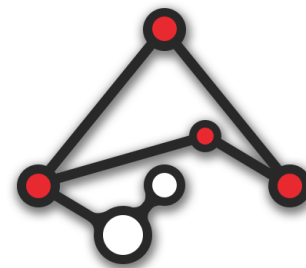


# JavaScript in FileMaker einsetzen

## JavaScript Track 4/4 (Kalender)

Dr. Adam G. Augustin



[www.agametis.de](http://www.agametis.de)

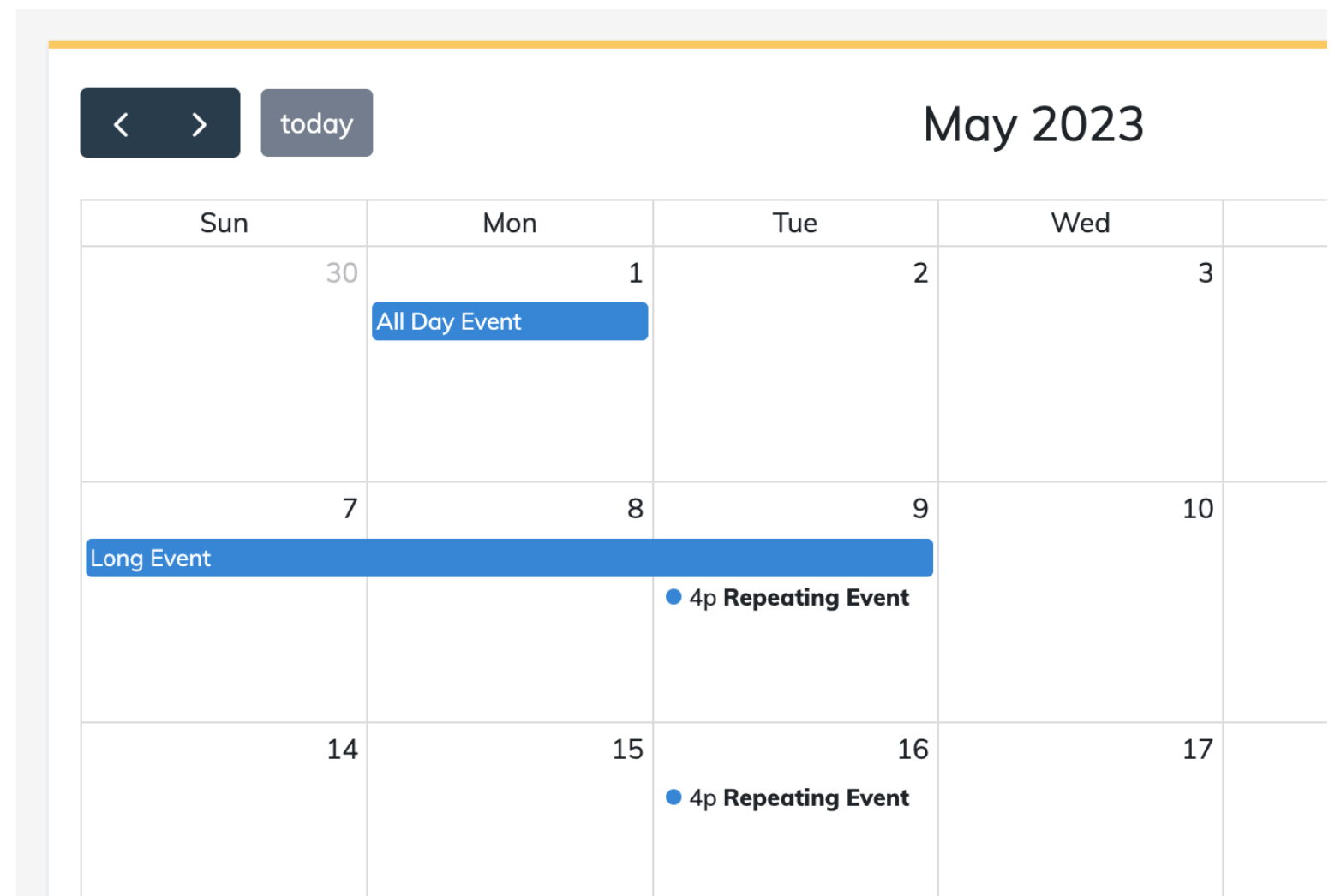
# Wer bin ich?

- 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](http://www.agametis.de)



# Inhalt

- Tooling und Debugging während der Entwicklung (damit es einfacher wird)
- “fullcalendar” als Kalender Bibliothek
- Grundfunktionen
- Externe Quellen
- Demos
- FAQ



# Wo wollen wir hin?

Records

Show All

New Record

Delete Record

Find

Sort

Share

Layout: event

View As:

Preview

|   | id  | id_source | allDay | start            | end              | title          | editable | tooltip               | + |
|---|-----|-----------|--------|------------------|------------------|----------------|----------|-----------------------|---|
|   | 156 | Feiertage | 1      | 05.06.2023 00:00 | 09.06.2023 00:00 | Monatsanfang   | 1        | Ja das ist der Anfang |   |
|   | 170 | Schulung  | 0      | 31.05.2023 06:00 | 31.05.2023 13:00 | Zahnarzttermin | 1        |                       |   |
| + |     |           |        |                  |                  |                |          |                       |   |
|   |     |           |        |                  |                  |                |          |                       |   |

- Visualisierung von FileMaker Daten in einem Kalender
- Dynamischer/Interaktiver Datenaustausch zwischen dem Web Viewer/JS und FileMaker

<

Heute

>

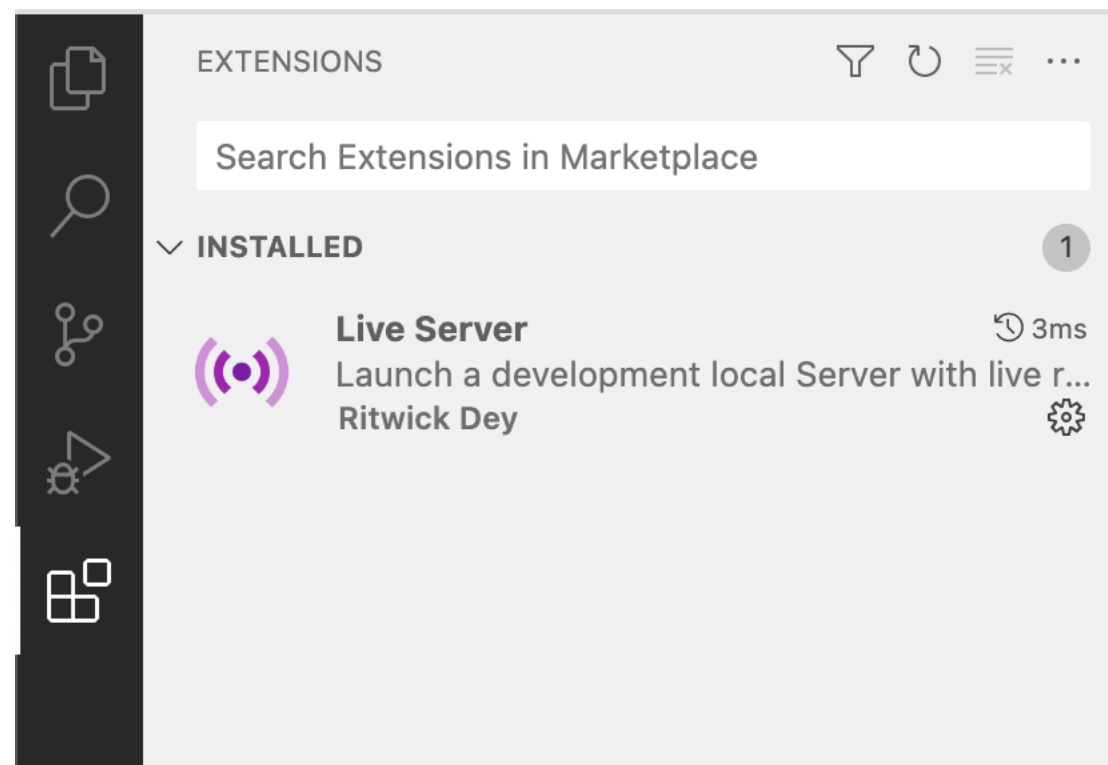
Juni

| Mo                            | Di            | Mi   |  |
|-------------------------------|---------------|--|--|
| <div>KW22</div> <div>29</div> | <div>30</div> | <div>31</div> <div>● 06 Uhr Zahnarzttermin</div> |  |
| <div>KW23</div> <div>5</div>  | <div>6</div>  | <div>7</div>                                     |  |
| <div>Monatsanfang</div>       |               |  |  |
|                               |               |  |  |
| <div>KW24</div> <div>12</div> | <div>13</div> | <div>14</div>                                    |  |

# Tooling und Debugging (während der Entwicklung)

# Tooling (der Werkzeugkasten)

- Editor: z.B. Microsoft Visual Studio Code (VSCode) <https://code.visualstudio.com/>
- Mit Erweiterungen können Funktionen nachinstalliert werden.
  - Webserver in Form der VSCode-Erweiterung “Live Server”



Open Source und kostenfrei

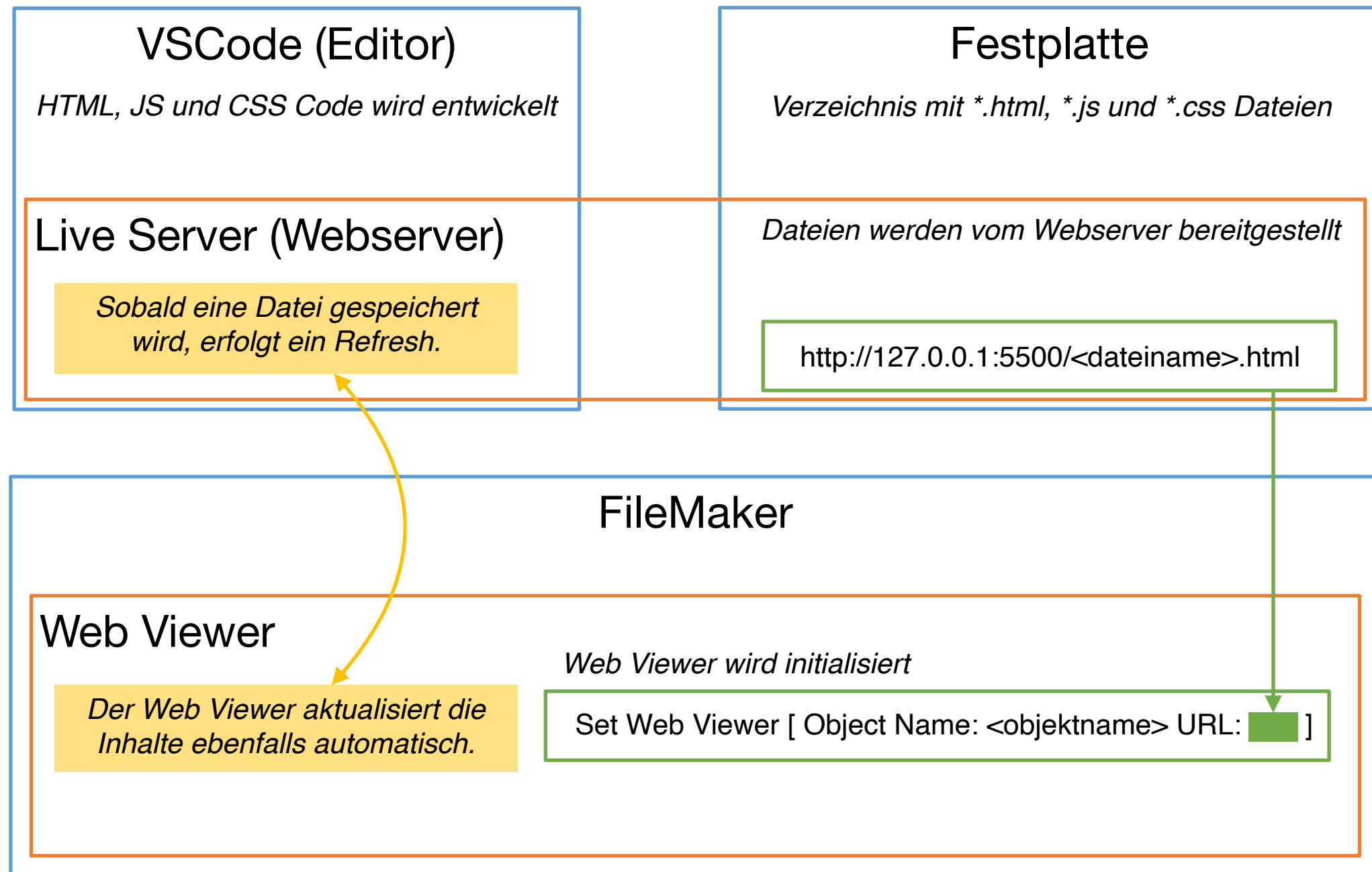
Webserver direkt in der  
Statusbar starten und stoppen

# Tooling (auf der FileMaker Seite)

- FileMaker ab 19.3 (weil, ab dieser Version quasi kein Unterschied in JavaScript zwischen Windows und Mac/iOS vorhanden)
- Der Web Viewer muss mit einem Objektnamen versehen werden, damit man ihn mit dem Skriptschritt “Set Web Viewer/Web Viewer festlegen” ansprechen kann.
  - Im Web Viewer muss in der Regel keine URL oder ähnliches angegeben werden.
  - Der Web Viewer ist quasi leer und wird zur Laufzeit initialisiert (“gefüllt”).
- Daten für den Web Viewer werden nach Möglichkeit immer mit dem Skriptschritt “Execute FileMaker Data API” geholt:
  - Grund: Daten sind sofort in einer JSON-Struktur verfügbar.
- Daten werden nach Möglichkeit immer mit dem Skriptschritt “Perform JavaScript in Web Viewer” an den Web Viewer übergeben:
  - Grund: Inhalte des Web Viewers müssen nicht vollständig geladen werden, weil nur die Daten aktualisiert werden.

*Gilt grundsätzlich und nicht nur während der Entwicklung*

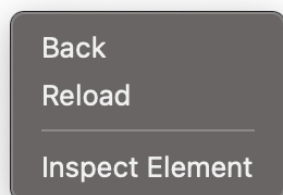
# Tooling (während der Entwicklung)



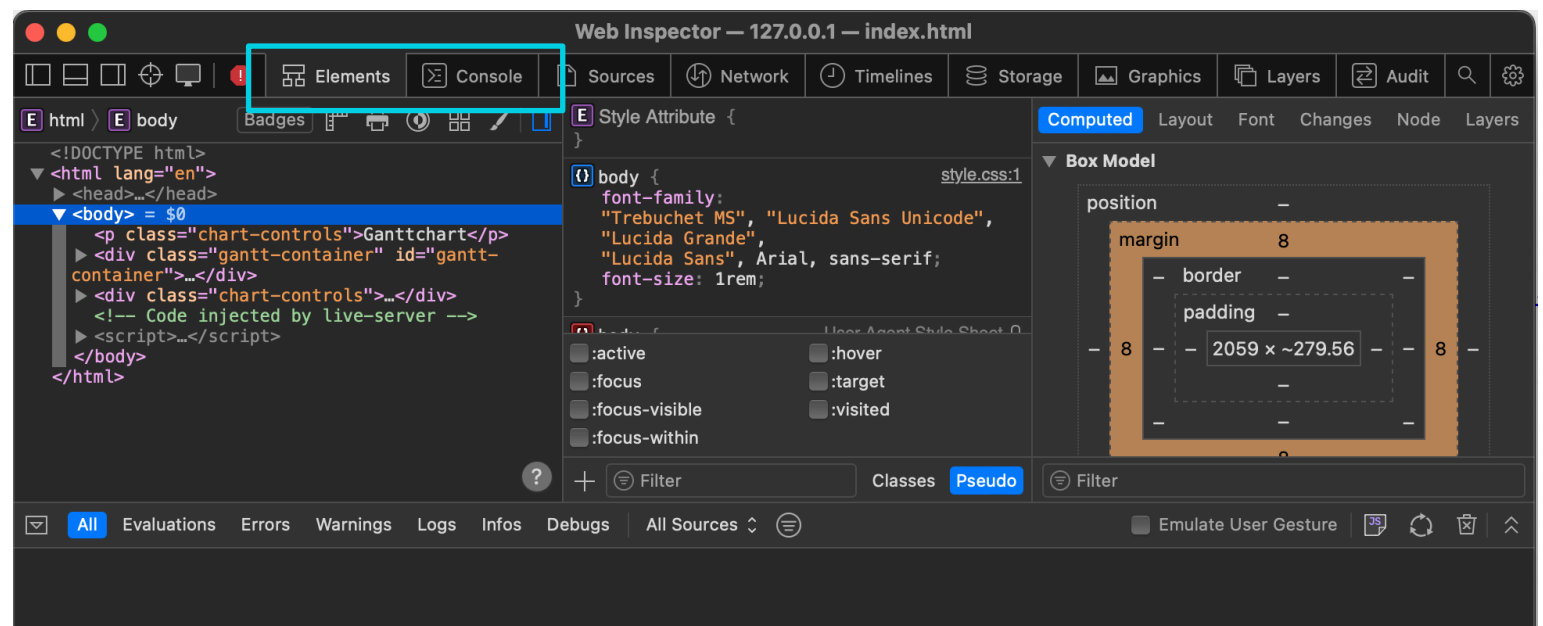


# Debugging - 1

- Debugging des JavaScript Codes **direkt im FileMaker Web Viewer** möglich:
  - Mit der rechten Maustaste kann der Web Inspektor (Developer Tools) aufgerufen werden (Inspect Element/Untersuchen in FM).
  - Auf dem Mac muss der Inspektor einmalig aktiviert werden. Im Terminal mit folgendem Befehl:
    - “defaults write com.filemaker.client.pro12 WebKitDebugDeveloperExtrasEnabled -bool YES”
  - Unter Windows ist der Web Inspektor immer verfügbar.



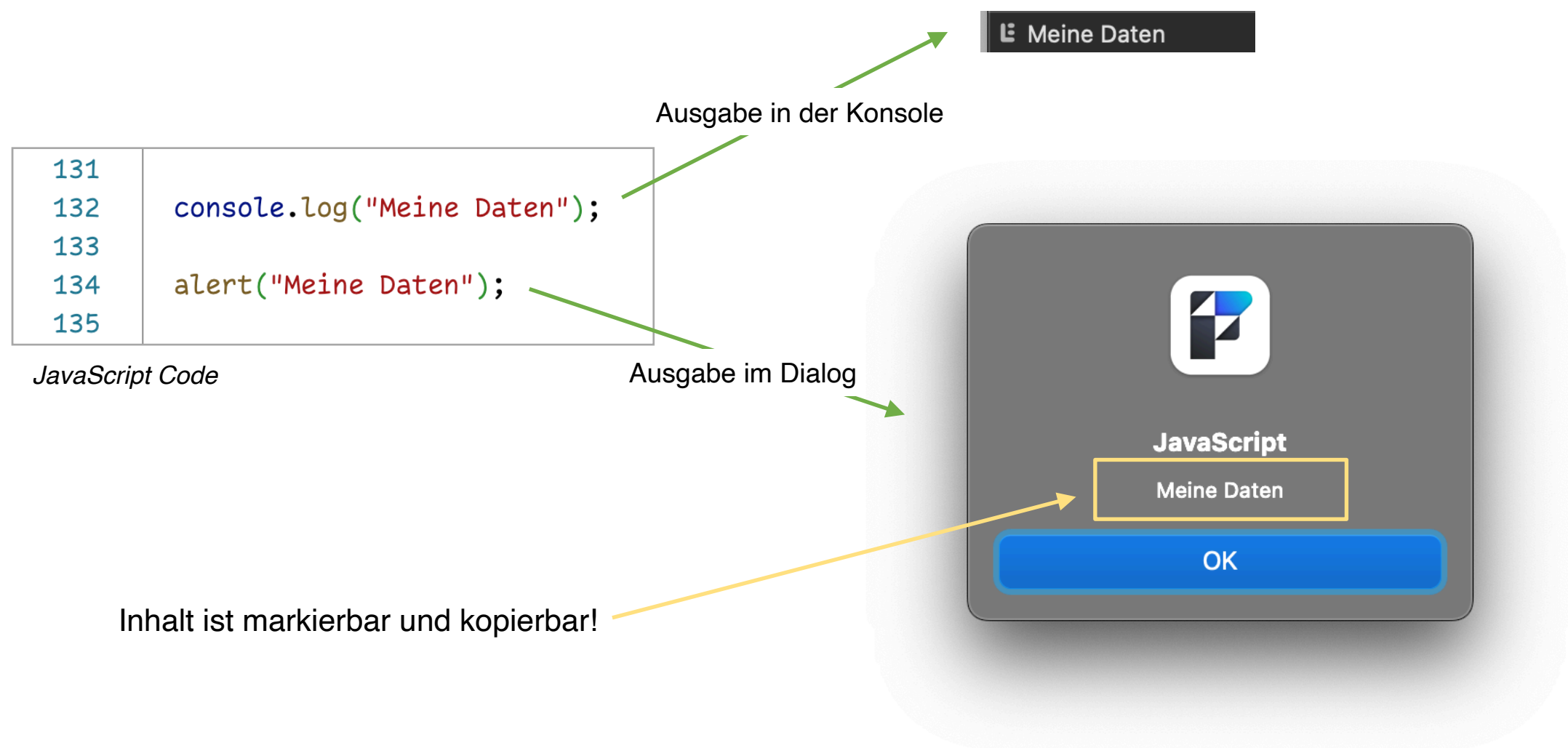
Rechte Maustaste



Konsole unter MacOS

# Debugging - 2

Mit “`console.log(daten)`” (in der Konsole) oder “`alert(daten)`” (als Dialog) können Informationen (z.B. Zwischenstände) im Web Viewer mit JavaScript ausgegeben werden.



# Kalender Add-On vs. JS-Bibliothek

# Wieso selbst einen JS-Kalender einbinden?

## Kalender als Add-ons

- **Pro:**
  - Einfach zu implementieren
  - Schnell erlernbarer Funktionsumfang
  - Überhaupt keine JavaScript-Kenntnisse notwendig
- **Kontra:**
  - Nicht updatefähig
  - Funktionsumfang sehr eingeschränkt

## Kalender selbst implementiert

- **Pro:**
  - Volle Flexibilität der verwendeten Bibliothek
  - updatefähig
  - Weitere Vorteile folgen auf den nächsten Seiten.
- **Kontra:**
  - Aufwendige Implementierung
  - Zusätzliche Lizenzkosten möglich
  - Basiskenntnisse in JavaScript sehr von Vorteil

# Kalender - JS-Bibliothek

- FullCalendar Bibliothek von <https://fullcalendar.io/>
- Ausgewachsene Bibliothek, die kaum Wünsche offen lässt:
  - “Powerful: With over 300 settings, and more being added every release, FullCalendar can do just about anything.”
  - “Lightweight: Reduce your project's bundle size by using FullCalendar's modular plugins.”
- CDN Links für den schnellen Einstieg verfügbar unter <https://www.jsdelivr.com/>
  - Gesamtübersicht aller Plugins: <https://www.jsdelivr.com/?query=author%3A+fullcalendar>
- Auch mittels lokaler Dateien nutzbar.

# Kalender Features

# Kalender - Grundfunktionen

- Kostenlose Ansichten: Monat, Woche, Tag, Liste, Jahr (ab Version 6.1)
- Weitere Ansichten (kostenpflichtig für kommerzielle Zwecke): Resource/Schedule horizontal und vertikal mit unterschiedlichen Zeiträumen
- Mehrere Quellen/Kalender definierbar: interne Datenquellen (FileMaker) und externe Datenquellen (z.B. Google-Kalender).
- Dateneingabe(-definition) und Datenänderung direkt im Web Viewer möglich.
- iOS/Touch-Device kompatibel

# Kalender - Interessante Eventeigenschaften

Viele Eigenschaften sind auf unterschiedlichen “Ebenen” beeinflussbar und somit sehr flexibel einsetzbar:

- “**editable**”: auf drei Ebenen verfügbar: System, eventSource, Event
  - “**startEditable**”, “**durationEditable**”, “**resourceEditable**”
- “**display**”: z.B. mit dem Wert “background” kann das Event nur im Hintergrund als Info angezeigt werden (Feiertage, Geburtstage, spezielle Firmentermine)
- “**backgroundColor**”, “**borderColor**”, “**textColor**”: genauso wie “**editable**” sehr fein kontrollierbar
- “**extendedProps**”: damit sind Events um beliebige Eigenschaften erweiterbar: (z.B. in der Demo mit Tooltips)

Vollständige Liste der Eigenschaften:

event: <https://fullcalendar.io/docs/event-object>

eventSource (Kalender): <https://fullcalendar.io/docs/event-source-object>



# Externe Kalender einbinden

- Wenn die Daten nicht nur aus FileMaker kommen sollen. (z.B. weil der Kunden bereits den Google-Kalender benutzt und möchte die Daten in seiner FM-Lösung ebenfalls sehen)
- Am Beispiel des Google-Kalenders
  - Anleitung verfügbar unter: <https://fullcalendar.io/docs/google-calendar>
- Voraussetzungen: <https://console.cloud.google.com/>
  - API “Google Calendar API” aktivieren
  - API-Key unter Credentials einrichten (kann beliebig eingeschränkt werden)
  - Kalender für den öffentlichen Zugriff freigeben (Zugriff wird über den API-Key kontrolliert)

Access permissions for events

☒ Make available to public

See all event details ▾

Integrate calendar

Calendar ID

ff83e42705a3a84b2733e51c6b36e874a40[REDACTED]@group.calendar.google.com

# Umgang mit Versionen

- Vorteile einer neuen Version liegen auf der Hand:
  - Bugfixes
  - Sicherheitsupdates
  - Funktionsupdates
- Aber Vorsicht: teilweise **breaking changes** zwischen den Hauptversionen möglich.
  - Vor dem Update immer die Dokumentation beachten und eigene Programmierung daraufhin überprüfen.
  - Die sehr gute Dokumentation des Projektes beschreibt ausführlich alle Änderungen.

# Herausforderungen beim Datumsformat

- Beim Datenaustausch zwischen unterschiedlichen Systemen muss man in der Regel auf die Formatierung des Datums/Zeitstempels achten.
- FileMaker arbeitet für gewöhnlich mit dem US Format (MM/TT/JJJJ).
- Seit FileMaker 2023 (FileMaker Pro 20.x) haben wir nun beim Skriptschritt “Execute FileMaker Data API” die Möglichkeit die Formatierung zu beeinflussen.
  - Neuer Parameter “**dateformats**” mit den drei Optionen
  - “0” - US Format (also wie bisher => default Ausgabe wenn nicht gesetzt)
  - “1” - Datumsformat der Datei/Feldes (beim Initialisieren der Datei automatisch festgelegt)
  - “2” - ISO8601 (JJJJ-MM-TT => ein Segen)
  - Vorsicht: der Parameter **dateformats** erwartet die übergebene Zahl als String (mit “JSONString” als Formatangabe ist es immer gegeben)

# Demo

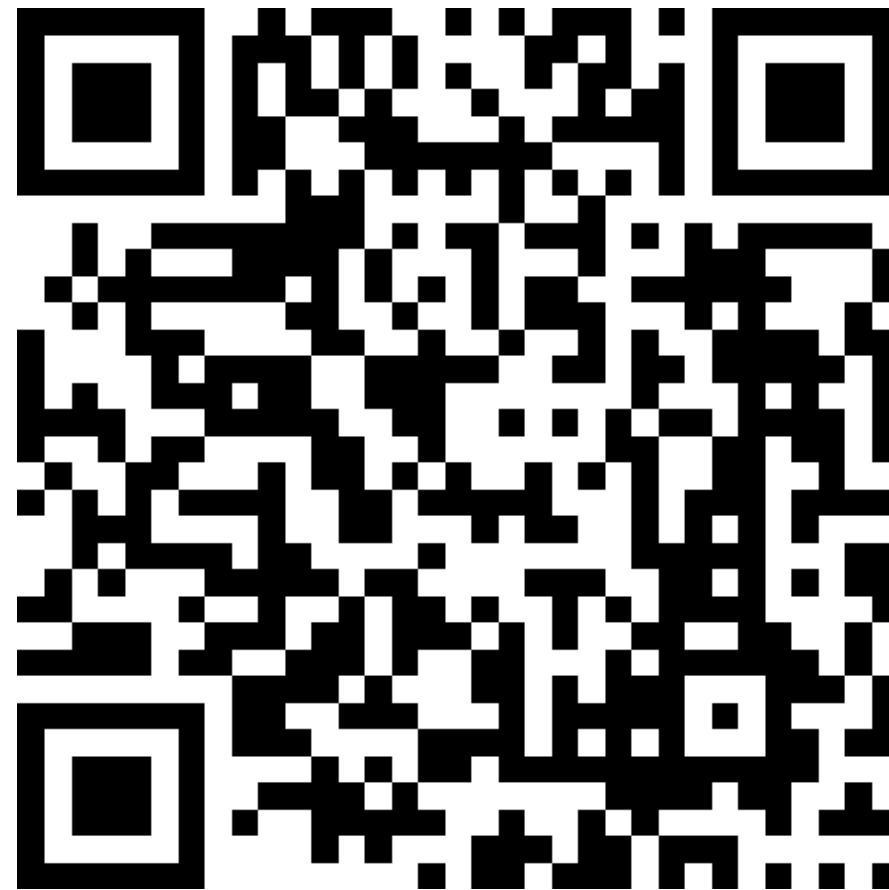


# Die Demos

- Demo1: Grundsätzliches Einbinden der Bibliothek mit Hilfe eines CDN.
- Demo2: Neue Jahresansicht in Version 6.1.
- Demo3: Google Kalender als externe Quelle.
- Demo4: Vollständige Integration in FileMaker:
  - Lokale JS-Bibliotheken werden bei Bedarf aus Container-Feldern geladen.
  - In FileMaker gespeicherte Daten werden im Kalender dargestellt.
  - Interaktive Erstellung und Modifikation von Daten.
  - Grundfunktionalität erweitert um eigene Logik zum Nachladen von Daten.
  - Grundfunktionalität erweitert um Tooltips.

# Downloads

<https://ag.amet.is/fmk2023>



# FAQ

**Vielen Dank für euer Interesse!**

# Vielen Dank unseren Sponsoren

