

Scripting

Mehr zum Thema Spreadsheets

Bevor sie diese Seite lesen, empfehlen wir folgende Seite zum [Thema Spreadsheets](#) zu lesen.

- [Grundlagen](#)
 - [Scripting zur Manipulation von Spalten](#)
- [Manipulationen von Spalten](#)
 - [String Manipulation](#)
 - [Zahlen Manipulation und Rechnen \(arithmetische Funktionen\)](#)
 - [Rechnen](#)
 - [Datumsfunktionen](#)
 - [Bedingungen / Boolean expressions \(IF/ELSE\)](#)
- [Zugriff auf Spalten / Variablen mit Sonderzeichen](#)
- [Ausgeschlossene Freemarker Build-Ins und Direktiven](#)

Für komplexere Umwandlungsaufgaben hat Synesty Studio die Scriptsprache [Freemarker](#) integriert, die es erlaubt komplexere Formeln, Funktionen und Logik auszuführen. Damit hat man die Möglichkeit, ähnlich wie mit Funktionen aus Tabellenkalkulationsprogrammen durch Bedingungen (if/else), Schleifen oder Manipulation von Zeichenketten Einfluss auf die Werte zu nehmen.

Folgende Funktionen (sog. [Built-Ins](#) oder Direktiven) werden am häufigsten benötigt:

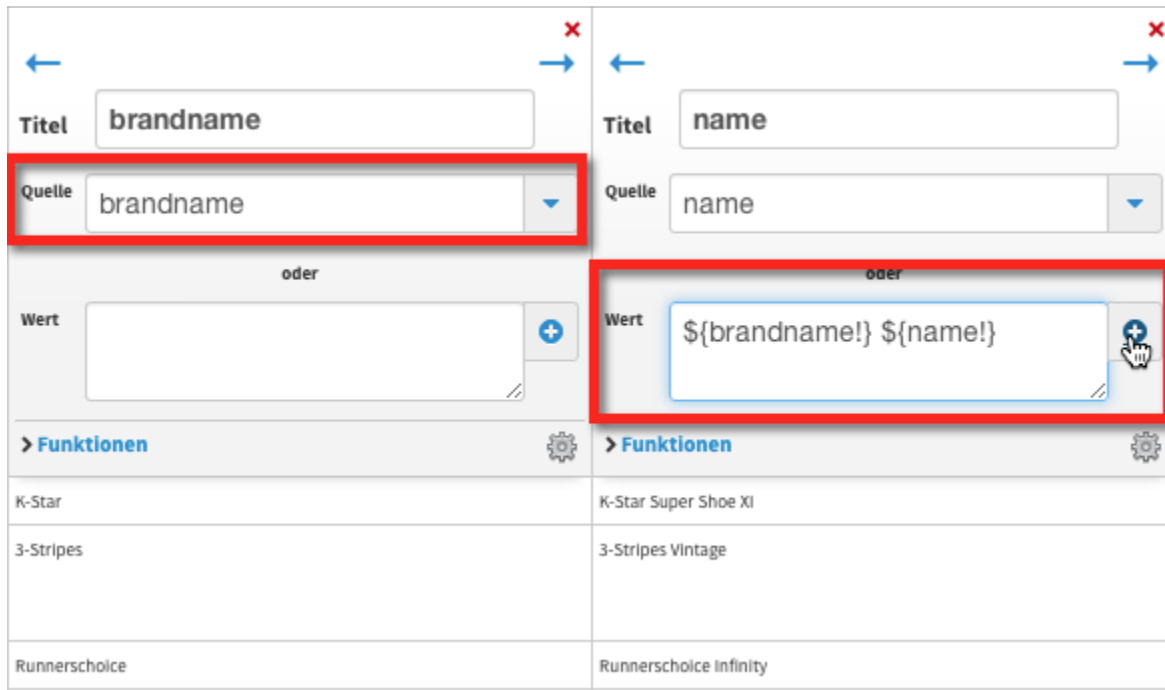
- [if / else](#)
- [String Manipulationen](#)
- [Listen und Sequences](#)
- [Mathematische Funktionen und Zahlen](#)
- [Datumsfunktionen](#)

Grundlagen

Im SpreadsheetMapper gibt es zwei Möglichkeiten einen Input Wert für eine Spalte festzulegen:

- [Quelle](#)
- [Wert](#)


Folgender Screenshot zeigt, wie ein Spaltenwert durch die Auswahlbox *Quelle* und einmal mit dem *Value Feld* gefüllt wird.










Titel brandname	Titel name
Quelle brandname	Quelle name
oder	oder
Wert	Wert \${brandname!} \${name!}
> Funktionen	> Funktionen
K-Star	K-Star Super Shoe XI
3-Stripes	3-Stripes Vintage
Runnerschoice	Runnerschoice infinity


Scripting zur Manipulation von Spalten

Scripting ist hauptsächlich im **Wert-Feld**, aber auch im **Skript-Feld** anwendbar und erfolgt durch die Eingabe von sogenannten Freemarker Ausdrücken.

Wert  `${brandname!} ${name!}`

▼ Funktionen 

-  Suchen & Ersetzen
-  Rechenoperation
-  Runden
-  Datumsfunktionen
-  Querverweis
-  Mapping

Skript 

Skript:

```
<#if value! != "">${value}
<#else>Foo</#if>
```

Super Shoe XI

Sonderfall: Skript-Feld



In den meisten Fällen werden sie das Wert-Feld verwenden. Das Skript-Feld ist nur eine Möglichkeit per Scripting den finalen Wert nach der Ausführung aller vorheriger Funktionen zu verändern. Im Skript-Feld steht der aktuelle Wert in Form der Variable `${value!}` zur Verfügung.

Folgende Beispiele können mit unserem [Beispiel-Spreadsheet](#) ausprobiert werden.

Beispiel

```
${name}
```

```
<#if name?contains("a")>
  Die Spalte 'Name' beinhaltet ein A.
<#else>
  Es kommt kein A in der Spalte A vor.
</#if>
```

 Jeder Source-Spaltenname steht auch als Freemarker-Ausdruck für das Value-Feld zur Verfügung. Eine Liste verfügbarer Namen zum leichten Einfügen erhalten sie durch Klick auf das  Zeichen neben dem Value-Feld. Das Ausrufezeichen am Ende von `${name!}` weist Freemarker an **null-Werte** als leeren String (") darzustellen.

Manipulationen von Spalten

i Alle folgenden Code-Beispiele beziehen sich auf das Wert-Feld einer jeden Spalte. Zum ausprobieren kopieren sie den entsprechenden Code-Block einfach in das Wert-Feld und drücken den *Vorschau anzeigen* Button.

String Manipulation

Freemarker besitzt viele sog. **built-ins** zur Manipulation von Strings, also Zeichenketten.

Folgender Ausdruck gibt unseren `$(name!)` in GROßBUCHSTABEN aus:

```
$(name!|upper_case)
```

Das Fragezeichen `|` wendet das "built-in" `upper_case` auf die Variable `name` an.

i Mehrere Funktionen gleichzeitig

Man kann mehrere Freemarker built-ins hintereinander hängen ("chain").

Beispiel:

Beispiel:

Nur den Text zwischen den Tags `<p>` und `</p>` ausgeben:

```
$( "Test 123 <p>test</p>" | keep_after(' <p>' ) | keep_before(' </p>' ) }
```

Ausgabe: test

Zahlen Manipulation und Rechnen (arithmetische Funktionen)

Freemarker besitzt **built-ins** zur Manipulation von Zahlen.

```
$(price|number|string.currency)
```

Die Ausgabe ist `$42.00` Die verwendete Locale ist die des Servers, d.h. `en_US`. Das `?number` Built-In ist notwendig, um aus einem String erst eine Zahl zu machen. Wenn die entsprechende Variable bereits eine Zahl ist, ist `?number` nicht notwendig.

Um den Preis in deutscher Locale auszugeben wäre es

```
<#setting locale="de_DE">
$(price|number|string.currency)
```

Die Ausgaben wäre bei `de_DE` **42,00 €**



Punkt als Dezimaltrenner

Hinweis: Freemarker erwartet bei mathematischen Funktionen immer einen Punkt als Dezimaltrenner. D.h. falls Zahlen mit Komma als Dezimaltrenner existieren, muss das Komma erst durch einen Punkt ersetzt werden. Eine Möglichkeit wäre, dazu die `?replace` Funktion zu benutzen:

```
#{OrderTotalGross?replace(",",".")?number + OrderShippingCosts?replace(",",".")?number}
```

Rechnen

Mit Freemarker kann man auch rechnen. Lesen Sie dazu mehr über [arithmetische Funktionen](#).

Beispiel:

```
#{1 + 2}
# gibt 3 aus

#{spalte1?number + spalte2?number}
# gibt die Summer von zwei Spalten aus.
```

Datumsfunktionen

[Datumsmanipulation](#) ist auch mit Freemarker möglich.

```
<#assign purchaseDate = .now>
#{purchaseDate?string('yyyy-MM-dd HH:mm:ss zzzz')}
```

Ausgabe ist z.B. *2013-07-12 19:19:15 Central European Summer Time*

Datumsangaben Vergleichen

Vergleich zweier Datumsangaben

```
<#if ("2020-12-11 20:09:42"?datetime('yyyy-MM-dd HH:mm:ss') > "2020-12-10 10:09:42"?datetime('yyyy-MM-dd HH:mm:ss')) >.
Das erste Datum ist neuer.
<#else>
Das erste Datum ist älter.
</#if>
```

Siehe dazu auch das [Cookbook zu Datumsarithmetik / Rechnen mit Datumsangaben](#).

Bedingungen / Boolean expressions (IF/ELSE)

Wenn/Dann Logik wird durch sog. Boolesche Ausdrücke realisiert. Damit lassen sich Spalteninhalte in Abhängigkeit von anderen Quellspalten manipulieren.

Zum Beispiel könnte man sagen, dass die Versandkosten 0 EUR betragen sollen, wenn der Preis größer 100 EUR ist. Ansonsten könnten die Versandkosten 5 EUR betragen.

Beispiel Versandkosten

```
<#if (price?number >= 100)>0 EUR<#else>5 EUR</#if>
```

Beispiel Prüfung, ob eine Zeichenkette ein bestimmtes Zeichen enthält

```
<#if name?contains('u')>
This is a name containing the letter 'u'
<#else>
No 'u' contained
</#if>
```

IF/ELSE mit String-Vergleichen:

```
<#if brandname! == ''>
Brandname is empty.
<#else>
${brandname!}
</#if>
```

Bei Vergleichen von Zeichenketten sollten Sie auf Leerzeichen achten und diese ggf. mit `?trim` entfernen bzw. "wegschneiden".

```
<#if brandname!?trim == ''>
Brandname is really empty (spaces trimmed).
<#else>
${brandname!}
</#if>
```

Prüfen, ob Zeichenkette mit einem bestimmten Prefix startet (siehe [starts_with](#))

```
<#if brandname!?starts_with('myprefix')>
Yes, brandname starts with myprefix
<#else>
No, brandname does not start with the prefix.
</#if>
```

Zugriff auf Spalten / Variablen mit Sonderzeichen

Wenn man im SpreadsheetMapper mit Freemarker auf Spalten mit Sonderzeichen wie z.B. Bindestrich zugreifen will, dann muss man eine andere Syntax verwenden.

Beispiel:

```
${meine-spalte}
```

Dieser Ausdruck bringt Freemarker durcheinander, weil er denkt man müsste (in Worten) "meine minus spalte" rechnen.

Um mit Freemarker auf Variablen (Spalten) mit Sonderzeichen zuzugreifen kannst man eine andere Syntax verwenden z.B.:

```
${row["meine-spalte"]}
```

In der Variable `$(row)` (kommt von Synesty) sind alle Spalten enthalten und man kann mit eckigen Klammern darauf zugreifen.

Ausgeschlossene Freemarker Build-Ins und Direktiven

Es können nahezu alle [Direktiven bzw. Build-Ins](#) der Freemarker Version 2.3.29 verwendet werden. Aus Sicherheitsgründen gibt es einige, wenige Ausnahmen die hauptsächlich im Bereich "[Seldom used and expert built-ins](#)" zu finden sind.

Explizit ausgeschlossen sind die Build-Ins:

- ?api
- ?new

und die Direktiven:

- <#import>
- <#include>