Univerzitet Singidunum

Web programming

www.singidunum.ac.rs

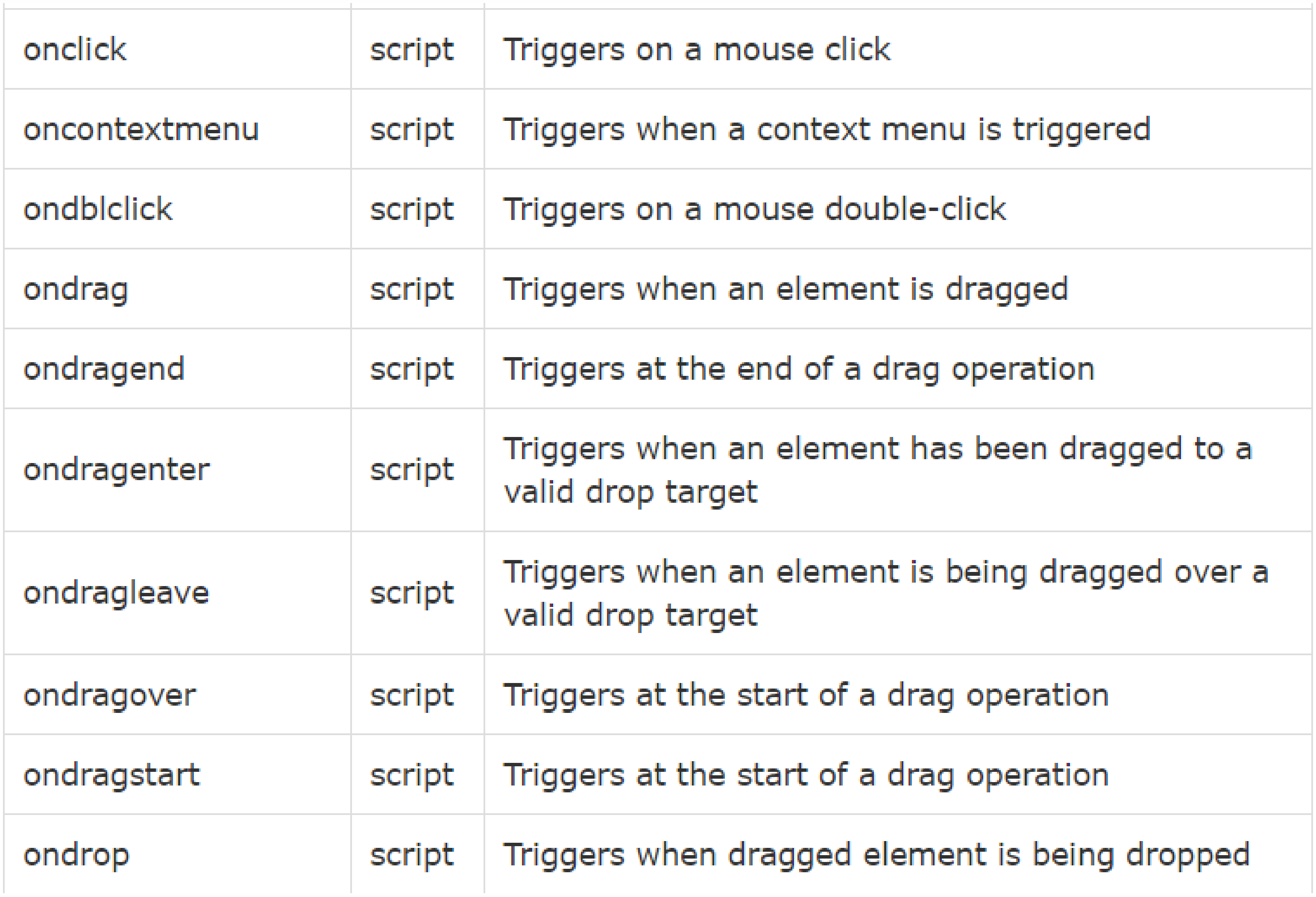
JavaScript Events

- JavaScript's interaction with HTML is handled through events that occur when the user or the browser manipulates a page.
- When the page loads, it is called an event. When the user clicks a button, that click too is an event. Other examples include events like pressing any key, closing a window, resizing a window, etc.
- Developers can use these events to execute JavaScript coded responses, which cause buttons to close windows, messages to be displayed to users, data to be validated, and virtually any other type of response imaginable.
- Events are a part of the Document Object Model (DOM) Level 3 and every HTML element contains a set of events which can trigger JavaScript Code.

JavaScript Events

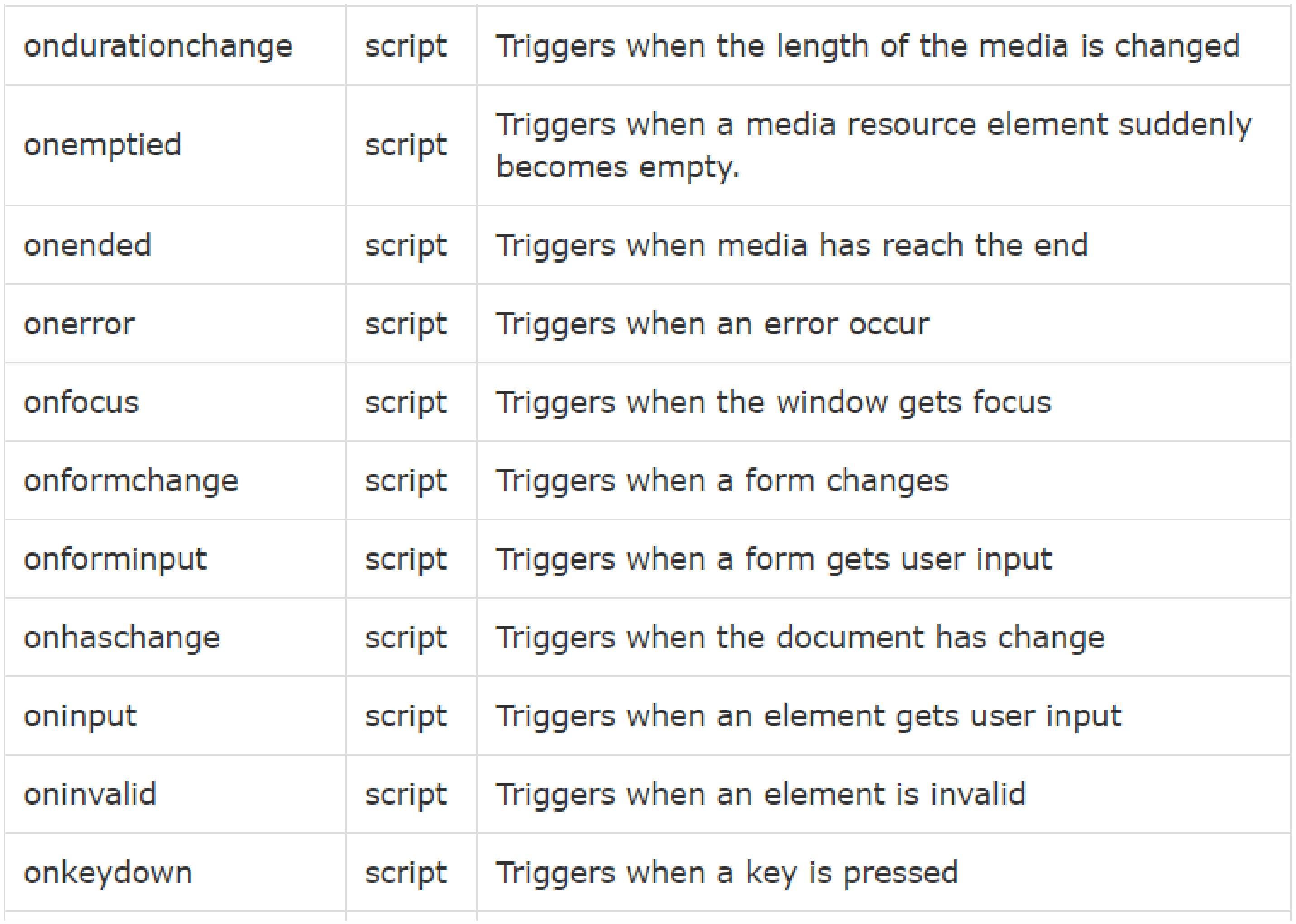
Attribute	Value	Description
Offline	script	Triggers when the document goes offline
Onabort	script	Triggers on an abort event
onafterprint	script	Triggers after the document is printed
onbeforeonload	script	Triggers before the document loads
onbeforeprint	script	Triggers before the document is printed
onblur	script	Triggers when the window loses focus
oncanplay	script	Triggers when media can start play, but might has to stop for buffering
oncanplaythrough	script	Triggers when media can be played to the end, without stopping for buffering
onchange	script	Triggers when an element changes

JavaScript Events



onclick	script	Triggers on a mouse click
oncontextmenu	script	Triggers when a context menu is triggered
ondblclick	script	Triggers on a mouse double-click
ondrag	script	Triggers when an element is dragged
ondragend	script	Triggers at the end of a drag operation
ondragenter	script	Triggers when an element has been dragged to a valid drop target
ondragleave	script	Triggers when an element is being dragged over a valid drop target
ondragover	script	Triggers at the start of a drag operation
ondragstart	script	Triggers at the start of a drag operation
ondrop	script	Triggers when dragged element is being dropped

JavaScript Events



ondurationchange	script	Triggers when the length of the media is changed
onemptied	script	Triggers when a media resource element suddenly becomes empty.
onended	script	Triggers when media has reach the end
onerror	script	Triggers when an error occur
onfocus	script	Triggers when the window gets focus
onformchange	script	Triggers when a form changes
onforminput	script	Triggers when a form gets user input
onhaschange	script	Triggers when the document has change
oninput	script	Triggers when an element gets user input
oninvalid	script	Triggers when an element is invalid
onkeydown	script	Triggers when a key is pressed

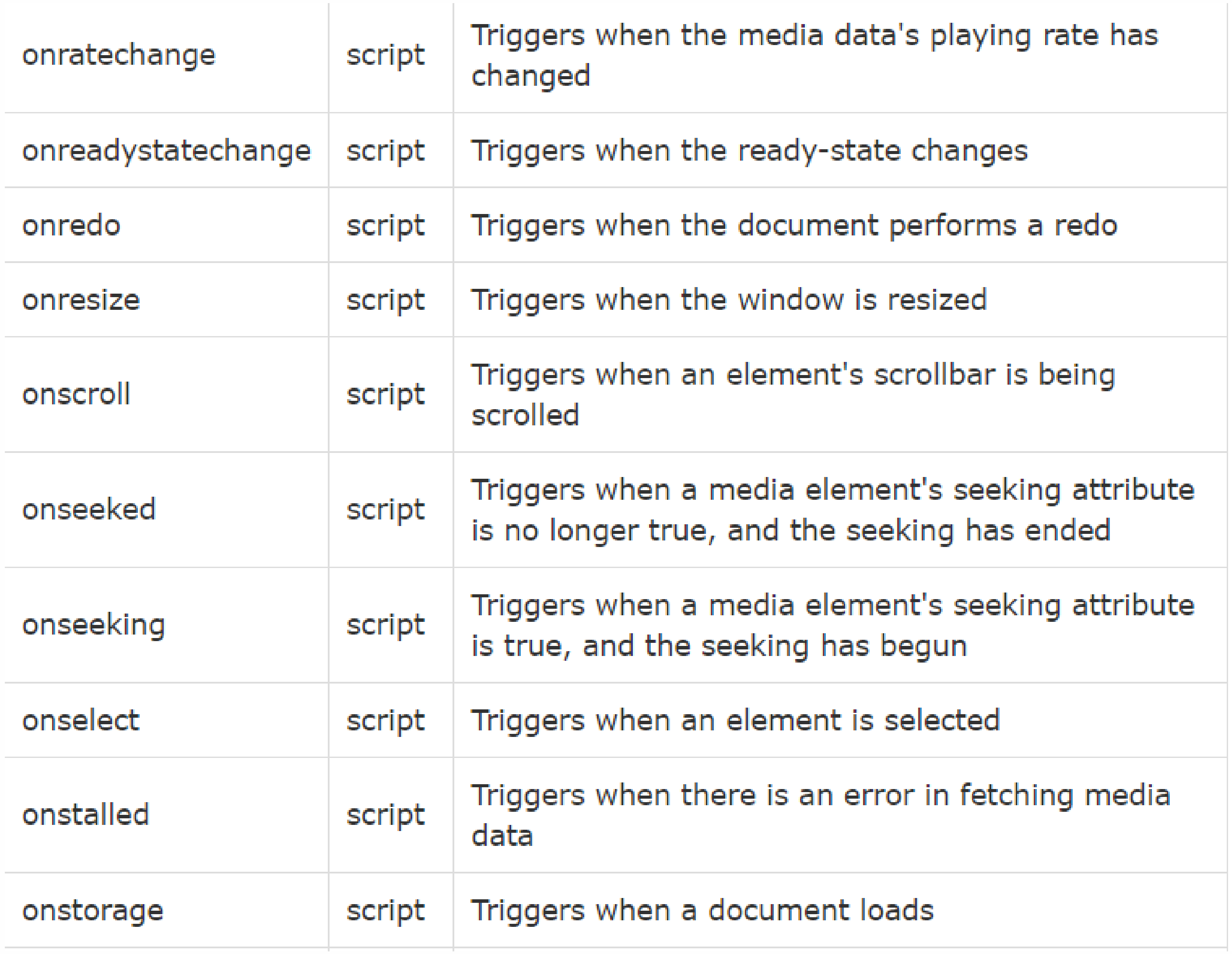
JavaScript Events

onkeypress	script	Triggers when a key is pressed and released
onkeyup	script	Triggers when a key is released
onload	script	Triggers when the document loads
onloadeddata	script	Triggers when media data is loaded
onloadedmetadata	script	Triggers when the duration and other media data of a media element is loaded
onloadstart	script	Triggers when the browser starts to load the media data
onmessage	script	Triggers when the message is triggered
onmousedown	script	Triggers when a mouse button is pressed
onmousemove	script	Triggers when the mouse pointer moves
onmouseout	script	Triggers when the mouse pointer moves out of an element
onmouseover	script	Triggers when the mouse pointer moves over an element

JavaScript Events

onmouseup	script	Triggers when a mouse button is released
onmousewheel	script	Triggers when the mouse wheel is being rotated
onoffline	script	Triggers when the document goes offline
onoine	script	Triggers when the document comes online
ononline	script	Triggers when the document comes online
onpagehide	script	Triggers when the window is hidden
onpageshow	script	Triggers when the window becomes visible
onpause	script	Triggers when media data is paused
onplay	script	Triggers when media data is going to start playing
onplaying	script	Triggers when media data has start playing
onpopstate	script	Triggers when the window's history changes
onprogress	script	Triggers when the browser is fetching the media data

JavaScript Events



onratechange	script	Triggers when the media data's playing rate has changed
onreadystatechange	script	Triggers when the ready-state changes
onredo	script	Triggers when the document performs a redo
onresize	script	Triggers when the window is resized
onscroll	script	Triggers when an element's scrollbar is being scrolled
onseeked	script	Triggers when a media element's seeking attribute is no longer true, and the seeking has ended
onseeking	script	Triggers when a media element's seeking attribute is true, and the seeking has begun
onselect	script	Triggers when an element is selected
onstalled	script	Triggers when there is an error in fetching media data
onstorage	script	Triggers when a document loads

JavaScript Events

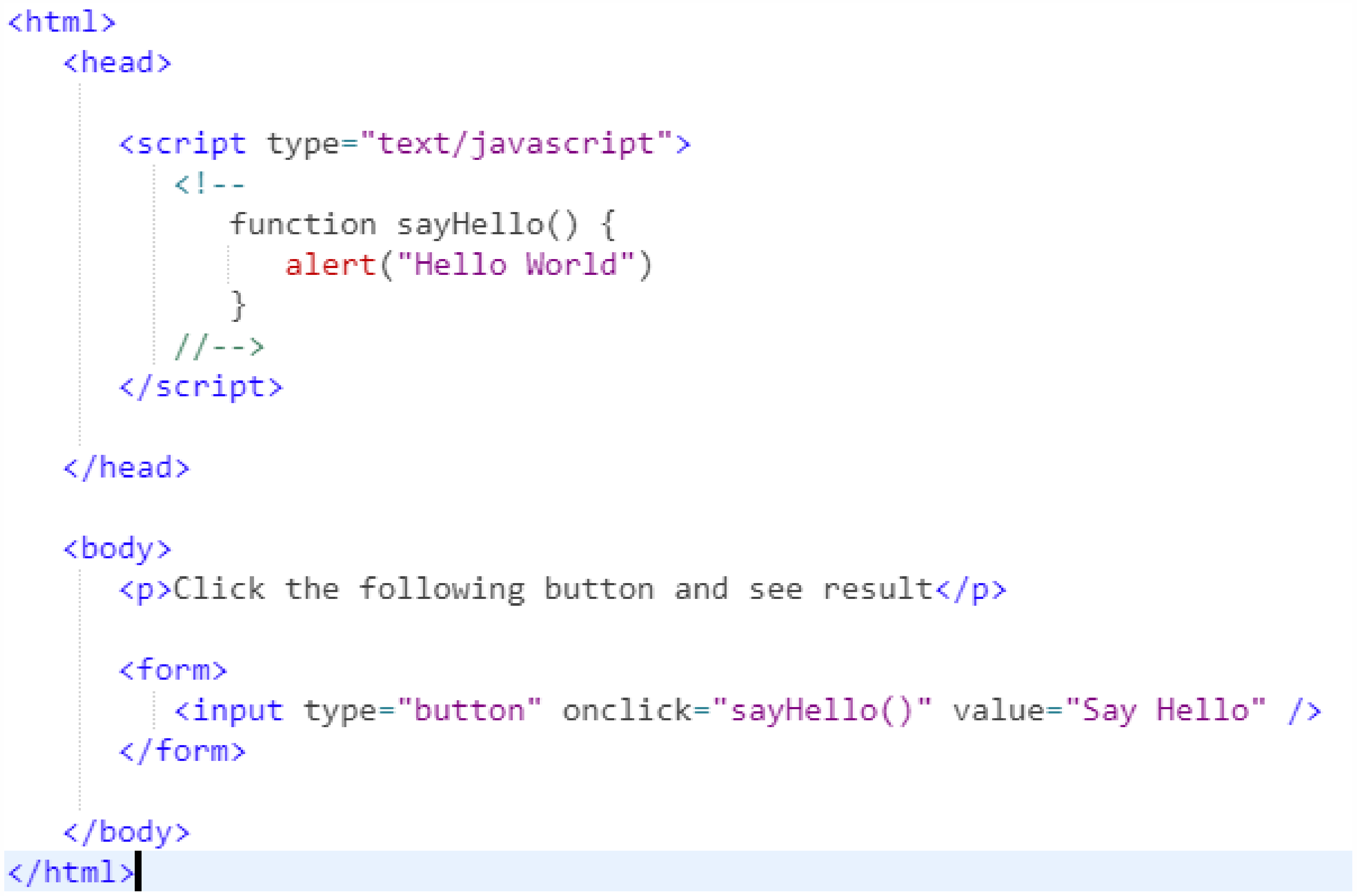
onsubmit	script	Triggers when a form is submitted
onsuspend	script	Triggers when the browser has been fetching media data, but stopped before the entire media file was fetched
ontimeupdate	script	Triggers when media changes its playing position
onundo	script	Triggers when a document performs an undo
onunload	script	Triggers when the user leaves the document
onvolumechange	script	Triggers when media changes the volume, also when volume is set to "mute"
onwaiting	script	Triggers when media has stopped playing, but is expected to resume

Univerzitet Singidunum

JavaScript Events

- **onclick Event Type**
- This is the most frequently used event type which occurs when a user clicks the left button of his mouse.
- You can put your validation, warning etc., against this event type.

JavaScript Events



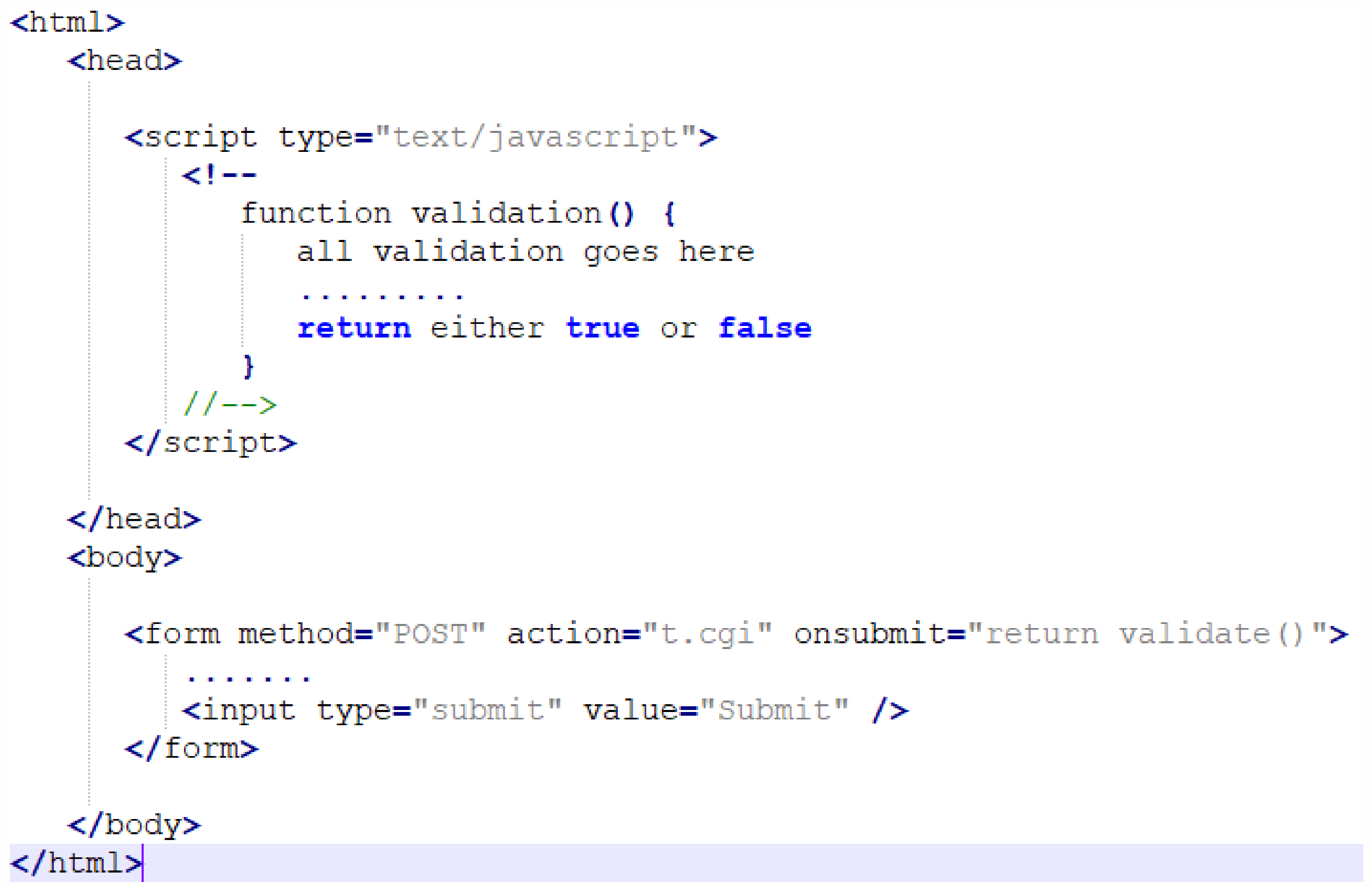
```
<html>
   <head>

      <script type="text/javascript">
         <!--
            function sayHello() {
               alert("Hello World")
            }
         //-->
      </script>

   </head>

   <body>
      <p>Click the following button and see result</p>

      <form>
         <input type="button" onclick="sayHello()" value="Say Hello" />
      </form>

   </body>
</html>
```

JavaScript Events

- **onsubmit** is an event that occurs when you try to submit a form. You can put your form validation against this event type.
- The following example shows how to use onsubmit.
- Here we are calling a **validate()** function before submitting a form data to the webserver. If **validate()** function returns true, the form will be submitted, otherwise it will not submit the data.

Univerzitet Singidunum

JavaScript Events



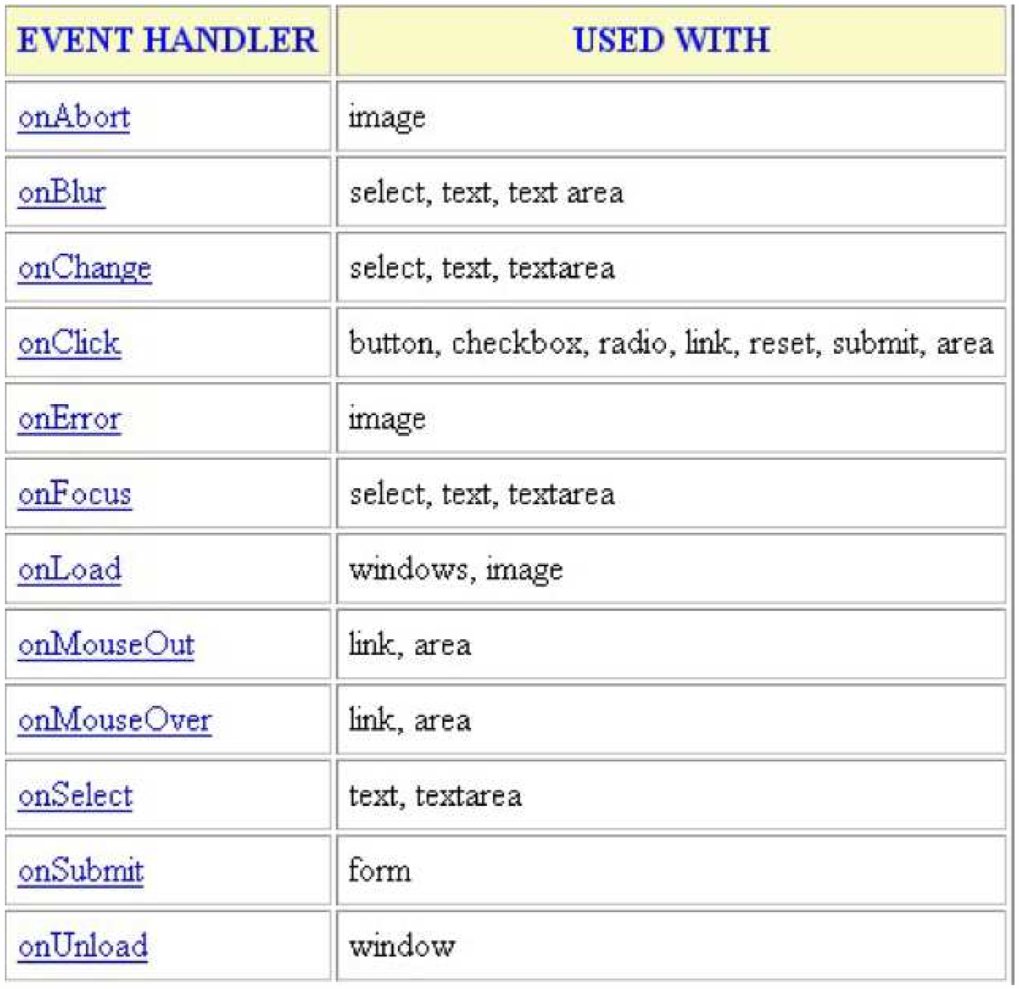
```
<html>
   <head>

      <script type="text/javascript">
         <!--
            function validation() {
               all validation goes here
               .........
               return either true or false
            }
         //-->
      </script>

   </head>
   <body>

      <form method="POST" action="t.cgi" onsubmit="return validate()">
         .......
         <input type="submit" value="Submit" />
      </form>

   </body>
</html>
```

JavaScript Events

- **onmouseover and onmouseout**
- These two event types will help you create nice effects with images or even with text as well.
- The **onmouseover** event triggers when you bring your mouse over any element and the **onmouseout** triggers when you move your mouse out from that element.

JavaScript Events

- Events and tags to which they can be assigned



EVENT HANDLER	USED WITH
onAbort	image
onBlur	select, text, text area
onChange	select, text, textarea
onClick	button, checkbox, radio, link, reset, submit, area
onError	image
onFocus	select, text, textarea
onLoad	windows, image
onMouseOut	link, area
onMouseOver	link, area
onSelect	text, textarea
onSubmit	form
onUnload	window

- <A>
  - click (onClick)
  - mouseOver (onMouseOver)
  - mouseOut (onMouseOut)
- <AREA>
  - mouseOver (onMouseOver)
  - mouseOut (onMouseOut)
- <BODY>
  - blur (onBlur)
  - error (onError)
  - focus (onFocus)
  - load (onLoad)
  - unload (onUnload)
- <FORM>
  - submit (onSubmit)
  - reset (onReset
- <IMG>
  - abort (onAbort)
  - error (onError)
  - load (onLoad)

- <INPUT TYPE = "button">
  - click (onClick)
- <INPUT TYPE = "checkbox">
  - click (onClick)
- <INPUT TYPE = "reset">
  - click (onClick)
- <INPUT TYPE = "submit">
  - click (onClick)
- <INPUT TYPE = "text">
  - blur (onBlur)
  - focus (onFocus)
  - change (onChange)
  - select (onSelect)
- <SELECT>
  - blur (onBlur)
  - focus (onFocus)
  - change (onChange)
- <TEXTAREA>
  - blur (onBlur)
  - focus (onFocus)
  - change (onChange)
  - select (onSelect)

What is Event Handling?

- Capturing events and responding to them
- The system sends events to the program and the program responds to them as they arrive
- Events can include things a user does - like clicking the mouse - or things that the system itself does - like updating the clock

Event Driven Programs

- Programs that can capture and respond to events are called ‘event-driven programs’
- JavaScript was specifically designed for writing such programs
- Almost all programs written in JavaScript are event-driven

JavaScript Handling of Events

- Events handlers are placed in the BODY part of a Web page as attributes in HTML tags
- Events can be captured and responded to directly with JavaScript one-liners embedded in HTML tags in the BODY portion
- Alternatively, events can be captured in the HTML code, and then directed to a JavaScript function for an appropriate response