

# Dynamische Mehrsprachigkeit und Text to Speech für Adobe Captivate



Für diese Aufgabe wird der Apache-Server benötigt!

# Überblick

- Didaktische Vorteile
- Technische und finanzielle Vorteile
- Texte dynamisch austauschen in Adobe Captivate
- Audiotexte dynamisch erzeugen in Adobe Captivate

# Texte in Captivate dynamisch austauschen mit JavaScript

## Verwendung von AJAX mit JavaScript in der Captivate-Autorenumgebung

- Seit der Version Adobe Captivate 8 gibt es eine JavaScript-Schnittstelle
- Ein einfacher Editor ist enthalten
- Überdies steht das JavaScript-Framework jQuery zur Verfügung

### Beispiel

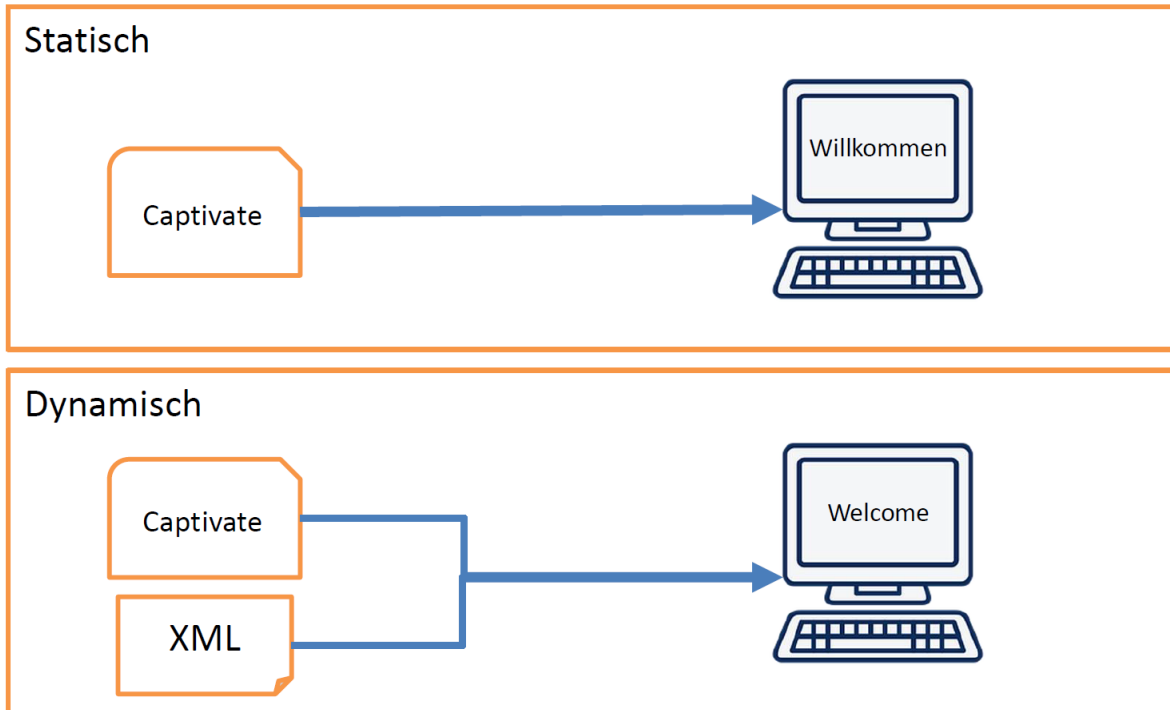
# Adobe Captivate koppeln mit Content Management Systemen

Adobe Captivate via XML, AJAX und JavaScript dynamisch mit Medien aus einem CMS befüllen.



# Captivate – JavaScript – jQuery - AJAX - XML

## Inhalte austauschen mit Javascript in Captivate



# Captivate – JavaScript – jQuery - AJAX - XML

Das Captivate-Verzeichnis nach der Erstellung der HTML5-Daten

Name	Type
 dr	File folder
 assets	File folder
 ar	File folder
 callees	File folder
 vr	File folder
 index.html	Chrome HTML Do...
 project.txt	Text Document
 goodbye.html	Chrome HTML Do...

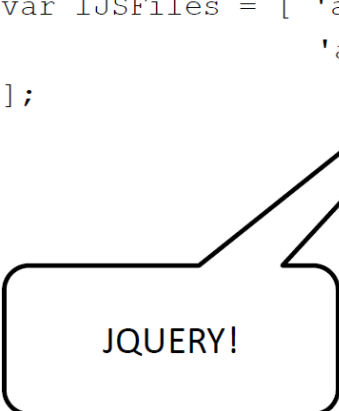
# Captivate – JavaScript – jQuery - AJAX - XML

Es besteht die Möglichkeit mit **JavaScript und/oder jQuery** zu arbeiten

HTML Datei

Zeile 98:

```
var lJSFiles = [ 'assets/js/jquery-1.11.3.min.js',  
                'assets/js/CPM.js', 'assets/playbar/playbarScript.js'  
];
```



JQUERY!

# Captivate – JavaScript – jQuery - AJAX - XML

Entwicklungsumgebung:

1. **Apache Server (z.B: XAMPP) muss für diese Aufgabe gestartet sein!**  
Z.B.: *c:/xampp/xampp-control.exe*
2. Web-Editor (z.B.: Dreamweaver)
3. Captivate (möglich mit Capt. 8, 9, 2017, 2018, 2019)



# Captivate – JavaScript – jQuery - AJAX - XML

Einrichtung der Folien in Captivate:

1. In eine Folie wird das AJAX-JavaScript über das JS-Editorfenster eingefügt
2. Für jedes Textfeld ist eine Variable in Captivate zu erstellen.  
(Menue → Projekt → Variablen)
3. Die Variablen werden mit Auszeichnung anstatt der Texte in die Textfelder geschrieben  
(\$\$VarName\$\$)
4. Die Variablen müssen erweitert werden, wenn mehr als 50 Zeichen Text aufgenommen werden sollen
5. Sprachbuttons z. B. in Form von Flaggen müssen die AJAX-Funktion mit der entsprechenden Sprachdatei starten

# Captivate – Öffnen des JS-Editors in einer Captivate-Folie

The image shows a screenshot of the Captivate software interface. On the left, a JavaScript editor window is open, displaying the following code:

```
JavaScript  
  
sprache = "../unity_vuforia_d.xml";  
lang_change();  
  
function lang_change() {  
$.ajax({  
  type: "GET",  
  url: sprache,  
  dataType: "xml",  
  success: function(xml) {  
    window.cpAPIInterface.setVariableValue("f1t1",  
      $(xml).find("entity")[0].firstChild.nodeValue);  
    $(xml).find("entity")[2].firstChild.nodeValue);  
    window.cpAPIInterface.setVariableValue("f4t1",  
      $(xml).find("entity")[3].firstChild.nodeValue);  
    window.cpAPIInterface.setVariableValue("f4t2",  
      $(xml).find("entity")[4].firstChild.nodeValue);  
    window.cpAPIInterface.setVariableValue("f5t1",  
      $(xml).find("entity")[5].firstChild.nodeValue);  
  }  
});  
}
```

The code is enclosed in a red box with the number '4' next to it. Below the code editor are buttons for 'Hilfe...', 'OK', and 'Abbrechen'. At the bottom of the slide, the text 'AR-App für iPhones' is visible.

On the right, the 'Aktionen' (Actions) panel is shown. The 'Aktionen' tab is selected and highlighted with a red box and the number '1'. Below it, the 'Beim Erreichen:' (When Reached) section has a dropdown menu set to 'JavaScript ausführen', which is also highlighted with a red box and the number '2'. Below that, the 'Skriptfenster' (Script Window) dropdown is highlighted with a red box and the number '3'. The 'Projekt weiter abspielen' (Continue Project) checkbox is checked. The 'Beim Verlassen:' (When Leaving) section has a dropdown menu set to 'Keine Aktion'.

# Captivate – JavaScript – **jQuery** - AJAX – XML

```
sprache = "../unity_vuforia_d.xml";    /* Tipp! XML-Daten nicht in den Projektordner legen! */

function lang_change() {
    $.ajax({                            /* jQuery-Variante von AJAX. jQuery ist in Captivate schon vorhanden! */
        type: "GET",
        url: sprache,
        /* Variable mit Inhalt einer URL oder reine URL angeben http://www.test.de */
        dataType: "xml",
        success: function(xml) {
            window.cpAPIInterface.setVariableValue("f1t1", $(xml).find('entity')[0].firstChild.nodeValue);
            window.cpAPIInterface.setVariableValue("f2t1", $(xml).find('entity')[1].firstChild.nodeValue);
            window.cpAPIInterface.setVariableValue("f3t1", $(xml).find('entity')[2].firstChild.nodeValue);
            window.cpAPIInterface.setVariableValue("f4t1", $(xml).find('entity')[3].firstChild.nodeValue)
        }
    });
}

lang_change();
```

# Captivate – Die Schnittstelle

## cpAPIInterface

<https://helpx.adobe.com/de/captivate/using/common-js-interface.html>

Dieses Objekt ist das Hauptobjekt mit der vollständigen JavaScript-Benutzeroberfläche. Es enthält viele kleine Hilfsprogrammfunktionen, die oft zum Ausführen von JavaScript-Aktionen erforderlich sind.

cpAPIInterface ist ein Objekt, das im Fensterbereich verfügbar ist. Um auf das Objekt der Benutzeroberfläche zuzugreifen, finden Sie im Folgenden die empfohlene Nutzung : *window.cpAPIInterface*

### Methoden:

Name	Beschreibung	Parameter	Verwendung
getVariableValue	Gibt den Wert des vorgegebenen Variablennamens zurück.	-	window.cpAPIInterface. getVariableValue ("cpQuizInfoStudentID");
setVariableValue	Legt den Wert des vorgegebenen Variablennamens mit dem vorgegebenen Wert fest.	<i>variableName</i> :String	window.cpAPIInterface. setVariableValue ("cpQuizInfoStudentID", "John");

# Captivate – Die Schnittstelle

```
window.cpAPIInterface.setVariableValue("f3t1", $(xml).find('entity')[2].firstChild.nodeValue);
```

cpAPIInterface ist ein Objekt, das im Fensterbereich verfügbar ist. Um auf das Objekt der Benutzeroberfläche zuzugreifen, finden Sie im Folgenden die empfohlene Nutzung : *window.cpAPIInterface*

Variablenname in der Folie

```
<entity>Zweite Folie</entity>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<deutsch>
  <data>
    <entity>E-Learning zur Erstellung einer
      Augmented-Reality-
      Anwendung.</entity>
    <entity>Erstellt von Jennifer Hoffstätter
      und Laura Weingartner
    </entity>
    <entity>Zweite Folie</entity>
  </data>
</deutsch>
```

# Captivate – JavaScript – jQuery - AJAX - XML

Variablen

Typ: Benutzer | v | Neu hinzufügen

Name: f3t1 | Aktualisieren

Wert: f3t1 | Entfernen

Verwendung

Beschreibung: Folie 3 Text1 | Nicht verwen... |  Geolocation

Sortieren: Alle | v

Text\_Entry\_Box\_1  
cpQuizInfoStudentID  
cpQuizInfoStudentName  
f1t1  
f1t2  
f2t1  
f2t2  
f3t1  
f4t1



Hilfe... | Schließen

Im Menü unter Projekt → Variablen das Variablenfenster öffnen

Variablenname eintragen. Wert und Beschreibung müssen nicht unbedingt eingetragen werden.




# Captivate – JavaScript – jQuery - AJAX - XML

`$$f1t1$$`



`$$f1t2$$`

`$$f1t3$$`



Hochschule Karlsruhe  
Technik und Wirtschaft  
UNIVERSITY OF APPLIED SCIENCES

Variablen mit doppelten Dollarzeichen in den Textfeldern eintragen.

# Captivate – JavaScript – jQuery - AJAX - XML

Variablen für große Texte (> 50 Zeichen ) erweitern.

The image shows a screenshot of the Captivate software interface. On the left, a slide thumbnail is visible with a red box around the text '\$\$f1t1\$\$' and a red '1' next to it. A dialog box titled 'Variable einfügen' is open, showing 'Variablentyp' set to 'Benutzer', 'Sortieren' set to 'Alle', and 'Maximale Länge' set to '500', with a red '3' next to the length field. On the right, the 'Stil' (Style) panel is visible, showing 'Stilname' as '+[Default Caption Style]', 'Beschriftungstyp' as 'transparent', and 'Zeichen' (Font) set to 'Trebuchet MS', 'Regular', size '36'. A red box highlights the font face dropdown, and a red '2' is next to it. The bottom of the slide shows the text 'AR-App für iPhones' and the Hochschule Karlsruhe logo.



# Captivate – JavaScript – jQuery - AJAX - XML

Unter Interaktionen → Schaltflächen zwei Schaltflächen erstellen. Die Flaggenbilder für die Sprachauswahl als Hintergrundbilder in diese Schaltflächen setzen. Die Funktionsaufrufe und die Sprachzuweisungen im JavaScript-Editor hinzufügen.

The screenshot displays the Captivate software interface. On the left, a hand holds a smartphone showing a Pokémon Go game. A red box labeled '1' highlights a flag icon in the bottom left corner. In the center, a JavaScript editor window is open, showing the code: `sprache = "../unity_vuforia_e.xml";` and `lang_change();`. A red box labeled '4' surrounds the code. On the right, a settings panel is visible. A red box labeled '2' highlights the 'JavaScript ausführen' (Execute JavaScript) option under 'Bei Erfolg:' (On success:). Another red box labeled '3' highlights the 'Skriptfenster' (Script window) dropdown menu. The 'Projekt weiter abspielen' (Continue project) and 'Unbegrenzte Versuche' (Unlimited attempts) options are checked. The 'Anzahl Versuche' (Number of attempts) is set to 1. The 'Mausklick zulassen' (Allow mouse click) option is checked. The 'Tastenkombination:' (Keyboard combination) section shows 'Keine' (None) selected. The bottom of the interface shows 'Hilfe...' (Help...), 'OK', and 'Abbrechen' (Cancel) buttons.

# Testen der Anwendung

1. Das Captivate-Projekt speichern
2. Das Captivate-Projekt veröffentlichen aber nicht ausführen!
3. In die Datei index.html den link zu responsivevoice.js eintragen  

```
<script src="http://code.responsivevoice.org/responsivevoice.js">  
</script>
```
4. Die Anwendung unter dem Apache-Server starten und testen  
z.B.: (<http://localhost/capt-projekt1/index.html>)
5. TIPP! Die XML-Daten sollten nicht im Captivate-Projektordner liegen, sonst werden sie bei jeder Veröffentlichung überschrieben

# Text To Speech-Software und Adobe Captivate

Auch in Captivate kann die Software ResponsiveVoice™ der Firma LearnBrite verwendet werden.

Die JavaScript-Bibliothek der responsivevoice.js wird per URL

**`http://code.responsivevoice.org/responsivevoice.js`**

in die von Captivate erzeugte index.html-Datei eingebunden.

Danach können über die gleichen Funktionsaufrufe wie im ersten Beispiel Wörter oder Sätze akustisch ausgegeben werden.

# Literatur

<https://helpx.adobe.com/de/captivate/using/common-js-interface.html>

<http://www.technischeredaktion.com/multimediamprojekte>

Mayer, R. E. (2009): Multimedia Learning Second Edition, Cambridge University Press

Siegel, K. (2017) Adobe Captivate 2017: Beyond The Essentials. IconLogic, Incorporated

Stiller, Klaus D. (2011) Das Modalitätsprinzip im multimedialen Instruktionsdesign: Empirische Basis, theoretische Erklärungen und vernachlässigte Faktoren. Universität Regensburg, <https://epub.uni-regensburg.de/22508/>

**Prof. Dipl.-Ing. Martin Schober** | Informations- und Medientechnik

Tagungsbeirat tekomp e.V.

Projektpartner EVEIL3D - Lernen in virtuellen Welten

**Hochschule Karlsruhe - Technik und Wirtschaft**

Fakultät für Informationsmanagement und Medien

Postanschrift: Postfach 24 40, 76012 Karlsruhe

Besucheranschrift: Amalienstr. 81-87 | 76133 Karlsruhe | Raum AM 113

fon +49 (0)721 925 - 2990 | fax +49 (0)721 925 - 1125

mobil +49 (0)173 945 82 18

[martin.schober@hs-karlsruhe.de](mailto:martin.schober@hs-karlsruhe.de)

[technischeredaktion.com/multimediaprojekte](http://technischeredaktion.com/multimediaprojekte)