



Einführung in die Programmierung

- Google Apps Script & JavaScript -

Claudia Scheffler

17. September 2013



1 Document Object Model

2 Vordefinierte Objekte

3 Events



1 Document Object Model

2 Vordefinierte Objekte

3 Events



1 Document Object Model

2 Vordefinierte Objekte

3 Events



Fragen ?



Document Object Model

Was ist das DOM ?



Document Object Model

Unter dem Document Object Model (DOM) versteht man eine plattform- und programmiersprachenunabhängige Schnittstelle, die den Zugriff auf die Bestandteile eines Dokuments, d.h. das Hinzufügen, Löschen und Ändern des Inhaltes, der Struktur und der stilistischen Merkmale des Dokuments erlaubt.



Document Object Model

Was ist das DOM ?

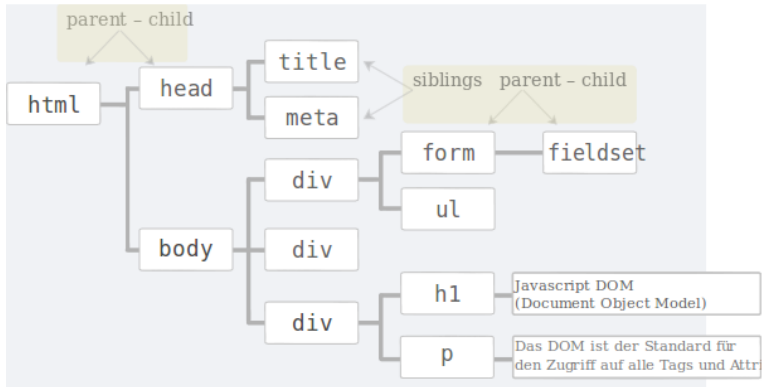


- Standard für den Zugriff auf alle Tags und Attribute der gesamten Webseite
- Baumstruktur setzt die Elemente eines HTML-Dokumentes in Beziehung zueinander
 - mit JavaScript kann so von einem Element aus durch die gesamte Seite navigiert werden
 - HTML-Tags eingefügt oder entfernt werden
- HTML-Element ist der Vaterknoten des Head- und des Body-Elements
 - Head-Element hat ebenfalls zwei Kinder:
 - Title
 - Meta
 - sind Nachfahren desselben Eltern-Elements



Document Object Model

Was ist das DOM ?





Document Object Model

Knotentypen



- Knotentypen:
 - Dokumentknoten stellt die gesamte Baumstruktur dar
 - Dokumentfragmentknoten stellt einen Teil der Baumstruktur dar
 - Elementknoten entspricht exakt einem Element in HTML
 - Attributknoten entspricht exakt einem Attribut in HTML
 - Textknoten stellt den textuellen Inhalt eines Elements dar
- Attributknoten stellen einen besonderen Knotentypen dar
 - sie kommen nicht als Knoten in der Baumstruktur vor
 - Elementknoten gebildet
 - keine Kinder von Elementknoten, sondern Eigenschaften



Document Object Model

Aufgaben des Javascript DOM für HTML-Seiten



- Methoden und Eigenschaften des DOM erzeugen:
 - erzeugen neue Tags
 - manipulieren Attribute
 - hängen Elemente mitsamt allen verschachtelten Elementen an eine andere Stelle des Dokuments
 - steuern den Fluss der Ereignisse, wenn das Dokument bereits im Fenster des Benutzers geladen ist



Document Object Model

Vorteile der Anwendung



- für ein angezeigtes Dokument im nach hinein die Darstellung zu verändern → Animation von Webseiten
- Zugriffe auf Formularinhalte
- Ein- und Ausblenden von spezifischen Seiteninhalte
- Unterscheidbarkeit der Elemente einer HTML-Seite
- ersparen das aufwändige Neuladen eines Dokuments



Document Object Model

Beispiel



Beispiel

```
<html>
<head>
<title>Beispiel 1</title>
</head>
<body>
<form name="mein_form" >
<input type="text" name="feld" value="Hallo Welt" >
<input type="button" name="aenderungsbutton" value="Text ändern"
onclick="document.mein_form.feld.value='Hallo JavaScript'" >
</form>
</body>
</html>
```



Document Object Model

Beispiel



Beispiel

```
<html>
<head>
<title>Beispiel 2</title>
</head>
<body>

<form name="mein_form" >
<iinput type="button" name="aenderungsbutton" value="Bild ändern"
onclick="document.images['meinLieblingsbild'].src='bild2.gif'" >
</form>
</body>
</html>
```



Fragen ?



Vordefinierte Objekt



- Array
- Boolean
- **Date**
- Function
- **Math**
- Number
- RegExp
- **String**



Vordefinierte Objekte

Zeichenketten und das Objekt String



- Zeichenketten existieren in zwei Arte:
 - elementare Werte des Typs String
 - String-Objekt (Instanzen des Objektes String)
- Unterschied ist selten von Bedeutung
 - JavaScript kann nach Bedarf zwischen den Formen wechseln
 - Methoden von String können ohne weiteres auf elementare String-Werte angewendet werden



Vordefinierte Objekte

Zeichenketten und das Objekt String



- Erzeugung eines String-Objektes:

Beispiel

- `text = new String("Ich möchte Javascript lernen!");`
- `text = "Ich möchte JavaScript lernen!";`

- Methoden der String-Objekte bieten Möglichkeiten:
 - Informationen zu ermitteln
 - ein Teil der Zeichenkette zu betrachten
 - Zeichenketten in Form und Inhalt zu verändern
- alle Methoden können hier nachgeschlagen werden:
SelfHTML



Vordefinierte Objekte

Das Objekt Date



- Möglichkeit mit Zeit und Datumsangaben zu arbeiten
- Beispiel:
 - Anzahl von Tagen zu berechnen, die noch bis zu einem bestimmten Tag verbleiben
- intern wird das Datum und die Zeit durch eine Zahl dargestellt, die die Anzahl der Millisekunden seit dem 1.1.1970 angibt
- bei der Umrechnung dieser Zeit kann die jeweilige Zeitzone angegeben werden
- Grundlage der Zeitberechnung:
 - Systemzeit des Rechners, auf dem das Skript ausgeführt wird
 - sollte ein Nutzer eine falsche Zeit in seinem Rechner eingestellt haben, dann wird auch die falsche Zeit im Script verwendet



Vordefinierte Objekte

Das Objekt Date



- Erzeugung eines Date-Objektes:

Beispiel

```
heute = new Date();
```

- wird keine Parameter übergeben, so wird das aktuelle Datum bzw. Uhrzeit verwendet
- ebenfalls besteht die Möglichkeit den Konstruktor mit Parametern zu verwenden:

Beispiel

- `anEinemTag = new Date(jahr, monat, tag, stunden, minuten, sekunden);`
- `anEinemTag = new Date("monat tag, jahr std:min:sek");`



Vordefinierte Objekte

Das Objekt Date



- Ist es notwendig beispielsweise das Datum "23.5.2001" in der Variablen anEinemTag zu speichern, kann dies mit beiden Möglichkeiten erreichen werden:

Beispiel

- `anEinemTag = new Date(2001, 4, 23);`
- `anEinemTags = new Date("May 23, 2001");`

Tipp!

Aufgrund der internen Zählweise von JavaScript wird für den Monat Januar der Wert 0 vergeben usw. Bei den Tagen wird allerdings normal durchnummeriert. Wird das Datum als String angegeben so muss die englischen Monatsname verwendet werden.



Vordefinierte Objekte

Das Objekt Math



- Möglichkeit mathematische Rechnungen durchzuführen
- Besonderheit unter den JavaScript-Objekten
 - Interpreter stellt automatisch ein einziges Objekt **Math** zur Verfügung
 - ein eigenes Math-Objekt kann nicht erzeugt werden
 - es müssen keine eigenen Variablen für das Objekt deklariert werden
 - zum Aufruf der Methode und Eigenschaften muss nur programmiert werden:

Syntax

`Math.Methodenname()` / `.Eigenschaft`



Fragen ?



Events

Was ist eine Event ?



Event (Ereignis)

Ein Event ist eine Veränderung eines Zustandes oder Parameters in einem System, auf die ein Programm oder System reagiert. Events können zufällig oder vorsätzlich ausgelöst werden.



Events

Was ist eine Event ?



- bei zufällig ausgelösten Ereignissen ist der Zeitpunkt des Eintretens nicht bekannt
- bei den ausgelösten Ereignissen ist der Zeitpunkt des Eintretens vorher klar bestimmbar
 - Betätigen eines Bedienelementes
 - Mausklick
 - Eintragen eines Textes in ein Formularelement
- ermöglichen dem Entwickler, sehr präzise die Aktionen des Nutzers im Browser zu beobachten, auszuwerten und darauf zu reagieren



Events

Event-Handler als HTML-Attribute



Event-Handler

Event-Handler können als Bindemitglied zwischen HTML und JavaScript angesehen werden.



Events

Event-Handler als HTML-Attribute



- Einbindung eines Event-Handler:
 - Handler wird als Attribut in einem HTML-Tag angegeben
- Erkennung der Event-Handler.
 - die Namen beginnen immer mit einem **on**
 - HTML-Attributnamen sind nicht "case-sensitiv"
- Beispiel:
 - `onClick()`
 - `onMouseover()`

Beispiel

```
<a href="#" onclick="alert('Hallo! Sie haben den Link  
angeklickt.')"> Bitte klicken </a>
```



Events

Event-Handler als Objekteigenschaften



- Handler können auch als Objekteigenschaften programmiert werden

Beispiel

```
<html>
<head>
<title>Event-Handler</title>
<script type="text/javascript">
function welcome() {
  alert("Willkommen");
}
window.onload = welcome;
</scrip> </head>
<body>
</body>
</html>
```



Events

Rückgabewerte von Event-Handle



- liefern Rückgabewerte an den Browser
- diese Werte steuern, wie der Browser diese Events weiter verarbeitet
- **Mit einfachen Worten:**
 - Browser ruft eine Funktion auf, die im Handler angegeben wurde, kann diese einen booleschen Wert zurück liefern
 - wird dieser mit **return** an den Browser weiter geben, so folgt eine Reaktion vom Browser
 - **true:** Ausführung der normalen Eventbehandlung des Browsers
 - **false:** Unterdrückung der Eventbehandlung



Fragen ?

Claudia Scheffler
siclsche@stud.hszg.de