

JavaScript

Forms

Form Validation

Cookies

Vijesh M. Nair
Assistant Professor
Dept. of CSE (AI-ML)

What JavaScript can do

- ⊕ Control document appearance and content
- ⊕ Control the browser
- ⊕ Interact with user
- ⊕ Interact with forms
 - ⊕ Validate user input
- ⊕ Use cookies

HTML Forms

- ⊕ `<form>` is a block-level element in the body of your HTML page
- ⊕ A form element can contain text input boxes, buttons, checkboxes, pull-down menus and images
- ⊕ Your document can have multiple form elements
- ⊕ A special button called Submit is used to send the form data to the server

JavaScript and Forms

- ⊕ With a server-side program, an HTML form needs a submit button
- ⊕ With JavaScript, you can react to events of individual input elements.
- ⊕ For some types of tasks, you may never need to submit at all.

Form Objects

- ⊕ An HTML form is represented in JavaScript by a Form object
 - ⊕ `document.forms[]` is an array containing all the forms in the document
- ⊕ Each form object has an `elements[]` attribute which is an array of input elements
 - `document.forms[i].elements[j]`
- ⊕ A name attribute in the tag allows you to access the elements by name

Form Objects

- ⊕ Form methods include `submit()` and `reset()`
- ⊕ Submit and Reset buttons trigger `onsubmit` and `onreset` respectively
 - ⊕ `onsubmit` and `onreset` handlers invoked just before corresponding method is called
 - ⊕ If a handler returns false, the corresponding `submit()` or `reset()` method will not be invoked

Form Elements

- ⊕ Use the `<input>` tag to create controls
 - ⊕ `type="input-type"` needed to specify which type of control
 - ⊕ `name` attribute needed to identify the element
- ⊕ There are special tags for `<button>`, `<select>` and `<textarea>`

Form Elements and Events

Object	HTML	Events
Button	<code><input type="button"></code> <code><button></code>	onclick
Checkbox	<code><input type="checkbox"></code>	onclick
Radio	<code><input type="radio"></code>	onclick
Select	<code><select></code> <code><select multiple></code>	onchange
Option - belongs to Select object		
Text	<code><input type="text"></code>	onchange
Password	<code>(input type="password"></code>	onchange
Textarea	<code><textarea></code>	onchange

Form Elements and Events

Object	HTML	Events
FileUpload	<input type="file">	onchange
Hidden	<input type="hidden">	none
Reset	<input type="reset"> <button type="reset">	onclick
Submit	<input type="submit"> <button type="submit">	onclick

Names in Forms

- ⊕ If a form is to be submitted to a server side program, the name attribute of every element must be set
- ⊕ Within the javascript code, you can use the name of a form or form element instead of indexing into the corresponding array
 - ⊕ For the form defined by
 - ⊕ `<form name="everything">...</form>`
 - ⊕ `document.everything` and `document.forms[0]` are equivalent ways to refer to the form

Properties of Form Elements

- ⊕ type - read only
- ⊕ form - read only reference to form element is part of
- ⊕ name - name that can be used to refer to the element
- ⊕ value - string sent to web server on submission
 - ⊕ text entered by user for text and textarea
 - ⊕ text displayed on a button
 - ⊕ string set in HTML code for radio and checkbox

Event Handlers for Form Elements

- ⊕ `onclick` - triggered by mouse click
- ⊕ `onchange` - triggered by changing a value
- ⊕ `onfocus` - triggered when element receives focus
- ⊕ `onblur` - triggered when element loses focus

Validation

- ⊕ Check that all required fields have data

```
e.value==null || e.value=="
```

- ⊕ Check the format of fields that contain information like email addresses, URLs, phone numbers, ...

- ⊕ Use the RegExp class to create patterns

```
var pattern = /(\d{3}) \d{3}-\d{4}/
```

- ⊕ Use an alert to display error messages

Cookies

- ⊕ A cookie is a small amount of named data stored by the browser and associated with a particular web site
- ⊕ provide a way of saving state of a web page
 - ⊕ last for current session by default
- ⊕ Server-side programs use cookies
- ⊕ JavaScript can manipulate cookie data

Cookie Attributes

- ⊕ `name` - each cookie has a name
- ⊕ `value` - value associated with the cookie
- ⊕ `expires` - cookies are transient unless this is set
- ⊕ `path` - web page(s) with which cookie is associated
- ⊕ `domain` - allows cookie to be available to different web servers
- ⊕ `secure` - Boolean attribute that specifies whether cookies must be transmitted over a secure (https) connection

Storing Cookies

⊕ Creating a transient cookie

```
document.cookie = "version="
    + escape( document.lastModified);
```

⊕ escape encodes characters that aren't allowed in the cookie

⊕ Storing other attributes with a cookie

⊕ Use semicolon to separate attribute=value pairs

```
document.cookie = "version="
    + escape( document.lastModified)
    + "; expires=" + nextYear.toGMTString();
```


Reading Cookies

- ⊕ `document.cookie` returns a string containing all the cookies for the document.
- ⊕ `name=value` pairs separated by semicolons
- ⊕ value does not include attributes

Thank You..!