UTM
UNIVERSITI TEKNOLOGI MALAYSIA

Online Learning

Javascript - DOM Form

Sarina Sulaiman

Faculty of Computing

JS: DOM Form

- Form Object
  - The Form object represents an HTML form.
  - For each instance of a <form> tag in an HTML document, a Form object is created.
  - Main form methods
    - document.forms[0].submit()
    - document.forms[0].reset()
    - document.forms[0].action = 'register.php';

JS: DOM Form

- document.forms[number].elements[number]
- Every <input>, <select> and <textarea> is form element.

JS: DOM Form

- <form name="personal" action="something.pl" onsubmit="return checkscript()"> <input type=text size=20 name=name>
- <input type=text size=20 name=address>
- <input type=text size=20 name=city>
- </form>
- Now you can access these elements by:
  - document.personal.name
  - document.personal.address
  - document.personal.city

JS: DOM Form

- Getting data from Texts, textareas and hidden fields
  - user_input = document.forms[0].*text*.value
  - where *text* is the name of the text field, textarea or hidden field.
  - The value of this element gives the text, so we transfer it to user_input.
- Setting value
  - document.forms[0].text.value = 'The new value';

JS: DOM Form

- Select boxes
  - Select boxes are simple too:
    - user_input = document.forms[0].select.value;
  - To change the selected option in a select box, you have to change its selectedIndex,
    - document.forms[0].select.selectedIndex = 2;

JS: DOM Form

- Checkboxes
  - Checkboxes need a slightly different approach
  - We already know their values, but want to know whether the user has checked them.
  - The checked property tells us.
  - It can have two values: true or false.

JS: DOM Form

- if (document.forms[0].checkbox.checked)
  {
  - user_input = document.forms[0].checkbox.name

  }
- To check a checkbox, set its property checked to true:
  - document.forms[0].checkbox.checked = true

JS: DOM Form

- Radio buttons
  - Unfortunately it's not possible to see at once which radio button in a group the user has checked.
  - You need to go through all radio's and see which one's checked property is true.

JS: DOM Form

- Radio buttons
- for (i=0;i<document.forms[0].radios.length;i++)
  - { if (document.forms[0].radios[i].checked)
    {
      - user_input =
        document.forms[0].radios[i].value;

    }
  - }

JS: DOM Form

- Radio buttons
  - where radios is the name of the group of radio buttons.
  - Note that document.forms[0].radios is an array filled with all radio buttons.
  - Loop through all of them and see if it is checked.
  - If one is, transfer the value of that radio button to user_input.

JS: DOM Form

- Using elements[]
  - var form=document.getElementById("myForm");
  - form.elements[0].value;
  - form.elements[0].value = “new value”;

JS: Event Handling

- The building blocks of an interactive web page is the Javascript event system.
- An event in Javascript is something that happens with or on the web page. A few example of events:
  - A mouse click
  - The web page loading
  - Mousing over a hot spot on the web page, also known as hovering
  - Selecting an input box in an HTML form
  - A keystroke

JS: Event Handling

- Different occurrences generate different types of events.
- When the user moves the mouse over a hyperlink, it causes a different type of event than when the user clicks the mouse on the hyperlink.
- Even the same occurrence can generate different types of events based on context:
  - when the user clicks the mouse over a Submit button,
  - for example, it generates a different event than when the user clicks the mouse over the Reset button of a form.

JS: Event Handlers & HTML element

- Note: Events are normally used in combination with functions, and the function will not be executed before the event occurs!
- onload and onUnload
  - The onload and onUnload events are triggered when the user enters or leaves the page.
  - The onload event is often used to check the visitor's browser type and browser version, and load the proper version of the web page based on the information.
  - Both the onload and onUnload events are also often used to deal with cookies that should be set when a user enters or leaves a page.
  - For example, you could have a popup asking for the user's name upon his first arrival to your page.
  - The name is then stored in a cookie. Next time the visitor arrives at your page, you could have another popup saying something like: "Welcome John Doe!".

JS: Event Handlers & HTML element

- onFocus, onBlur and onChange
  - The onFocus, onBlur and onChange events are often used in combination with validation of form fields.
  - Below is an example of how to use the onChange event. The checkEmail() function will be called whenever the user changes the content of the field:
    - <input type="text" size="30" id="email" onchange="checkEmail()">;

JS: Event Handlers & HTML element

- onSubmit
  - The onSubmit event is used to validate ALL form fields before submitting it.
  - The checkForm() function will be called when the user clicks the submit button in the form.
  - If the field values are not accepted, the submit should be cancelled.
  - The function checkForm() returns either true or false.
  - If it returns true the form will be submitted, otherwise the submit will be cancelled.
    - <form method="post" action="xxx.htm" onsubmit="return checkForm()">

JS: Event Handlers & HTML element – Activity 14.2

- onMouseOver and onMouseOut
  - onMouseOver and onMouseOut are often used to create "animated" buttons.
  - onMouseOver animate when a button is click such as in a:hover
  - onMouseOut animate when a button is normal state

JS: Form verification

- Activity 14.1

Online Learning

UTM
UNIVERSITI TEKNOLOGI MALAYSIA

THANK YOU