



Quark

Writing JavaScript Samples for QuarkXPress

Sarbjit Singh

Agenda

- ▶ Getting Started
- ▶ Projects and Layouts
- ▶ Project Assets
- ▶ Alerts and Confirm Dialogs
- ▶ User Input
- ▶ App Properties
- ▶ QXML DOM
- ▶ PDF Handling
- ▶ Disk File Operations
- ▶ Third Party JS Libraries
- ▶ The JS Sample Framework

Getting Started

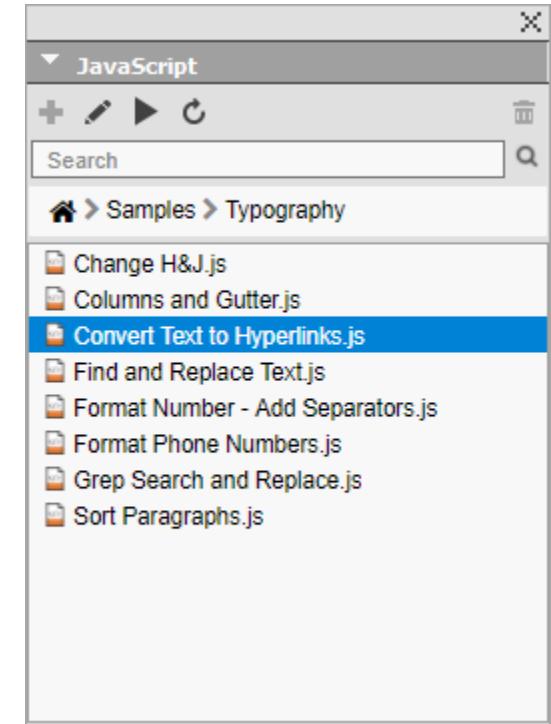
JavaScript Palette

- ▶ Create Scripts
- ▶ Edit Scripts
- ▶ Run Scripts
- ▶ Delete Scripts
- ▶ Copy / Paste Scripts



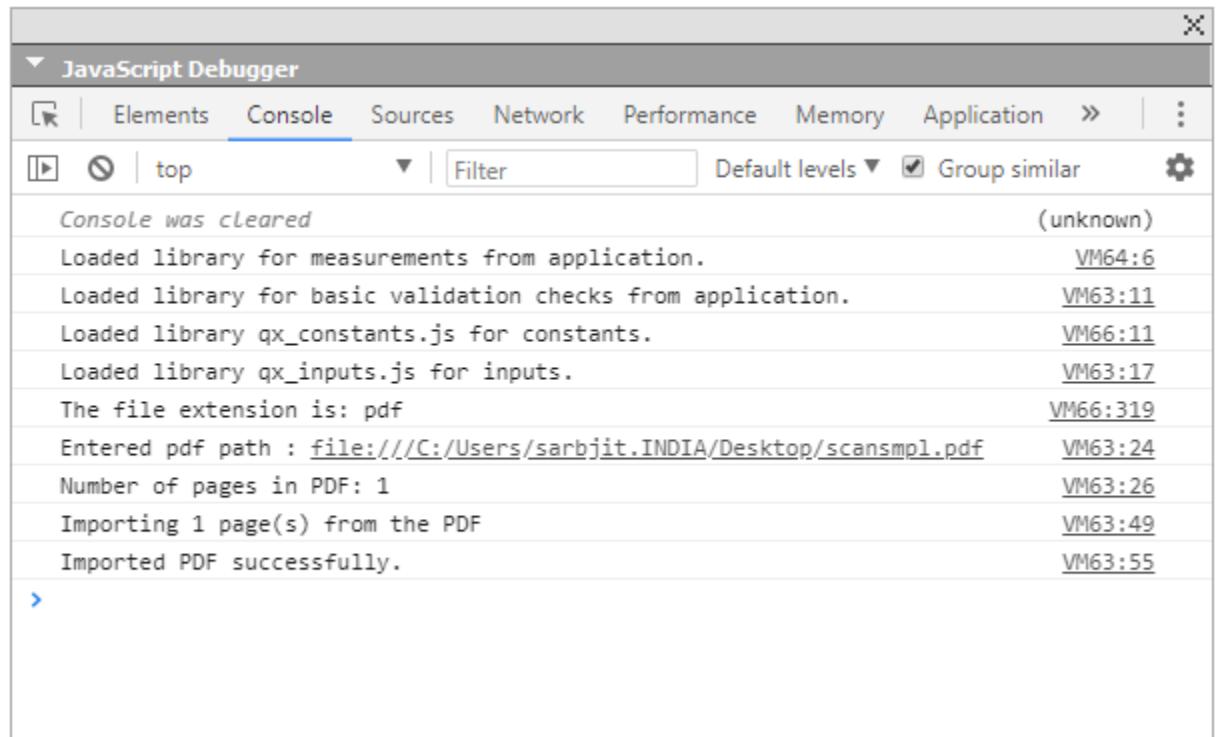
Note:

You can not edit / delete default scripts.



JavaScript Debugger Palette

- ▶ Test Scripts
- ▶ Debug Scripts
- ▶ Console Logging
- ▶ One Time Operations



The screenshot shows the 'Console' tab of the JavaScript Debugger palette. The interface includes tabs for Elements, Console, Sources, Network, Performance, Memory, Application, and more. A filter bar at the top allows for filtering log entries. The main area displays a list of log messages:

- Console was cleared (unknown) VM64:6
- Loaded library for measurements from application. VM64:6
- Loaded library for basic validation checks from application. VM63:11
- Loaded library qx_constants.js for constants. VM66:11
- Loaded library qx_inputs.js for inputs. VM63:17
- The file extension is: pdf VM66:319
- Entered pdf path : `file:///C:/Users/sarbjit.INDIA/Desktop/scansmpl.pdf` VM63:24
- Number of pages in PDF: 1 VM63:26
- Importing 1 page(s) from the PDF VM63:49
- Imported PDF successfully. VM63:55



Tip:

You may attach the Debugger to JavaScript UI, if you open it with the Ctrl/Cmd key pressed.

App Package

- ▶ Default Scripts path:

Windows: C:\Program Files\Quark\QuarkXPress 2018\XTensions\JSServices\Samples

Mac: QuarkXPress 2018.app > Contents > SharedSupport > Scripts > JSServices > Samples

- ▶ User Scripts path:

Windows: C:\Users\[USER_NAME]\Documents\Quark\QuarkXPress 2018\js

Mac: [USER_DOCUMENTS]\Quark\QuarkXPress 2018\js



Note:

1. Do not edit the files inside the package. Always create a copy first.
2. The files inside the “Dependencies” folder are NOT listed in the palette.

QX.js API NameSpaces

- ▶ The “app” object
- ▶ The “fs” object

Demo

- ▶ Creating a project from console.

app.createProject();



Demo



Quark

Projects and Layouts

Creating a Project

- ▶ `createProject(projectName, layoutName, layoutMediaType, printLayoutOptions)`



Note:

1. Digital Layouts are NOT fully supported yet.



Opening an existing Project

- ▶ `var myProject = app.openProject("c:\\temp_c\\demo\\demo.qxp");`



Note:

1. Observe how we have formatted the file path.



Creating a Layout in a Project

- ▶ Get the Current Project:
- ▶

```
var activeProject = app.activeProject();
```

- ▶ Add a Layout:
- ▶

```
activeProject.createLayout(layoutName, layoutType, layoutOptions);
```



Note:

1. Digital Layouts are NOT fully supported yet.



Other Project Operations

- ▶ `.closeProject()`
- ▶ `.saveProject()`
- ▶ `.saveAsProject()`
- ▶ `.getAssets()` *
- ▶ `.getLocation()`

* More on this later

Other Layout Operations

- ▶ `.delete()`
- ▶ `.duplicate()`
- ▶ `.exportLayoutAsPDF()`
- ▶ `.getLocation()`
- ▶ `.getDOM() *`
- ▶ `.getPrintLayoutOptions()`

* More on this later

Get Project Assets

- ▶ Char Styles
- ▶ Para Styles
- ▶ Color
- ▶ Underline Styles
- ▶ H And J
- ▶ Lists
- ▶ Output Styles
- ▶ Fontset
- ▶ Hanging Char Class
- ▶ Hanging Char Set
- ▶ Bullets and Numbering
- ▶ Outline Styles
- ▶ Conditional Styles
- ▶ Binding Styles
- ▶ Table Styles
- ▶ Footnote Styles
- ▶ Footnote Container Styles
- ▶ Text Background Styles
- ▶ Item Styles



Note:

Only some of these assets can be applied as classes in QML, for now.

Get Project Assets... contd

- ▶ `activeProject.getAssets(app.constants.assetTypes.kAssetParaStyle);`



Note:

Use “`app.constants.assetTypes`” to specify which assets you need.



Alerts and Confirm Dialogs

Show Alerts

- ▶ `alert("Hello");`
- OR
- ▶ `app.dialogs.alert("Hello");`

Confirm (OK/Cancel)

- ▶ `confirm("Are you sure?");`
- OR
- ▶ `app.dialogs.confirm("Are you sure?");`



User Input

Show Prompt

- ▶ `prompt("What is your name");`
- OR
- ▶ `app.dialogs.prompt("What is your name");`



Demo



Quark

App Properties

Standard App Information

- ▶ **app.dir** : path to QuarkXPress folder.
- ▶ **app.language** : Code for Current UI Language
- ▶ **app.name** : "QuarkXPress"
- ▶ **app.version** : Current version



Get Script Folders

- ▶ **app.getAppScriptsFolder()** : path to Scripts folder in the App folder.
- ▶ **app.getUserScriptsFolder()** : path to Scripts folder in the User folder.



QXML DOM

What is DOM

The **Document Object Model (DOM)** is a programming API for HTML and XML documents. It defines the **logical structure of documents** and the way a document is accessed and manipulated. In the DOM specification, the term "document" is used in the broad sense - increasingly, XML is being used as a way of representing many different kinds of information that may be stored in diverse systems, and much of this would traditionally be seen as data rather than as documents. Nevertheless, XML presents this data as documents, and the DOM may be used to manage this data.

What is QXML DOM

The QXML DOM (**QuarkXPress Markup Language Document Object Model**) is the JavaScript programming interface for accessing a QuarkXPress layout's object model. QXML DOM is based on the World Wide Web Consortium (W3C) **Custom Elements** specification, which is a part of the W3C DOM specification. Every QXML DOM element is a custom HTML5 element that represents a corresponding QuarkXPress layout object with a custom tag name along with its properties and styling.

As it is based on the W3C HTML standard, you can expect to be immediately productive with scripting the QXML DOM if you are already familiar with scripting the HTML DOM. Like HTML DOM elements, all QXML custom elements are styled via custom CSS properties based on the W3C CSS Specification. The HTML 'style' attribute is used to specify the inline CSS properties to style QXML custom elements.

Structure

```
▼ <qx-layout layout-id="1" layout-type="print" layout-name="MyLayout">
  <qx-layer layer-id="-1" layer-name="Default" style="--qx-visibility:visible;--qx-keep-
    runaround:false;--qx-locked:false;--qx-suppress-output:false;--qx-color:
    rgb(127,169,247);"></qx-layer>
  <qx-layer layer-id="1" layer-name="TextLayer" style="--qx-visibility:visible;--qx-
    keep-runaround:true;--qx-locked:false;--qx-suppress-output:false;--qx-color:
    rgb(231,106,144);"></qx-layer>
  <qx-layer layer-id="2" layer-name="ImageLayer" style="--qx-visibility:visible;--qx-
    keep-runaround:true;--qx-locked:false;--qx-suppress-output:false;--qx-color:
    rgb(159,217,139);"></qx-layer>
  ► <qx-spread spread-index="1">...</qx-spread>
</qx-layout>
```

Structure...

```
<qx-spread spread-index="1">
  <qx-page page-index="1" page-position="right" applied-master-page="A-Master A">
    </qx-page>
  ><qx-box box-id="10" box-content-type="text" box-name="Box10" style="--qx-
background-color:none;--qx-transform:rotate(0deg) skew(0deg);--qx-border-radius:0in;-
--qx-corner-shape:rectangle;--qx-layer-name:TextLayer;--qx-page:1;--qx-maintain-
aspect-ratio:false;--qx-position:absolute;--qx-top:0.5in;--qx-left:0.5in;--qx-bottom:0.8229in;--qx-right:6.2222in;--qx-suppress-output:false;--qx-wrap-flow:auto;--qx-
border-color:Black;--qx-border-gap-color:none;--qx-border-opacity:1;--qx-border-
shade:1;--qx-border-width:0in;--qx-column-count:1;--qx-first-baseline-minimum:ascent;-
--qx-text-flip-horizontal:false;--qx-text-flip-vertical:false;--qx-column-gap:0.167in;--qx-first-baseline-offset:0in;--qx-run-text-around-all-sides:true;--qx-
align-items:start;--qx-text-transform:rotate(0deg) skew(0deg);--qx-padding:0pt;">...
</qx-box>
```

Structure...

```
▼ <qx-story story-id="0" style="--qx-writing-mode:horizontal-tb;">>
  ▼ <qx-p class="pr-Normal" style="--qx-text-align:center;">
    <qx-span style="--qx-font-size:24;" span-id="span_0">The </qx-span>
    <qx-span style="--qx-font-weight:bold;--qx-font-size:24;" span-id="span_1">
      QuarkXPress</qx-span>
    <qx-span style="--qx-font-size:24;" span-id="span_2"> </qx-span>
    <qx-span style="--qx-font-weight:bold;--qx-color:Cyan;--qx-font-size:24;" span-id="span_3">Aliens</qx-span>
  </qx-p>
</qx-story>
```

Structure...

```
<qx-img src='file:///C:/Users/sarbjit.INDIA/Desktop/JS - Demo/Pictures/  
alien2.jpg' style='--qx-offset-across:0in;--qx-offset-down:0in;--qx-opacity:1;--  
qx-transform:rotate(0deg) skew(0deg) scaleX(1) scaleY(1);'></qx-img>
```



Disk File Operations

The fs object

- ▶ .copyFile / .copyFileSync
- ▶ .existsSync
- ▶ .mkdir / .mkdirSync
- ▶ .readDir / .readDirSync
- ▶ .readFileSync
- ▶ .rmDir / .rmDirSync
- ▶ .stat / .statSync
- ▶ .unlink / .unlinkSync
- ▶ .writeFileSync



Note:

1. Refer to node.js documentation for more details about fs.
2. We only implemented a part of node features.

The JS Sample Framework

Helpful Libraries

- ▶ qx_constants.js
- ▶ qx_create_box.js
- ▶ qx_inputs.js
- ▶ qx_measurements.js
- ▶ qx_validations.js



Note:

1. You can see which libraries are being used, at the top in sample scripts.
2. Do not edit sample functions, copy and create your own.

Demo

- ▶ Use a library function for user input.

getValidNumericInput()

```
/* Function to get a Valid Numeric Input from user  
Asks for an Input from the user, showing the string message sent, validates it  
strPrompt: The string to show in the prompt dialog  
defaultValue: The string to show as the default value in the prompt dialog. Must be in  
points if Units are to be handled  
minValue: Decides the Minimum Input, Infinity by default. Must be in points if Units  
are to be handled  
maxValue: Decides the Maximum Input, Infinity by default. Must be in points if Units  
are to be handled  
isIntegerMust: Decides if fractions are allowed  
inputUnits: The units for user Input  
  
Returns: null if the user cancels the input || The valid user input  
*/  
function getValidNumericInput(strPrompt, defaultValue = "", minValue = Infinity,  
maxValue = Infinity, isIntegerMust = false, inputUnits = "") {
```



Third Party JS Libraries

Include external files

- ▶ `app.importScript(filePath);`



Demo



References

- ▶ QX.js API Reference: <http://bit.ly/qaxisapi>
- ▶ QXML API Reference: <http://bit.ly/qxmlapi>
- ▶ Visual Studio Code:
<https://code.visualstudio.com/download>
- ▶ JavaScript / HTML / CSS Tutorial:
<https://www.w3schools.com/>



Quark

Questions?

Visit us here please:

<https://www.facebook.com/groups/quarkxpress/>

or

<http://forums.quark.com>

Thank You



Quark