

QML & Javascript: No Browser Required

Jen Trieu
Michael Tims



Agenda

- Native apps. Why?
- Overview of Qt and QML
- How to use JavaScript skills to build native apps

Why build native apps?

- Work online & offline
- Access hardware and sensors
 - GPS, camera, Bluetooth, NFC, etc
- Great performance
- Publish apps to stores

The platform Challenge

- Lots of platforms
- Lots of devices
- Unique development patterns
 - Languages
 - Frameworks
 - IDEs
 - Workflows



You can build *native* apps
with your JavaScript skills... and Qt!

What is Qt and QML?

And how is it related to JavaScript?

Qt and QML

- Cross platform framework
- Build native apps
- Same source code compiled for each platform
- Powered by C++ on the backend (very fast)
- Exposed through QML (based on JavaScript)
- ArcGIS Runtime SDK for Qt
 - Mapping API provided by Esri
 - Brings the power of ArcGIS to your devices



QML

Highly readable
JSON/CSS-like
syntax

Declarative UI
elements

Imperative
JavaScript
code to handle
events

```
Rectangle {  
    MapView {  
        id: mv  
        anchors.fill: parent  
        Map {  
            id: map  
            BasemapStreetsVector {}  
        }  
    }  
    Button {  
        anchors {  
            left: parent.left  
            top: parent.top  
        }  
        text: "Zoom to Hawaii"  
        enabled: map.loadStatus === Enums.LoadStatusLoaded  
        onClicked: {  
            var point = ArcGISRuntimeEnvironment.createObject("Point", {  
                x: -157.564,  
                y: 20.677,  
                spatialReference: SpatialReference.createWgs84() } );  
            mv.setViewpointCenterAndScale(point, 4000000.0);  
        }  
    }  
}
```

ArcGIS
Runtime

Dynamic
property
binding

JavaScript & QML

- If you understand JSON and JavaScript, this will come naturally
- Call JavaScript functions from QML
- Bind properties to JavaScript expressions
- Implements 5th edition of ECMA-262
- Access to the language standard types and functions
 - Object, Array, Math, and Date
- JS is integrated in Qt Creator IDE
 - Intellisense in IDE
 - qmlint syntax tool
- Major Differences from the browser?
 - No DOM or window object
 - No Dojo, jQuery, or other frameworks

JavaScript Expressions



JavaScript expressions

- Bind properties to JS expressions
- Update UI automatically
- Any JS expression (no matter how complex) may be used in a property binding definition

Change UI depending on ambient light

```
MapView {  
  Map {  
    id: map  
    Basemap {  
      // Nest an ArcGISVectorTiledLayer Layer i  
      ArcGISVectorTiledLayer {  
        url: lightGrayBasemapUrl  
        visible: isAmbientLightBright  
      }  
      ArcGISVectorTiledLayer {  
        url: darkGrayBasemapUrl  
        visible: !isAmbientLightBright  
      }  
    }  
  }  
  GraphicsOverlay {  
    id: graphicsOverlay  
    renderer: isAmbientLightBright ? lightRender
```

Light Theme

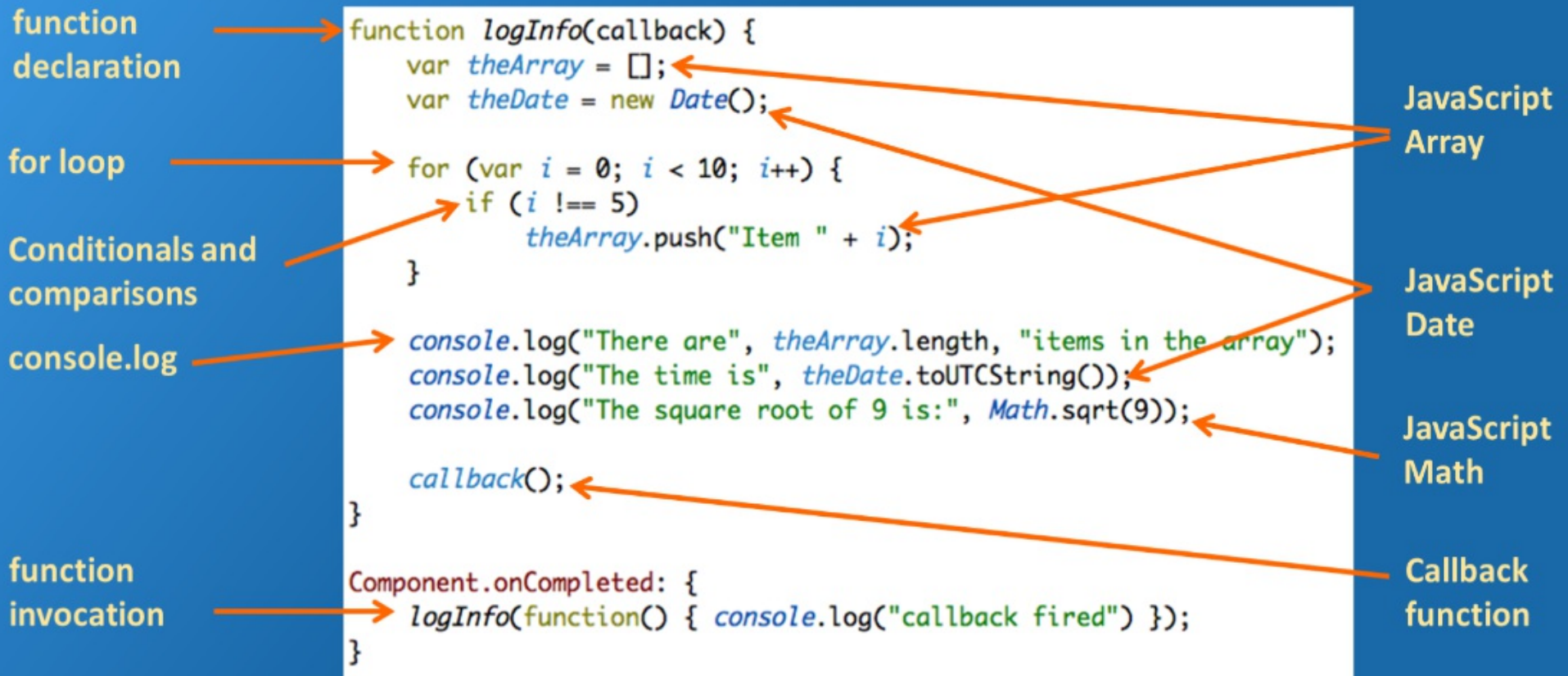
JavaScript Functions



JavaScript functions

- QML is declarative, but... You can (and will need to) call JavaScript functions from QML
 - Declare a visual component in QML (e.g. Button)
 - Write imperative JavaScript code to respond to an event
- "var" basic type used in functions (same as JavaScript var)
 - Can store numbers, strings, objects, arrays and functions
- Syntax based off ECMAScript 5th edition
 - <http://doc.qt.io/qt-5/ecmascript.html>

JavaScript functions



Importing standalone JavaScript

Red

QML file

helper.js

```
import QtQuick 2.6
import QtQuick.Controls 1.4
import QtQuick.Controls.Styles 1.4
import Esri.ArcGISExtras 1.1
import "helper.js" as JSHelper

Rectangle {
    id: rect
    property real scaleFactor: System.displayScaleFactor

    color: "red"
    anchors.fill: parent

    Text {
        id: text
        anchors.centerIn: parent
        font.pixelSize: 20 * scaleFactor
```


Importing standalone JavaScript

Red

QML file

helper.js

randomColor.js

```
import QtQuick 2.6
import QtQuick.Controls 1.4
import QtQuick.Controls.Styles 1.4
import Esri.ArcGISExtras 1.1
import "helper.js" as JSHelper

Rectangle {
    id: rect
    property real scaleFactor: System.displayScaleFactor

    color: "red"
    anchors.fill: parent

    Text {
        id: text
        anchors.centerIn: parent
        font.pixelSize: 20 * scaleFactor
```

Asynchronous Programming with QML



Asynchronous Programming with QML

- Signal and Handler Event System
- The event is a signal (ex: mouseClicked)
- The signal is responded to through a signal handler
 - Signal handler is the name of the signal with the "on" prefix (ex: onMouseClicked)
 - Write JavaScript to perform some procedure when a signal is emitted

```
MapView {  
    // Signal handler for mouse click event on the map view  
    onMouseClicked: {  
        console.log("you clicked at: ",mouse.x, mouse.y )  
    }  
}
```

Asynchronous Programming with QML

Button Route task

```
//when the user clicks on the "Solve Route" button
onClicked: {
    if (routeParameters !== null) {
        // set parameters to return directions
        routeParameters.returnDirections = true;

        // clear previous route graphics
        routeGraphicsOverlay.graphics.clear();

        // clear previous stops from the parameters
        routeParameters.clearStops();

        // set the stops to the parameters
        var stop1 = ArcGISRuntimeEnvironment.createObject(
            {geometry: stop1Geometry, name: "Origin"});
        var stop2 = ArcGISRuntimeEnvironment.createObject(
            {geometry: stop2Geometry, name: "Destination"});

        routeParameters.setStops([stop1, stop2]);

        // solve the route with the parameters
    }
}
```

Solve route

Powered by Esri

Loading new QML code

```
Load

import QtQuick 2.6
import QtQuick.Controls 1.4
import QtQuick.Controls.Styles 1.4
import Esri.ArcGISRuntime 100.0
import Esri.ArcGISExtras 1.1

Rectangle {
    property real scaleFactor: System.displayScaleFactor

    // Create MapView that contains a Map
    MapView {
        anchors.fill: parent
        Map {
            id: map
            // Set the initial basemap to Topographic
            BasemapTopographic {}
        }
    }
}
```

How is Esri using QML?

ArcGIS Runtime SDK for Qt



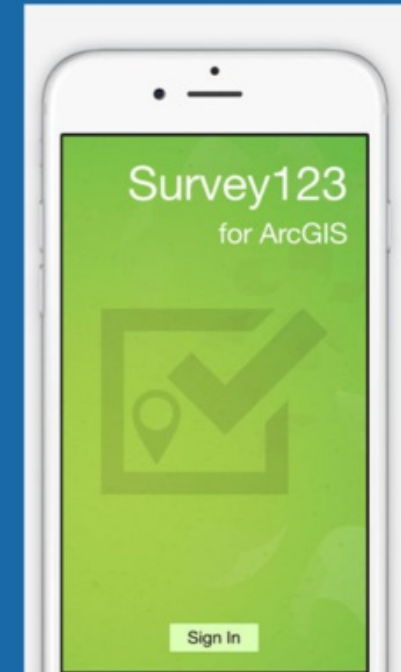
AppStudio for ArcGIS



Runtime Core (for testing)

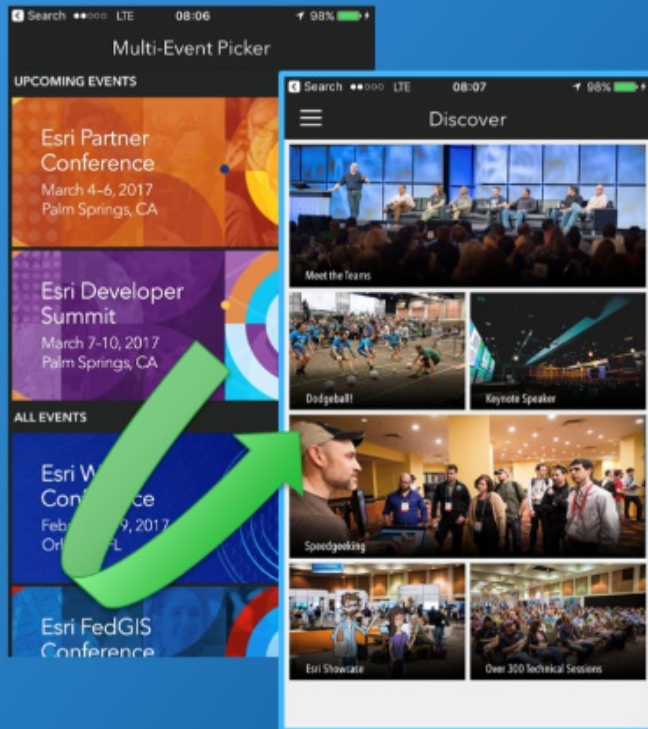


Survey 123

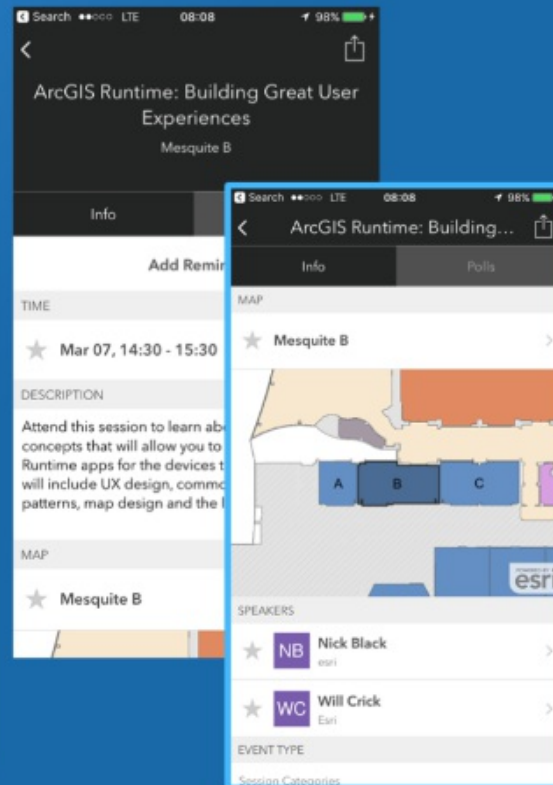


Please Take Our Survey!

Download the Esri Events app
and go to DevSummit

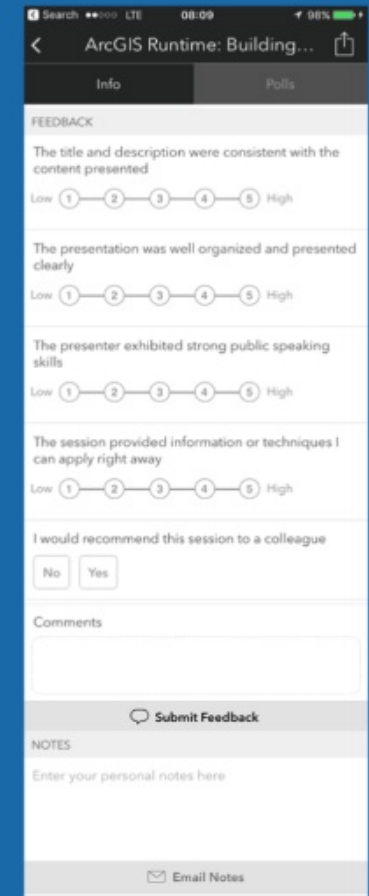


Select the session you attended



Scroll down to the
“Feedback” section

Complete Answers,
add a Comment,
and Select “Submit”





esri

THE
SCIENCE
OF
WHERE