

Web-Scripting

Javascript, jQuery, Dart

Überblick

- warum Javascript
- was ist Javascript
- Konzepte und Syntax
- OOP in Javascript
- Closures in Javascript

Überblick

- was ist jQuery
- Kernkonzept
- Animation mit jQuery
- Eventhandling mit jQuery

Überblick

- Was ist Dart
- Syntax und Konzepte

Warum Javascript

- Unterstützung in allen Browsern
 - Anwendungssoftware im Web
 - Spiele im Web
 - Große Anzahl Bibliotheken
-
- Zahl der Plattformen stark steigend
 - Fokus auf Mobile
 - Web zu Mobile Compiler
 - Unterstützt andere Konzepte

Was ist Javascript

- Clientbasierte Scriptsprache
- Kompiliert und ausgeführt im Browser
- 1995 von Netscape als LiveScript entwickelt
- Zur Ansteuerung von JavaApplet
- ECMAScript standardisierter Sprachkern (98)
- C ähnliche Syntax
- keine Verwandtschaft zu Java
- Prototypen basiert mit Closures

Sprachgrundlagen

- dynamisch typisiert
- C / Java ähnliche Syntax
 - Alle bekannten Strukturen (for, if, switch, while)
- Prototypen basierte Programmierung
- Funktion und Closures
- Unterstützt OOP

Objekte in Javascript - JSON

- Basis: window.Object
- Alles bis auf die Built-in typen ist ein Objekt
- Javascript Object Notation
- Definitions "Sprache" für Objekte
- einfacher als XML
- einfach zu serialisieren
- oft als Protokoll außerhalb von js eingesetzt
- gültiger Javascript Code

JSON

Code Beispiel

Funktionen

- Vollwertige Objekte
- Funktion als Argument
- Funktion als Rückgabetyp
- Konstruktoren Funktionen

Funktionen

Code Beispiel

OOP in Javascript

- Keine Klassen
- Funktionen sind vollwertige Objekte
- Prototypen basierte programmierung
- new Operator für Instanzen von Funktionen
- Konstruktor Funktion
- this operator
- Zugriff auf Member über Punkt Operator
- Zugriff auf Member über JSON notation

Prototypen basierte Programmierung

- Objekte entstehen durch Kopieren bestehender Objekte
- *prototype* Attribut: Zeiger auf Klonursprung
- Keine Trennung von Modell und Instanz
- Modell kann zur Laufzeit angepasst werden
- Schlechte Wartbarkeit
- Prototypen-Kette

Public, Protected und Private

- Zugriffsicherung ist nicht vorgesehen
- Zugriffssicherung über Variablenscope

Variablen Scope

- lokale : mit Schlüsselwort var deklariert
- global : alle anderen Variablen

Prototypen in Javascript

Code Beispiel

Closures

- erlauben Zugriff auf Variablen außerhalb des eigenen Scopes
- Auch dann wenn dieser Scope schon nicht mehr existiert
- Verknüpfung von Daten und Funktionen

Closures

Code Beispiel

Chaining

- Funktionen = Objekte
- Funktionen können Parameter sein
- Funktionen können Rückgabewert sein
- Closures erlauben impliziten Status

Chaining

Code Beispiel

Javascript Boolean

- Alles was nicht false ist, ist true
- False:
 - null, undefined
 - empty string
 - 0
- Empty Object ist true
- Lazy Evaluation
- Return Wert einer Booleschen Operation ist der Wert der die Operation gestoppt hat

Boolean

Code Beispiel

Javascript Module Pattern

- Erlaubt Vererbung
 - Erlaubt private Funktionen
 - Erlaubt private Member
 - Erlaubt explizite Globals
-
- Global Import
 - Module Export
 - Augmentation
 - Loose Augmentation
 - Tight Augmentation

Module Pattern

Code Beispiel

Closures und Performance

- Closures sind Performance intensiv
- Neuberechnung aller Zuweisung bei Instanziierung

This

- Wert von "this" bezieht sich immer auf das aufrufende Objekt
- Der Wert von "this" wird implizit übergeben
- Code Beispiel

Fallstricke bei Closures

- Closures in Schleifen

Javascript - wichtigste Quellen

Javascript Übersicht

<https://developer.mozilla.org/de/JavaScript>

Javascript Prototype Vererbung

https://developer.mozilla.org/en/Introduction_to_Object-Oriented_JavaScript

Javascript Module Pattern

<http://www.adequatelygood.com/2010/3/JavaScript-Module-Pattern-In-Depth>

Javascript und HTML5 : CreateJS

<http://createjs.com/>

jQUERY

- Javascript Bibliothek
- Stärken:
 - Browser Abstraktion
 - Selektion
 - Eventhandling
 - Animation von HTML elementen
 - CSS Modifikation
 - Ajax calls
- Kernkonzepte
 - CSS Selektion
 - Chaining

jQuery

Code Beispiel

Selektion

- \$ = jQuery = Konstruktor Funktion
- Elementsuche während der Initialisierung
- Suche mit CSS Selektoren
- Hilfsfunktionen für:
 - *getElementById / Name*
 - *querySelector*
 - *has (ab CSS4 im Browser unterstützt)*
- jQuery Code Beispiel

jQuery - wichtigste Quellen

jQuery - alle jQuery Packages im Überblick

<http://jquery.com/>

jQueryUI - Fertige GUI Widgets

<http://jqueryui.com/>

jQueryMobile

<http://jquerymobile.com/>

Dart

- Neue Programmiersprache fürs Web
- Einsatzgebiet: Server und Client
- Native Unterstützung in Chrome
- Compiliert zu Javascript
- Alpha-Phase

Dart - Sprachfeatures

- optional typisiert
- Klassenbasiert
 - Einfachvererbung
 - Interfaces
- Isolates
- Checked / Unchecked Mode
- Debugger + IDE
 - Autocompletion
 - Spellchecking

Dart - Einstieg

- Dart Editor im Web
- DartEditor - Eclipse fork
 - Compilieren von Server Applikationen
 - Compilieren zu Javascript
 - runtime debugging
- DartSDK
 - Dartium (Chrome + DartVM)
 - Samples

Dart

Code Beispiel

Dart - Besonderheiten

- Boolean Behandlung
 - Kurzschreibweise : `f() => x + 1`
 - Generics: `var i = List<String>()`
 - Operatorenüberladen
 - Kein Funktionsüberladen
 - Interface Default implementierung
 - Dokumentations Kommentare mit Markdown
- Isolates

Dart - wichtigste Quellen

<http://www.dartlang.org/>

Material

Folien und Beispiele befinden sich auf gitHub:

https://github.com/FriedrichWessel/Javascript_Samples

Quellen

<http://www.adequatelygood.com/2010/3/JavaScript-Module-Pattern-In-Depth>

<http://de.wikipedia.org/wiki/JavaScript>

<http://de.wikipedia.org/wiki/Closure>

<http://jibbering.com/faq/notes/closures/>

<https://developer.mozilla.org/de/JavaScript>

<http://stackoverflow.com/questions/111102/how-do-javascript-closures-work>

http://www.peterkropff.de/site/javascript/closures_feinheiten.htm

<https://developer.mozilla.org/en/JavaScript/Guide/Closures>

<http://mckoss.com/jscript/object.htm>

https://developer.mozilla.org/en/Introduction_to_Object-Oriented_JavaScript

<http://phrogz.net/js/classes/OOPinJS2.html>

<http://stackoverflow.com/questions/2470786/inheritance-of-closure-objects-and-overriding-of-methods>

<http://szeiger.de/blog/2007/08/17/super-constructor-calls-in-javascript/>

<http://javascript.crockford.com/private.html>

http://docs.jquery.com/How_jQuery_Works

http://www.authenticociety.com/blog/jQueryPluginTutorial_Beginner

<http://www.jsripters.com/jquery-disadvantages-and-advantages/>

<http://jquery.org/>

<https://developer.mozilla.org/en/DOM/Document.querySelector>

<http://www.dartlang.org>

Q&A

Danke!