the browser is online otherwise false.



```
<script>
   document.writeln("<br/>navigator.appCodeName : " + navigator.appCodeName);
   document.writeln("<br/>navigator.appName : " + navigator.appName);
   document.writeln("<br/>navigator.appVersion : " + navigator.appVersion);
   document.writeln("<br/>navigator.cookieEnabled : " + navigator.cookieEnabled);
   document.writeln("<br/>navigator.language : " + navigator.language);
   document.writeln("<br/>navigator.userAgent : " + navigator.userAgent);
   document.writeln("<br/>navigator.platform : " + navigator.platform);
   document.writeln("<br/>navigator.onLine : " + navigator.onLine);
</script>
```

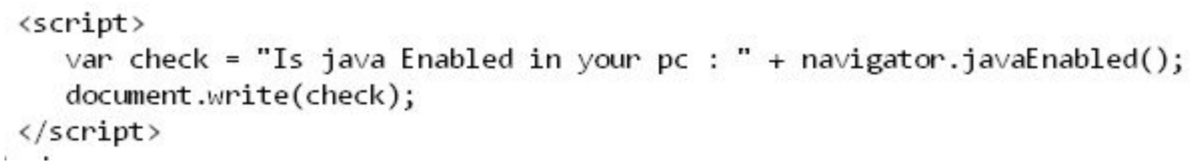
ıot. If you want to check



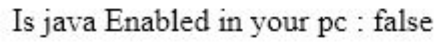
```
navigator.appCodeName : Mozilla
navigator.appName : Netscape
navigator.appVersion : 5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36
navigator.cookieEnabled : true
navigator.language : en-US
navigator.userAgent : Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36
navigator.platform : Win32
navigator.onLine : true
```

Example 2

In the following example, we are checking the method if we have installed the java or not in our pc. But in our pc java is not installed i.e. the output is false.



```
<script>
   var check = "Is java Enabled in your pc : " + navigator.javaEnabled();
   document.write(check);
</script>
```

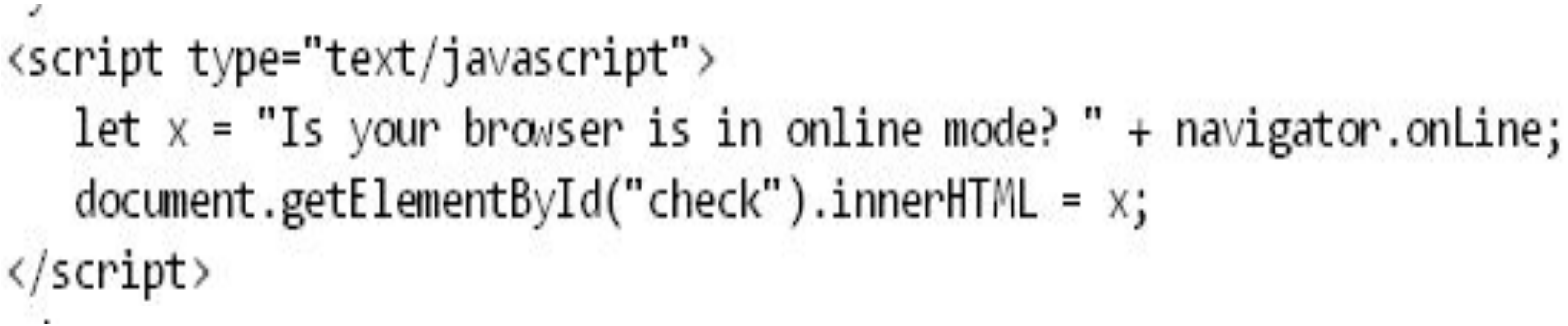


```
Is java Enabled in your pc : false
```

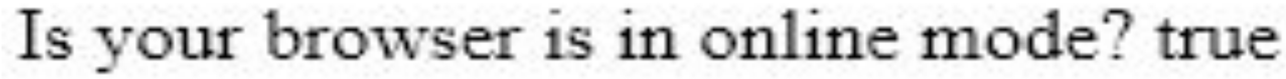
Example 3

The following example demonstrates the importance of the navigator object in JavaScript.

In this case, we will use the navigator object to obtain browser information to check if your browser is in online mode. If so then it displays true if it is in online mode else returns false and it displays the necessary browser information.



```
<script type="text/javascript">
   let x = "Is your browser is in online mode? " + navigator.onLine;
   document.getElementById("check").innerHTML = x;
</script>
```



```
Is your browser is in online mode? true
```