

Events

- Event : Notification that something has occurred
- Example situations that make the web browser generate an event
 - Browser finishes loading a document
 - When the user clicks on a button
 - When the user moves the mouse
 - Others
- **Event handler** (also known as **event listener**)
 - JavaScript function or code fragment that is executed when a particular event occurs
- Event handler registration
 - **Associating an event handler with a particular event**

Event-driven Programming

- **Normal (control flow-based) programming**
 - Approach
 - » Start at main()
 - » Continue until end of program or exit()
- **Event-driven programming**
 - Start at main()
 - Register event handlers
 - Await events & perform associated computation
- **GUIs (Graphical User Interfaces)**
 - Example of event-driven software

Event Handler Attributes for Most HTML

- Mouse Related
 - **onclick** - mouse button is pressed and released
 - **ondblclick** - mouse button is double-click over element
 - **onmousedown** - mouse is pressed down while cursor is over the element
 - **onmouseup** - mouse is released while the cursor is over the element
 - **onmouseenter** - mouse moves onto element
 - **onmouseover** - mouse pointer enters into an element and its child elements
 - **onmouseout** - mouse moves off element
 - **onmousemove** - mouse pointer is moved over an element

Accessing Data From Text Fields

- We can access data in text fields by first accessing the element using:
 - **document.getElementById(elementId)**
 - **document.querySelector(**#**elementId)**
- We can then access the value using **value**
- **Retrieving value of text field**

```
let login = document.getElementById("loginId").value;
```

or

```
let login = document.querySelector("#myTextField").value
```

Associating Function with Event

- We can define a function (callback) for an event associated with an element by first accessing the element using:
 - **`document.getElementById(elementId)` or `document.querySelector(#elementId)`**and then assigning a call back function to the event
- **Defining which function to call when an element (e.g., button) is clicked on**
`document.getElementById("processButton").onclick = callback;`
- **Another way to associate a function is to use `addEventListener`**
 - Allows several events to be added
`document.getElementById("displayValueButton").addEventListener("click", callback);`
- **Another way is to set the `onclick` property in the element**
 - `<input type="button" value="Display School Name" onclick="displaySchoolName()" />`
- **Example:** AssociateButtonWithFunction.html
- **Example:** GetValueInTextField.html, UpdateValueInTextfield.html

Form Data Access (Attributes)

- **We can access/modify attributes using `getAttribute()/setAttribute()`**

```
let imageElement = document.getElementById("myImage");  
let imageName = imageElement.getAttribute("src");  
imageElement.setAttribute("src", "imageFile.jpg");
```
- **You can access and modify the attribute directly**

```
alert(document.querySelector("#myImage").src)  
document.querySelector("#myImage").src = "testudo1.jpg"
```
- **Example:** GetSetAttribute.html, GetSetAttribute2.html

Animation Using setInterval/clearInterval

- You can execute code at a particular interval using **setInterval**
- **setInterval** returns an id that **clearInterval** uses to stop execution
- **Example:** Animation.html

Modifying a Page Area Using innerHTML

- `document.writeln()` - replaces the whole page after a page has been rendered. What if you want to update an area of the current page?
- **Example:** `InnerHTML.html`

Loading JavaScript from a File

- **Example:** LoadingJSFromFile.html