



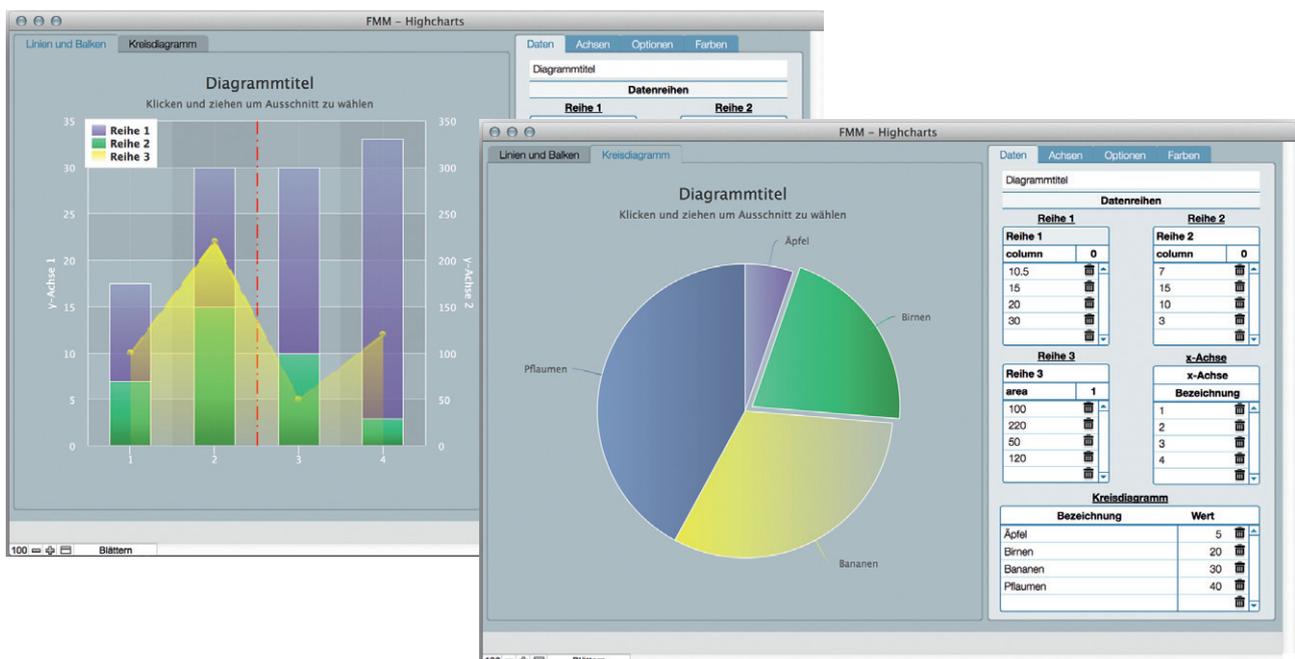
Nachdruck aus dem FileMaker Magazin
siehe auch: www.filemaker-magazin.de

Diagramme mit „Highcharts“

Erstellen Sie Diagramme, die auch in Runtimes, WebDirect und FileMaker Go funktionieren

Das Diagramm-Modul in FileMaker ist eine tolle Sache, hat allerdings einen entscheidenden Nachteil: Es funktioniert nicht in Runtime-Lösungen. Zwar gibt es auf dem Markt einige Plugins aber die meisten scheiden für mich aus, weil sie nicht mit FileMaker Go kompatibel sind oder über das Internet angesteuert werden. Eine Ausnahme bildet „Highcharts“, eine Offline-Lösung, die meine Prämisse nach universeller Einsetzbarkeit erfüllt, alle erdenklichen Arten von Diagrammen darstellen kann und vielseitig konfigurierbar ist. Mit diesem Beitrag möchte ich denjenigen, die dieses Modul ausprobieren möchten, eine kleine Starthilfe geben. „Highcharts“ ist für die nicht kommerzielle Nutzung kostenlos und kann über den angegebenen Link heruntergeladen werden.

Da der gesamte Code zu lang für diesen Artikel ist, habe ich Ihnen eine kommentierte Beispieldatei zur Verfügung gestellt, die Sie mit wenigen Modifikationen an Ihre Bedürfnisse anpassen und so alle grundlegenden Diagramme erstellen können. Im Folgenden werde ich auf einige ausgewählte Bestandteile eingehen und diese erläutern, die anderen funktionieren nach dem gleichen Prinzip. Lassen Sie sich von den vielen Möglichkeiten nicht abschrecken: Das System ist auf den zweiten Blick einfach, logisch und schnell zu erlernen. Zudem eröffnet das codebasierte Arbeiten die Möglichkeit, ganze Blöcke zu kopieren und an anderer Stelle wieder einzufügen. Nach einer kurzen Eingewöhnungsphase werden Sie überrascht sein, wie schnell Sie vorankommen und wie vielfältig die Möglichkeiten sind.



Funktionsweise

Highcharts basiert auf Javascript, wird im Web Viewer ausgeführt und benötigt das Modul *Highcharts.js* (aktuelle Version 4.0.3) sowie die **jQuery**-Bibliothek (aktuelle Version 2.1.1). Die Quellcodes sind sehr lang und müssen in einem globalen Feld gespeichert werden, da der Text im Web-Viewer-Modul maximal 3000 Zeichen umfassen kann.

Erstellen Sie zunächst die globalen Felder **Charts::jquery_js** und **Charts::highcharts_js** und kopieren Sie den Quelltext der Dateien *jquery.js* und *highcharts.js* hinein.

Diese Felder werden dann im Web Viewer eingebunden:

```
<script>" & Charts::jquery_js & "</script>
<script>" & Charts::highcharts_js & "</script>
```

Syntax

Auf der **Highcharts**-Website¹ finden Sie eine Auflistung und Beschreibung aller Parameter. Außerdem können Sie dort viele Optionen im Live-Modus testen. Anhand der Konfiguration der Datenreihen möchte ich beispielhaft die Syntax von **Highcharts** erläutern:

```

//=====
//===== Datenreihen =====
//=====
series:
[
  // Datenreihe 1
  {
    name: "" & Charts::Bezeichnung_Feld_1 & "",
    //color: '#D71216',
    type: "" & Charts::Art_Feld_1 & "",
    yAxis: "" & Charts::y_Achse_Reihe_1 & "",
    data: ["& Austauschen ( Liste ( Tabelle::Wert_Feld_1 ) ; "¶" ; ",")&"]
  },
  // Datenreihe 2
  {
    name: "" & Charts::Bezeichnung_Feld_2 & "",
    //color: '#D71216',
    type: "" & Charts::Art_Feld_2 & "",
    yAxis: "" & Charts::y_Achse_Reihe_2 & "",
    data: ["& Austauschen ( Liste ( Tabelle::Wert_Feld_2 ) ; "¶" ; ",")&"]
  }
]

```

Die Zeilenumbrüche (¶) am Anfang der Zeilen sind notwendig, damit **Highcharts** einwandfrei funktioniert. Betrachten wir die Zeile, in der der Name der Datenreihe festgelegt wird:

```
¶ name: "" & Charts::Bezeichnung_Feld_1 & "",
```

Textwerte werden zwischen einfache Anführungszeichen (') gesetzt. Um der Datenreihe den festen Namen „Datenreihe_1“ zu geben, muss die Zeile so lauten:

```
¶ name: 'Datenreihe_1'
```

Sollen Felder aus einer Tabelle eingebunden werden, benötigt man zusätzlich doppelte Anführungszeichen und &-Zeichen. Um beispielsweise statt des expliziten Wertes 'Datenreihe_1' das Feld **Charts::Bezeichnung_Feld_1** zu verwenden, ersetzt man

```
'Datenreihe_1'
```

durch

```
" & Charts::Bezeichnung_Feld_1 & ""
```

Achten Sie darauf, dass zwischen den einfachen und den normalen Anführungszeichen kein Leerzeichen steht. Außerdem müssen Sie – anders als bei FileMaker – die Groß- und Kleinschreibung beachten. So macht es einen Unterschied, ob Sie „yAxis“ oder „yaxis“ schreiben – Ersteres funktioniert, Letzteres nicht. Am Ende der Zeile steht ein Komma, weil weitere Parameter auf dieser Ebene folgen. Nach dem letzten Parameter entfällt das Komma.

Wenn Sie Felder einbinden, dürfen diese nicht leer sein. Ihr Diagramm wird erst angezeigt, wenn alle Felder gültige Werte enthalten.

Mit dem Parameter „yAxis“ legen Sie an dieser Stelle übrigens fest, ob die Werte auf der primären oder auf der sekundären y-Achse abgebildet werden. Da es sich hierbei um Zahlenwerte handelt (0 steht für die primäre y-Achse, 1 für die sekundäre y-Achse), entfallen die einfachen Anführungszeichen.

Die Zeile

```
¶ //color: '#D71216',
```

wurde durch den doppelten Schrägstrich deaktiviert. Dieser Parameter ermöglicht es, jeder Datenreihe eine dezidierte Farbe zuzuordnen. Da ich an anderer Stelle im Code eine Farbabfolge für Datenreihen festgelegt habe (siehe Beispieldatei), ist die Farbzuzuordnung an dieser Stelle nicht notwendig.

Die grafisch darzustellenden Werte werden in folgender Zeile festgelegt:

```
¶ data: ["& Austauschen ( Liste ( Tabelle::Wert_Feld_2 ) ; "¶" ; ",")&"]
```

Mithilfe der Funktion „Liste“ wird eine durch Zeilenumbruch getrennte Werteliste erzeugt. Da die **Highcharts**-Syntax jedoch durch Komma getrennte Werte erwartet, werden die Zeilenumbrüche (¶) mittels Austauschen durch ein Komma ersetzt. So weit, so gut, aber was ist, wenn Zahlen mit Nachkommastellen behandelt werden? In diesem Fall würde **Highcharts** den Wert „3,7“ als zwei Werte interpretieren: „3“ und „7“. Hierfür gibt es mehrere Workarounds. Gegebenenfalls können Sie ein Formelfeld erzeugen, mit dem Sie den eigentlichen Wert auslesen und Kommas durch Punkte ersetzen:

```
Austauschen (Tabelle::Wert_Feld_2; ","; ".")
```

Die Datenreihen 1 und 2 stehen jeweils in geschweiften Klammern, getrennt durch ein Komma (siehe Abb. vorige Seite).

Mit „type“ legen Sie den Diagrammtyp fest. Ich habe an dieser Stelle ein Feld eingefügt, sodass man im Blätternmodus per Mausklick beispielsweise von einem Liniendiagramm zu einem Balkendiagramm wechseln kann. Mögliche Werte im Standard-Modul sind: 'line', 'spline', 'area', 'areaspline', 'column', 'bar', 'pie' und 'scatter'. Generell können Sie alle Attribute durch Felder ansteuern und so das Diagramm im Blätternmodus modifizieren.

Farben

Highcharts akzeptiert mehrere Formate zur Eingabe von Farbwerten:

Name der Farbe

Beispiel: white, black, green, blue ...

Hexcode

Beispiel: #FFFFFF für weiß, #000000 für schwarz, ...

RGB-Werte ohne Transparenz

(rgb(Rotwert, Grünwert, Blauwert))

Beispiel: rgb(100, 200, 100)

RGB-Wert mit Transparenz

(rgba(Rotwert, Grünwert, Blauwert, Alphawert))

Beispiel: rgba(100, 100, 100, 0.7)

Der Alphawert liegt zwischen 0 und 1, wobei 0 für völlige Transparenz und 1 für vollständige Sichtbarkeit steht. Achten Sie darauf, dass die Klammer ohne Leerzeichen an „rgb“ bzw. „rgba“ angeschlossen ist.

Farbverläufe

```
!colors:
! [
!   {
!     linearGradient:
!       {
!         x1: 0,
!         y1: 0,
!         x2: 1,
!         y2: 0
!       },
!     stops:
!     [
!       [0, 'rgba( 154, 120, 200, 0.7)'],
!       [1, 'rgba( 144, 70, 150, 0.7)']
!     ]
!   },
!   {
!     linearGradient: linGradient,
!     stops:
!     [
!       [0, "" & Charts::Farbe_1_Reihe_2 & ""],
!       [1, "" & Charts::Farbe_2_Reihe_2 & "" ]
!     ]
!   }
! ],
```

Um Farbverläufe zu definieren, sind zwei Bestandteile notwendig (achten Sie auch hier auf die Groß- und Kleinschreibung): der „linearGradient“, der die Richtung des Farbverlaufes festlegt (horizontal, vertikal, diagonal), und die „stops“, die die beiden Endfarben beschreiben.

Wie Sie im obigen Beispiel sehen, sind die Farben des ersten Verlaufes explizit festgelegt, während ich bei der zweiten Farbe die Felder **Charts::Farbe_1_Reihe_2** und **Charts::Farbe_2_Reihe_2** integriert habe. So kann man diesen Farbverlauf im Blätternmodus steuern.

Empfehlungen

- Achten Sie auf Groß- und Kleinschreibung.
- Wie schon im Mathematikunterricht der fünften Klasse ist auch hier die Klammersetzung eine Plage. Wenn Sie auch nur eine Klammer falsch platzieren, springt das Diagramm nicht an. Ein Tipp von meiner Seite: Gewöhnen Sie sich ein konsequentes System an, denn eine sinnvolle optische Struktur hilft bei der Fehlersuche. Durch entsprechendes Einrücken mittels Tabulatoren (beim Mac im Web Viewer **Alt** + **→**) können Sie die Parameterebene optisch widerspiegeln. Klammern auf gleicher Parameterebene sollten Sie auch auf der gleichen vertikalen Linie setzen. So behalten Sie den Überblick.
- Gleiches gilt für die Kommasetzung: Wenn Ihr Diagramm nicht mehr angezeigt wird und Sie auf den ersten Blick keinen Fehler entdecken, überprüfen Sie die Kommasetzung am Ende der Zeilen. Oft führt man Änderungen durch und vergisst, auch die Anzahl der Kommas zu ändern.
- Bei RGB- bzw. RGBA-Werten darf sich zwischen dem Text und der Klammer kein Leerzeichen befinden.
- Speichern Sie den Web-Viewer-Inhalt besonders vor größeren Änderungen in einer Textdatei. Wenn Ihr Diagramm nicht mehr funktioniert, können Sie die letzte intakte Version wieder herstellen. Außerdem können Sie so eine Musterdatei erstellen, die Sie für neue Diagramme einfach in den Web Viewer kopieren und dann anpassen.
- Farben, Diagrammtyp, Markierung von Bereichen, Benennung der Achsen – alles ist im Layout steuerbar. So können Sie ein Standard-Diagramm erstellen, das Sie in Ihre Lösungen hineinkopieren und mit einem eigens dafür vorgesehenen Layout in Echtzeit modifizieren können.

Die Beispieldatei enthält, wie gesagt, den kompletten und kommentierten Code, den Sie als Muster verwenden und mithilfe der vorgestellten Syntax leicht modifizieren können. Ich hoffe, dieser Artikel ist Ihnen eine kleine Hilfe auf der Suche nach einer Möglichkeit, flexible und universell einsetzbare Diagramme zu erstellen. In diesem Sinne: Be inspired und bis neulich! ♦

Fußnote

¹ <http://api.highcharts.com/highcharts>

Links

Highcharts.js <http://code.highcharts.com/4.0.1/highcharts.js>
jQuery.js <http://code.jquery.com/jquery-2.1.1.min.js>
Highcharts API <http://api.highcharts.com/highcharts>



Das FileMaker Magazin

- Einzige, deutschsprachige Fachzeitschrift zu FileMaker
- Wissen aus erster Hand von anerkannten FileMaker Fachautoren
- Große Themenvielfalt für Anwender, Entwickler und Fortgeschrittene

Exklusiv für Premium-Abonnenten

- Sechs FMM Ausgaben pro Jahr
- Kostenlose Nutzung des Abonnentenbereichs auf www.filemaker-magazin.de
- PDF-Online-Archiv mit allen bisher erschienenen Ausgaben
- Jede Ausgabe mit kostenlosen Beispieldateien und Zusatzinfos zum Download

Unser Service

- Aktuelle Neuheiten, Tipps und Infos, Kleinanzeigen und vieles mehr jederzeit online auf unseren Webseiten
- Hilfe bei allen Fragen zu FileMaker im FMM Forum
- Kompetente Beratung zum Kauf von FileMaker Lizenzen: Einfach anrufen +49 (0)40 589 65 79 70.

Hier finden Sie **Aktuelles** zu FileMaker **Produkten**, egal ob Sie kaufen, mieten oder sich einfach informieren möchten.

Eine kostenlose **Leseprobe** des FileMaker Magazins erhalten Sie, wenn Sie hier klicken.

Wenn Sie sich für ein **FileMaker Magazin Abo** interessieren, klicken Sie bitte hier!