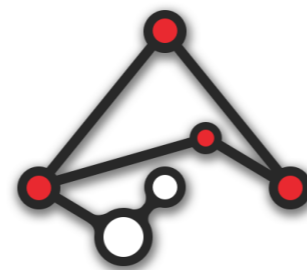


Anonyme JavaScript Funktionen in FileMaker

Einsetzen und Profitieren

Dr. Adam G. Augustin



www.agametis.de

Wer bin ich?

- Selbständiger FileMaker Entwickler im Raum München
- Beratung und Entwicklung seit über 10 Jahren
- Entwicklung von kundenspezifischen Datenbanken sowie Betreuung und Weiterentwicklung bestehender Lösungen
- FileMaker zertifiziert
- Zahlreiche Vorträge auf der FMK und dotfmp
- Web- und App-Entwicklung
- Mehr zu meinen Projekten mit Arbeitsbeispielen auf www.agametis.de



Inhalt

- Was sind anonyme Funktionen in JavaScript?
- Wie werden anonyme JS-Funktion in FileMaker definiert?
- Wie können anonyme JS-Funktionen in FileMaker eingesetzt werden?
- Ein paar Anwendungsbeispiele.
- FAQ

Anonyme Funktionen in JavaScript

Anonyme vs. nicht anonyme Funktionen in JavaScript

```
function zeigen() {  
  let x = 5;  
  console.log(x);  
}
```

Benannte (nicht anonyme) Funktion

```
function () {  
  let x = 5;  
  console.log(x);  
}
```

Anonyme Funktion

Selbstausführende anonyme Funktionen

```
function () {  
  let x = 5;  
  console.log(x);  
}
```

Anonyme Funktion

```
(  
  // hier kommt eine  
  // JavaScript Funktion rein  
)()
```

Zwei Klammern
"reichen schon"

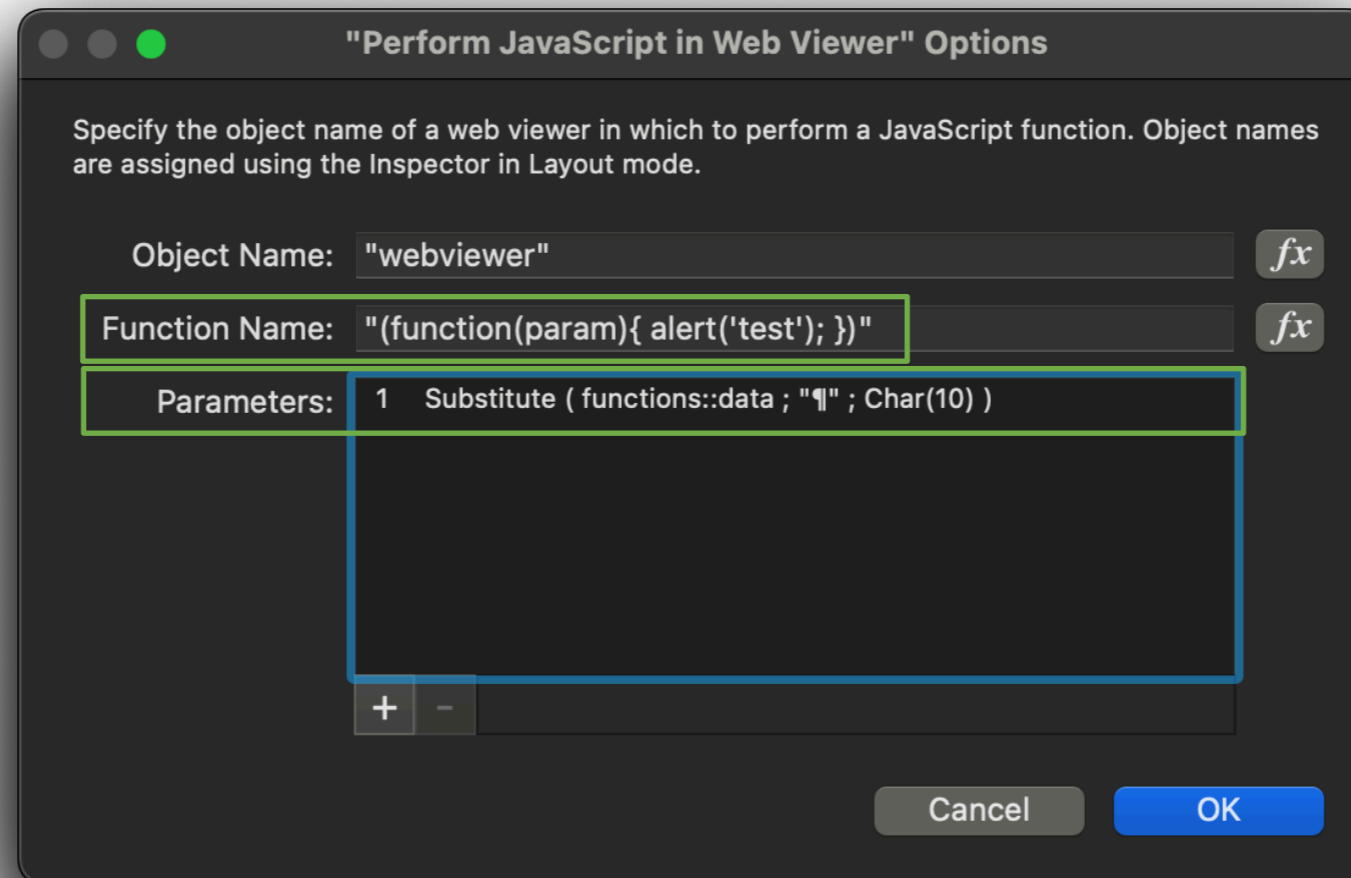
```
(function () {  
  let x = 5;  
  console.log(x);  
})();
```

Selbstausführende
anonyme Funktion

Eine Selbstausführende Funktion wird **sofort** nach der Deklaration ausgeführt.

Anonyme Funktionen in FileMaker

Anonyme Funktionen in FileMaker einsetzen



WICHTIG: im Feld "Function Name" wird die zweite Klammer **nicht** benötigt!
FileMaker fügt sie selbst hinzu.

HINWEIS: Falls bei den Parametern die Zeilenumschaltung Probleme bereitet, hilft das Ersetzen durch Char(10).

Umsetzung in FileMaker - Überblick

- Als Basis dient der WebViewer mit einer einfachen HTML-Struktur.
- In der anonymen Funktion können am Anfang bei Bedarf die benötigten Bibliotheken dynamisch mit Hilfe von JS (programmatisch) in die HTML Datei eingebunden werden.
- Danach folgt die eigentliche Aufgabe (weswegen man es eigentlich tut):
 - der eigentliche Code, der die gewünschte Aufgabe ausführt.
- Rückgabe der Ergebnisse aus dem WebViewer an FileMaker erfolgt am Ende des Codes mit Hilfe von `FileMaker.PerformScriptWithOptions()`

Umsetzung in FileMaker - HTML Struktur

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Dummy</title>
</head>
<body>
</body>
</html>

```

Basis HTML-Struktur

Umsetzung in FileMaker - HTML Struktur

A dark-themed code editor window with three colored window control buttons (red, yellow, green) at the top left. The code inside is:

```
<html>
  <head></head>
  <body></body>
</html>
```

Minimale HTML-Struktur reicht vollkommen

Umsetzung in FileMaker - JS-Bibliothek

```
let lib = document.createElement('script')
lib.src = '<CDN-URL>'
document.body.appendChild(lib)
```

JavaScript Bibliothek wird programmatisch in die HTML-Struktur eingefügt

```
<html>
  <head></head>
  <body></body>
</html>
```

Minimale HTML-Struktur

```
<html>
  <head></head>
  <body>
    <script src="<CDN-URL"></script>
  </body>
</html>
```

Minimale HTML-Struktur mit der eingefügten Bibliothek

Demo Beispiele

- Einfacher Dialog (keine zusätzliche JS-Bibliothek nötig)
- Daten an FM übergeben (keine zusätzliche JS-Bibliothek nötig)
- Summieren von Daten (keine zusätzliche JS-Bibliothek nötig)
- Datum - Maximum bestimmen (JS-Bibliothek momentjs)
- Datum - Formatieren (JS-Bibliothek momentjs)
- Zeichenketten kürzen (JS-Bibliothek lodash)
- Serialisierte PHP-Daten in JSON umwandeln (JS-Bibliothek phpUnserialize)

Demo



FAQ

Vielen Dank für euer Interesse!



ag.amet.is/fmk2024

Vielen Dank unseren Sponsoren

