


# Writing Custom JavaScript

There are several places within ViziApps where you can insert JavaScript code to transform data, generate and catch events, and extend functionality. This guide will help you get started. If you aren't familiar with JavaScript, please see the [References](#) section.

## JavaScript in the HTML Header

Text that you insert in the HTML header  is simply included into your app's [HTML header](#) when it runs. JavaScript variables and functions that you define in the HTML header have global scope and may be used throughout your ViziApps app. You are allowed to use `<meta>`, `<link>`, `<style>`, and `<script>` tags. You must enclose all JavaScript within `<script>` tags when inserting code into the HTML Header, like so:

```
<script>
var wholename;
function setWholeName( last, first ) {
    wholename = getFieldValue( last ) + ', ' + getFieldValue( first );
    return wholename;
}
</script>
```

JavaScript code appearing in the HTML Header executes before your app actually runs (note that function definitions don't actually run unless they are called). This may be useful in certain cases, however you shouldn't reference field names until after the app has been initialized. Here is the way to put code in the HTML Header that won't execute until the app is ready:

```
<script>
function appReady() {
    setFieldValue( 'startTimeLabel', (new Date()).toLocaleString() );
}
document.addEventListener("DOMContentLoaded", appReady );
</script>
```

## JavaScript in Field and Data Management Events


There are several places where short snippets of JavaScript can be inserted:



- Button and Image Button field tap event
- Table field row tap event
- Switch and Checkbox field change event

- Picker field selection event
- Data Management, Add Device Action

For each event, there are two separate areas where JavaScript can be entered: the first will be called to execute before the event is triggered, and the second will be called after the event completes. This calling sequence is always sequential, however, be aware that JavaScript is an event-driven language, and it's possible that you may call a function in the code before the event triggers, but the called function may not complete until a later time, even after the event and final JavaScript code is executed.

For any Data Management query, you can also insert JavaScript before or after any data command by  clicking on the **Add Device Action** button. Like the field events, the data management commands and device action JavaScript are executed in sequential order. This capability for inserting JavaScript is especially handy for manipulating data before or after a command - for example, your app may receive separate firstname and lastname fields from a database, but you would rather show a single string such as:

```
setWholeName( 'lastname', 'firstname' );
```

## Access to ViziApps Fields

### Get and Set Field Values

Many ViziApps fields contain a value that can be set or read through several non-coding methods, such as preset values in the field's properties box; a data query that transfers web data into or out of the field; or a user action such as checking a box or moving a slider. JavaScript lets you programmatically get or set the value of these fields just as if a non-coding action had get or set the value. So for a Hidden Field named *lastname* and a Text Label field named *lastNameLabel*, you could have this code:

```
setFieldValue( 'lastNameLabel', 'LAST NAME: ' + getFieldValue( 'lastname' ) );  
alert( 'Field set to ' + getFieldValue( 'lastNameLabel' ) );
```

### Table and Picker Selected Values

ViziApps Table Fields and Picker Fields work a little differently since they store an array of values, not just a single value. You may have defined a Table with two text fields and a hidden field, so within a Table Field named *contactsTable*, you might have defined field names such as *lastname*, *address*, and *userid*. You can get or set the Table Field name, in which case you would be referencing the particular row that the user has last selected, such as:

```
alert( 'The user selected ' + getFieldValue('lastname') + ' at ' +  
getFieldValue('address') );
```

## Get and Set Array Values

There are times where you might want to get or set all of the values held in the Table Field, not just the most recently selected value. In this case, each fieldname represents an array of string values in the order that they appear in the table, so you must set new values with a string array. For example you may have defined a Table Field called *daynameTable* with a variable that is an array of strings named *dayNames*, as in:

```
var dayNames =  
[ 'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday' ];  
setFieldArray( 'dayNameField', dayNames );  
console.log( 'The day names are: ' + getFieldArray( 'dayNameField' ) );  
// result - The day names are:  
Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday
```

## ViziApps built-in functions (APIs)

See the [ViziApps JavaScript Function Reference](#) for a list and description of the functions that are available for you to use. These functions can: get and set ViziApps field values; change the page; get GPS values; and more. Following are examples of a few important functions.

### Page Control

Programatic change to a new page is often done as a Device Action within Data Management, or as a response to a user tap event before a query.

```
v.gotoPage( 'page_name' );
```

### Alert

To popup an alert named 'alert\_field\_name':

```
v.showAlert('alert_field_name', 'Important, Please Read:', 'This is the alert  
message');
```

### Loading Indicator

To show and hide a loading indicator to the user:

```
// shows the loading indicator for 5 seconds
```

```
function doneLoadingHandler() {  
    v.hideLoading();  
}  
v.showLoading();  
setTimeout( doneLoadingHandler, 5000 );
```

## Hide and Show a Field

ViziApps fields can be hidden or made visible:

```
function showHideField( fieldName, showConditionField ) {  
    var showit = v.getFieldValue( showConditionField );  
    if ( showit === false || showit === 'false' || showit === 'off' || showit  
    === 0 )  
        v.hideField( fieldName );  
    else  
        v.showField( fieldName );  
}  
showHideField( 'optionButtonField', 'optionConditionHiddenField' );
```

## References

[W3Schools JavaScript Tutorial](#)

[W3Schools JavaScript Reference](#)

[Mozilla Developer Network JavaScript Guide](#)

[Mozilla Developer Network JavaScript Reference](#)

[Online JavaScript Regular Expression Tester](#)

[O'Reilly JavaScript Cookbook](#)

[Apache Cordova Documentation](#)

From:

<https://viziapps.com/dokuwiki/> - **ViziApps Help Wiki**

Permanent link:

[https://viziapps.com/dokuwiki/writing\\_custom\\_javascript](https://viziapps.com/dokuwiki/writing_custom_javascript)

Last update: **2015/01/14 12:04**

