## Ziel

Die Lernenden wissen, wie mit JavaScript auf Benutzerinteraktionen reagiert werden kann.

Die Lernenden wenden die Registrierung von Eventhandlern mithilfe der Unterlagen korrekt an.

## Ausgangslage

Über Funktionen wissen Sie nun schon fast alles, was es zu wissen gibt. Aber noch immer fehlte etwas: **Interaktion!** Der Einsatz von JavaScript wird erst dann richtig sinnvoll und spannend, wenn die Programme auf Benutzereingaben oder Mausbewegungen reagieren können. Dies wird ermöglicht durch die sogenannten *Eventhandler* (deutsch: Ereignisbehandlung). Ihre HTML Seiten werden nun so richtig dynamisch und lebendig…

## Hilfsmittel

Buch „Schrödinger lernt HMTL5, CSS3 und JavaScript“, Kapitel 11.

Internet.

## Vorgehensweise

1. Lesen Sie im Buch "Schrödinger" im Kapitel 12 die Seiten 482 bis 488.
2. Beantworten Sie die Theorie-Fragen
3. Lösen Sie die praktischen Aufgaben.

# Theorie-Fragen

1. Was ist ein Eventhandler? Erklären sie in eigenen Worten.

Antwort: Ein Event – Handler stellt eine Verknüpfung zu einer Script-Sprache her.

-------------------------------------------------------------------------------------

-------------------------------------------------------------------------------------

-------------------------------------------------------------------------------------

-------------------------------------------------------------------------------------

1. Auf Seite 486 im Buch "Schrödinger" wird erwähnt, dass der Internet Explorer die Methode addEventListener() erst seit der Version 9 kennt. In allen älteren Versionen des IE muss dagegen die Methode attachEvent()verwendet werden, um Events zu registrieren. Deshalb wird auf Seite 487 eine JavaScript-Objektabfrage verwendet, um zu prüfen, ob ein Browser die addEventListener() Methode unterstützt oder nicht.

var node = document.getElementById("myButton");  
if (node.addEventListener) {  
 //...   
}   
else if (node.attachEvent) {  
 //...  
}

Sie haben gelernt, Browserweichen zu erstellen durch Abfrage der Eigenschaft navigator.userAgent. Was denken Sie, weshalb wird hier nicht mit einer Browserweiche versucht zu bestimmen, ob der Browser eine ältere IE Version ist und damit attachEvent() statt addEventListener()verwendet werden muss?

Antwort:

-------------------------------------------------------------------------------------

-------------------------------------------------------------------------------------

-------------------------------------------------------------------------------------

-------------------------------------------------------------------------------------

-------------------------------------------------------------------------------------

1. Recherchieren Sie im Internet, wie stark IE 8, IE 7, IE 6 und IE 5 noch verbreitet sind, weltweit und in der Schweiz, in den letzten 6 Monaten.

|  |  |  |
| --- | --- | --- |
|  | **Verbreitung Welt   in %** | **Verbreitung CH in %** |
| **IE 8** |  |  |
| **IE 7** |  |  |
| **IE 6** |  |  |
| **IE 5** |  |  |

Ist es Ihrer Meinung nach weiterhin notwendig, IE Versionen älter als 9 zu unterstützen und jedes Mal, wenn ein Event registriert werden muss, mühsam eine JavaScript-Objektabfrage zu programmieren? Werden Sie in den Websites, die Sie in Zukunft programmieren, weiterhin mit attachEvent()diese alten IE Versionen unterstützen oder werden Sie darauf verzichten? Bitte begründen Sie Ihren Entscheid.

Antwort:

-------------------------------------------------------------------------------------

-------------------------------------------------------------------------------------

-------------------------------------------------------------------------------------

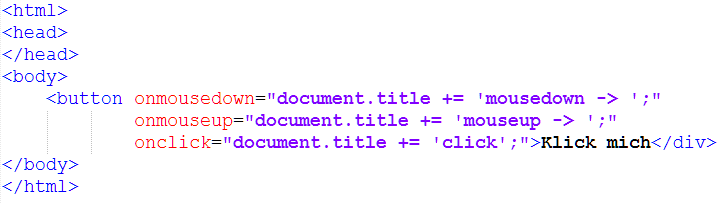
-------------------------------------------------------------------------------------

-------------------------------------------------------------------------------------

-------------------------------------------------------------------------------------

# Praxisaufgaben

1. JavaScript unterstützt eine ganze Reihe von Ereignissen und dazugehörigen Event-Handlern. Zu diesen Ereignissen zählen insbesondere Maus- und Tastatureingaben. Bei einem Mausklick zum Beispiel tritt das bestens bekannte click-Ereignis auf. Eine weitere Verfeinerung des Mausklicks bieten die zusätzlichen Ereignisse mousedown und mouseup, die eintreten, wenn die Maustaste gedrückt respektive wieder losgelassen wird. Die Frage ist nun, in welcher Reihenfolge treten diese drei Ereignisse bei einem Mausklick auf? Sie können die Reihenfolge am folgenden Beispiel ausprobieren. Klicken Sie auf die Schaltfläche und beobachten Sie, wie sich der Titel des Dokuments verändert. Wir setzen dazu die Eigenschaft document.title.



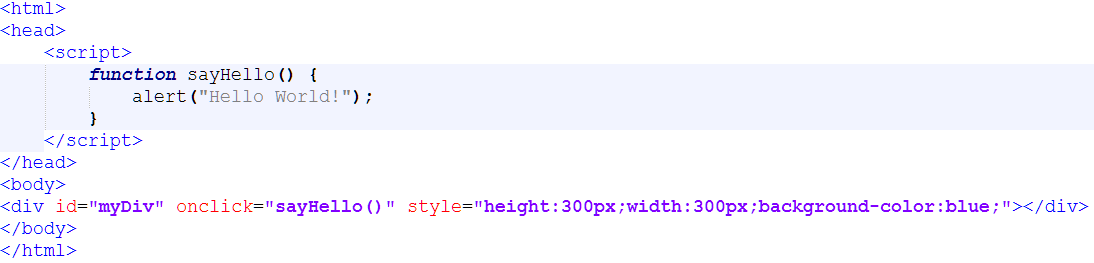
Notieren Sie hier nun die drei Events click, mousedown, mouseup in der richtigen Reihenfolge, wie sie bei jedem Mausklick gefeuert werden.

1: mousedown

2: click

3: mouseup

1. Ihr Kollege möchte den click-Event testen. Jedes Mal, wenn der Benutzer mit der Maus auf ein div-Element klickt, soll ein Meldungsfenster mit dem Text "Hello World!" angezeigt werden. Hier ist sein Code:



Mit dem Resultat ist ihr Kollege zufrieden; der Code funktioniert. Sie aber sind skeptisch, da Sie gehört haben, dass der Weg über das HTML-Attribut *onclick* nicht mehr zeitgemäss sei. Schlecht daran ist u.a., dass so HTML-Code und JavaScript-Code vermischt werden. Besser wäre es, alles JavaScript immer innerhalb eines <script>-Tags zu programmieren. Deshalb registrieren JavaScript-Profis heutzutage die Eventhandler auf eine andere Art.

Entfernen Sie das *onclick*-Attribut und verbessern Sie den Code Ihres Kollegen, indem Sie den Event mit der Methode addEventListener()registrieren. Sie finden seine HTML-Datei im Verzeichnis "Aufgabendateien": 3.1\_oncklick.html.

1. Erweitern Sie die Datei aus der vorherigen Aufgabe indem Sie die folgenden zusätzlichen Dinge implementieren:
2. Wenn der Benutzer am Mausrad dreht, egal in welcher Richtung, soll die Meldung "Mouse wheel has been rotated!" in einer Dialogbox ausgegeben werden.
3. Wenn mit der Maus über das div-Element gefahren wird, sollen das <div>-Element und das (HTML-)Dokument die Hintergrundfarbe tauschen: Das div erhält die Hintergrundfarbe des Dokumentes (weiss) und das Dokument erhält die Hintergrundfarbe des div (blau).   
   Verlässt die Maus das <div>-Element wieder, sollen die Hintergrundfarben erneut getauscht werden, sodass die Hintergrundfarben wieder so sind, wie sie waren bevor die Maus über das div-Element huschte.
4. <div>-Element ein- und ausblenden: Ein Doppelklick mit der Maus (irgendwo) auf das HTML-Dokument soll das <div>-Element verschwinden lassen und ein erneuter Doppelklick soll es wieder anzeigen.

# Erwartete Resultate

1. Alle Theoriefragen sind schriftlich (elektronisch) beantwortet und das Dokument mit den Antworten ist an dem dafür vorgesehenen Platz abgelegt/gespeichert.
2. Die Praxisaufgaben sind gelöst und die Lösung im dafür vorgesehenen Verzeichnis abgelegt.
3. Erledigte Arbeiten und gemachte Erfahrungen sind im Arbeitsjournal geloggt.