QML and JavaScript: No Browser Required

Jen Trieu

Lucas Danzinger

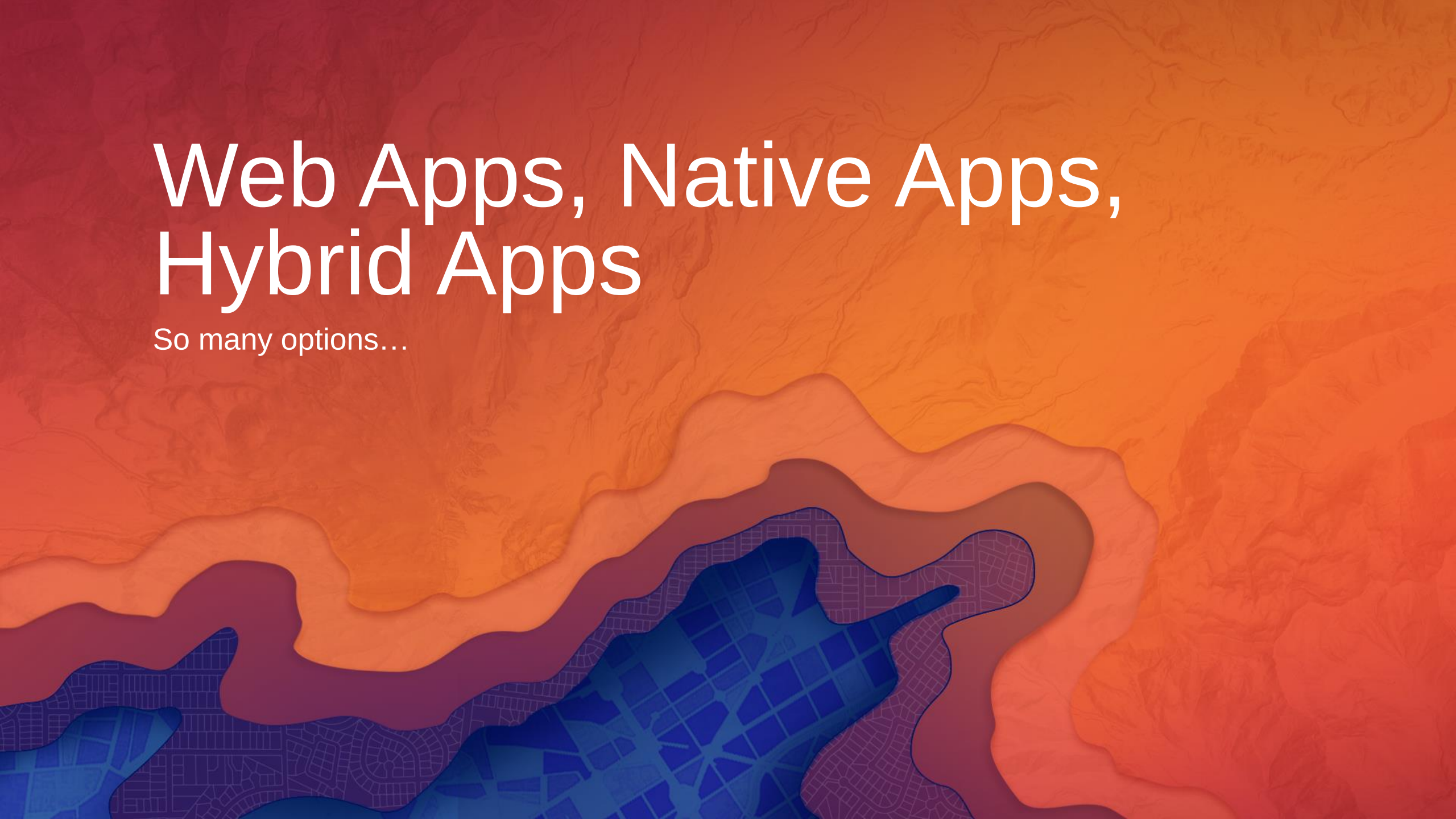
Agenda

Subhead Here

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

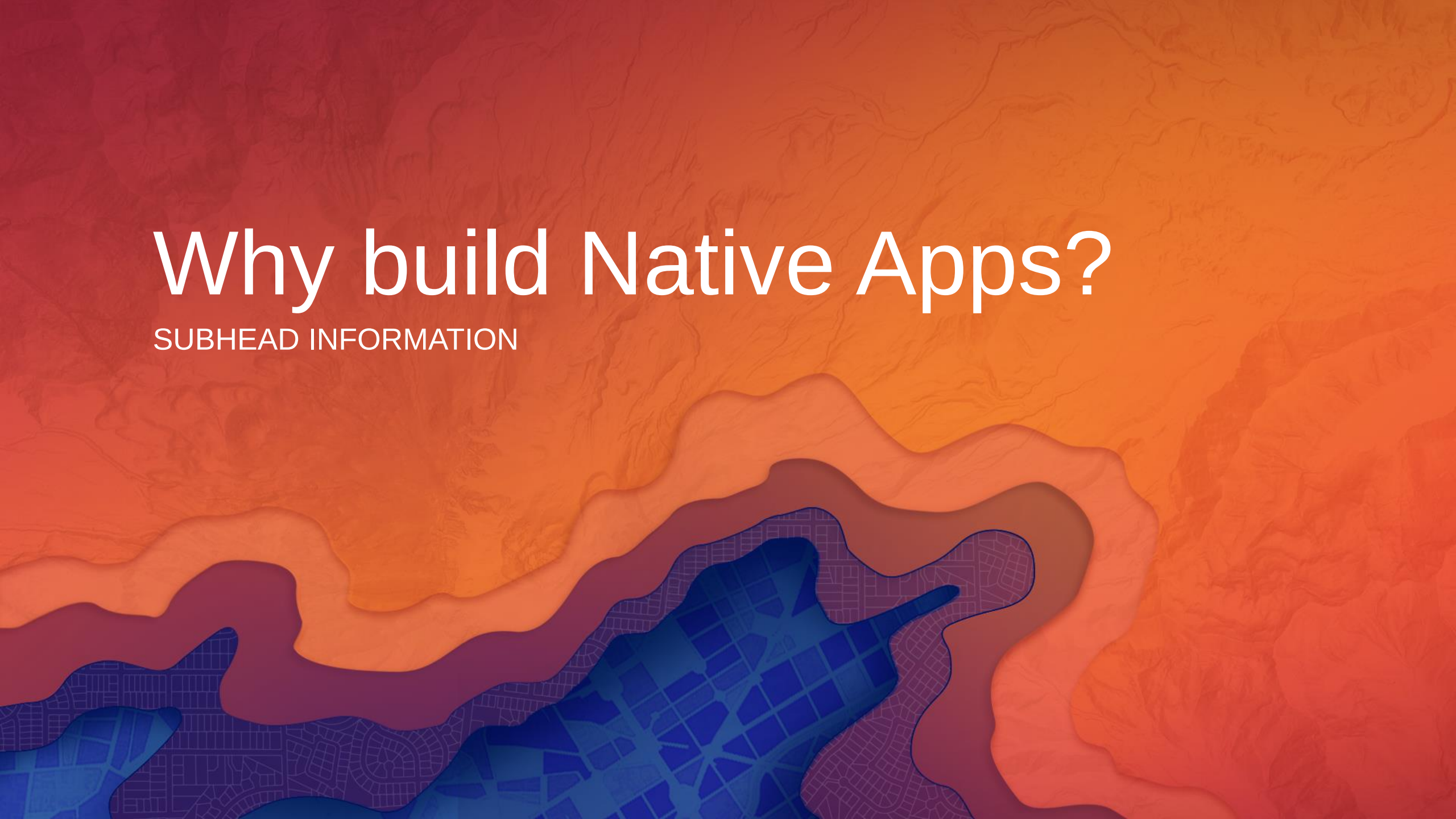
Web Apps, Native Apps, Hybrid Apps

So many options…



Why build Native Apps?

SUBHEAD INFORMATION



Why build Native Apps?

Subhead Here

- **Work online & offline (access device file system)**
- **Access hardware and sensors**
  - **GPS, camera, Bluetooth, NFC, etc**
- **Great performance**
- **Publish apps to stores**

The challenge

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

Swift
Objective C
MacBook
iOS
iPhone
Windows Desktop
XCode
Mac OS X
iPad
Java
Cocoa
Windows
Android
Galaxy S 5
.NET
Visual Studio
Nexus 10
Surface
Android Studio
C#
Eclipse
Linux
Windows Universal
Embedded
Gradle

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

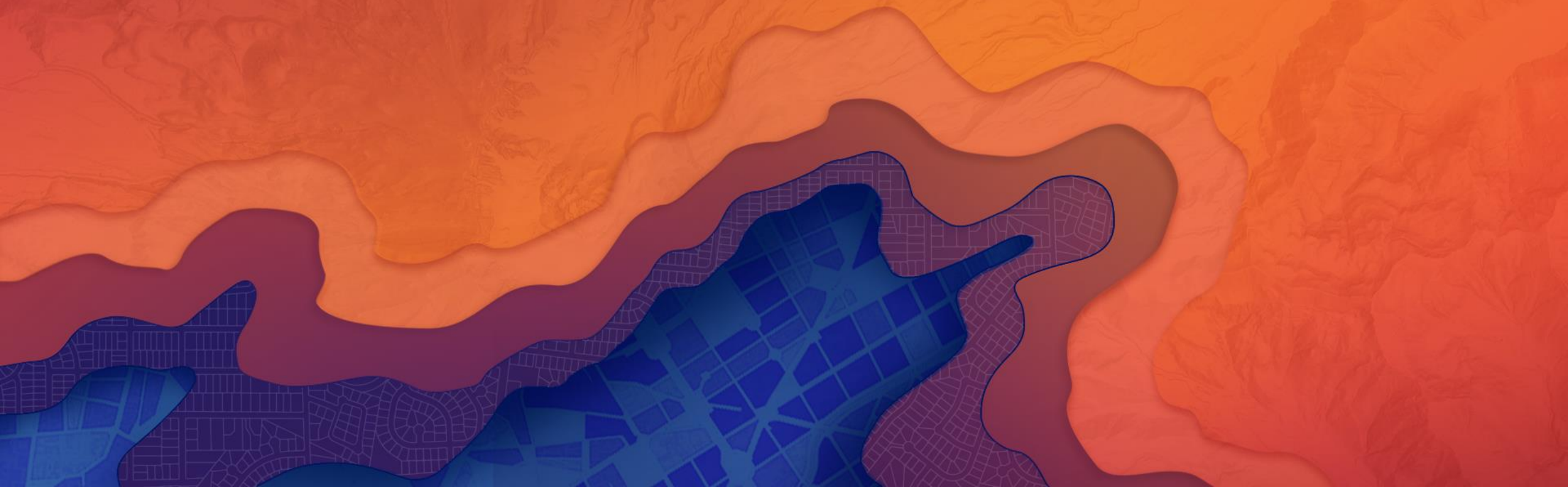
Dynamic property binding

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
  - qmllint syntax tool
- Major Differences from the browser?
  - No DOM or window object
  - No Dojo, jQuery, or other frameworks

JavaScript Expressions

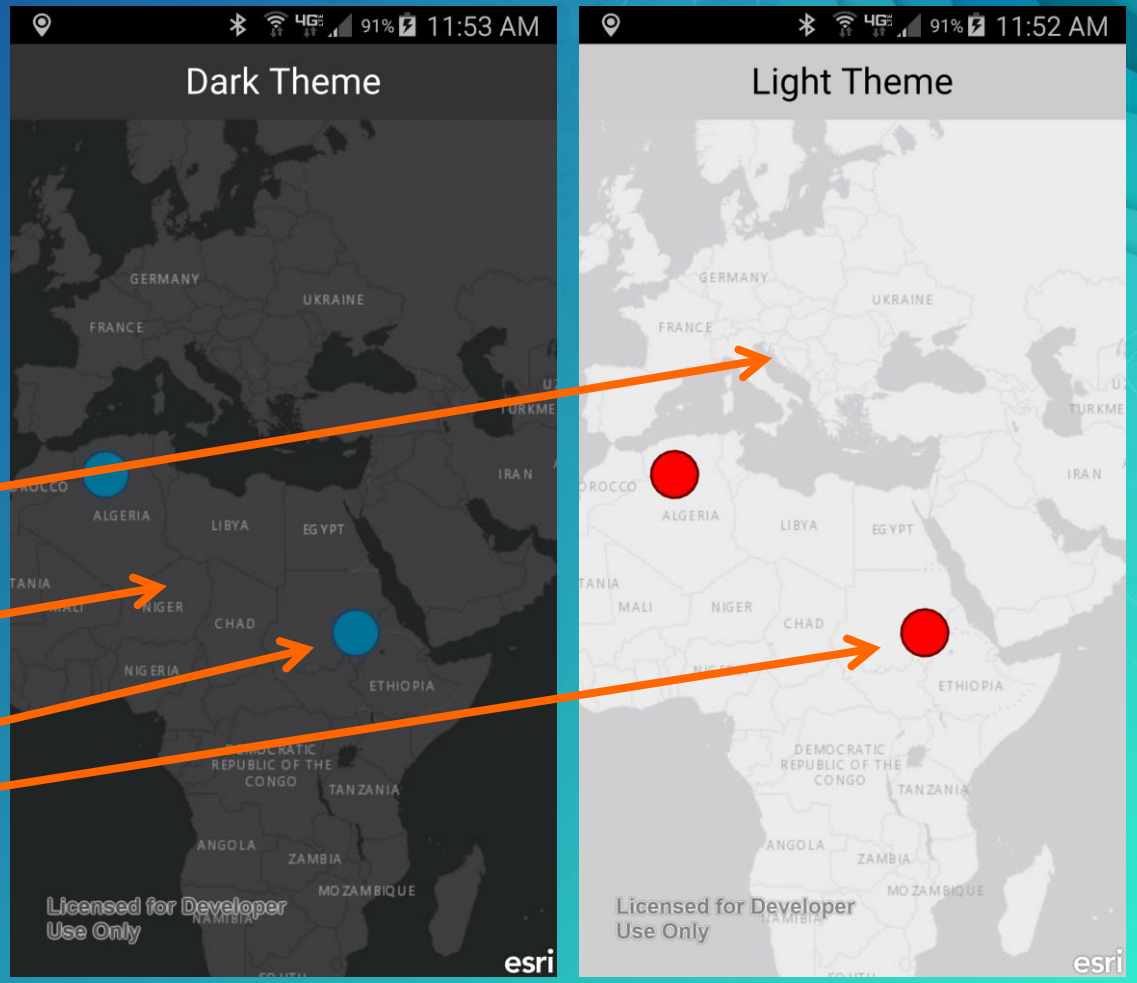
SUBHEAD INFORMATION



JavaScript expressions

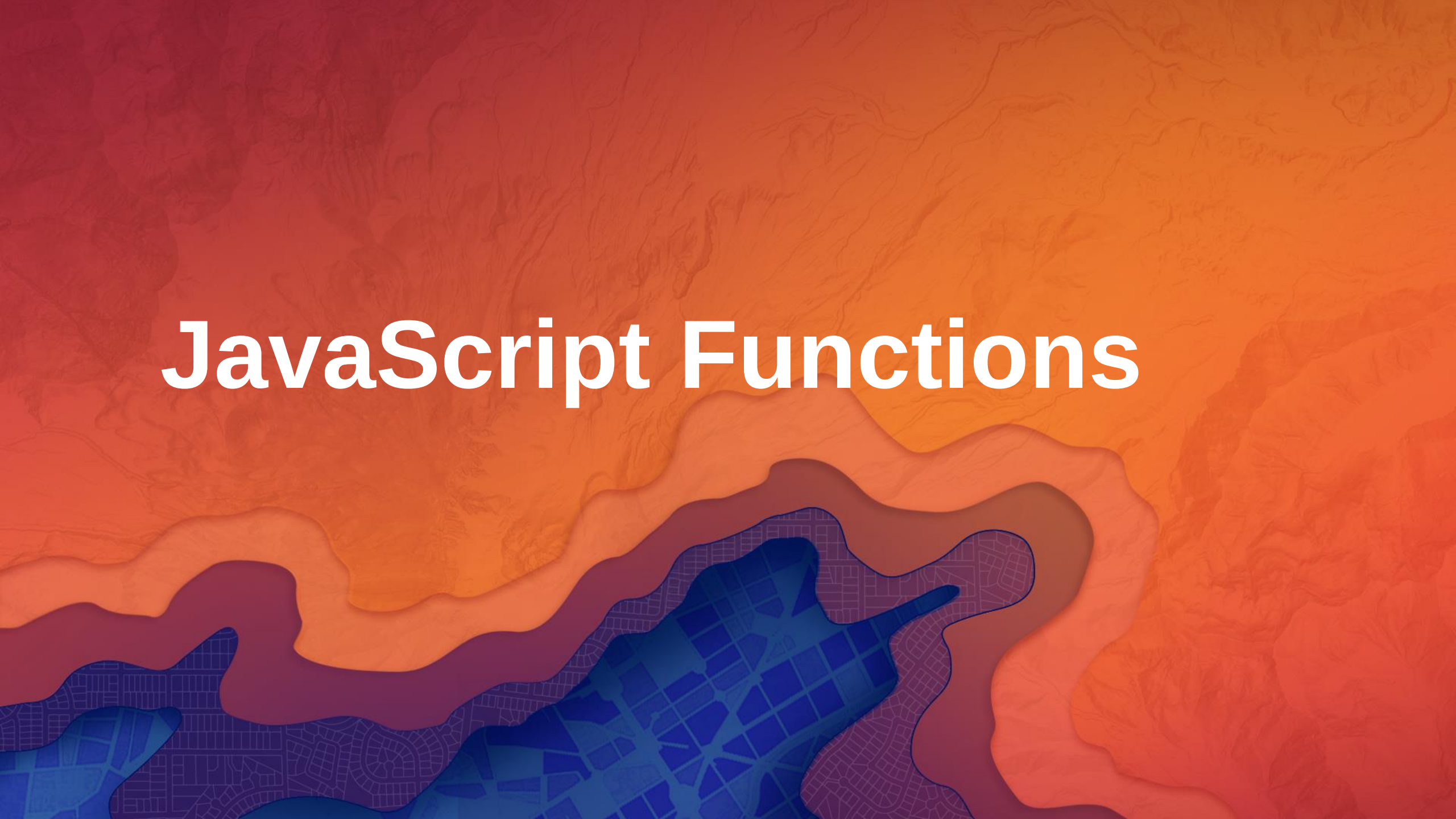
- Bind properties to JS expressions
- Update UI automatically
- Any JS expression (no matter how complex) may be used in a property binding definition
- ex: Change UI depending on ambient light

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



Updating the UI Automatically with Ambient Light Sensor

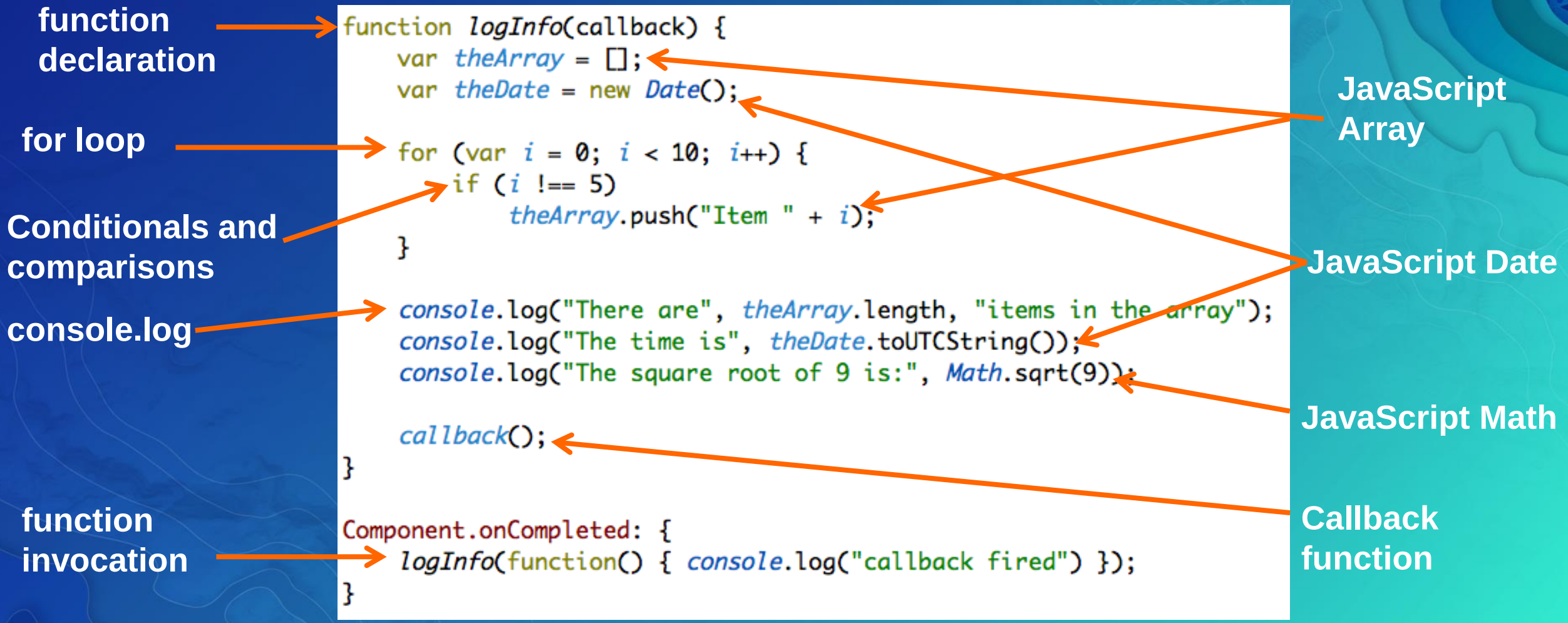
JavaScript Functions



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 ECMA-262 spec
  - Object, Array, Math, and Date
  - Syntax for if statements, switch statements, ternary operators, for loops, while loops, and more are no different than using JavaScript in the browser
  - Some features from ECMAScript 6 being introduced
  - http://doc.qt.io/qt-5/qtqml-javascript-functionlist.html

JavaScript functions

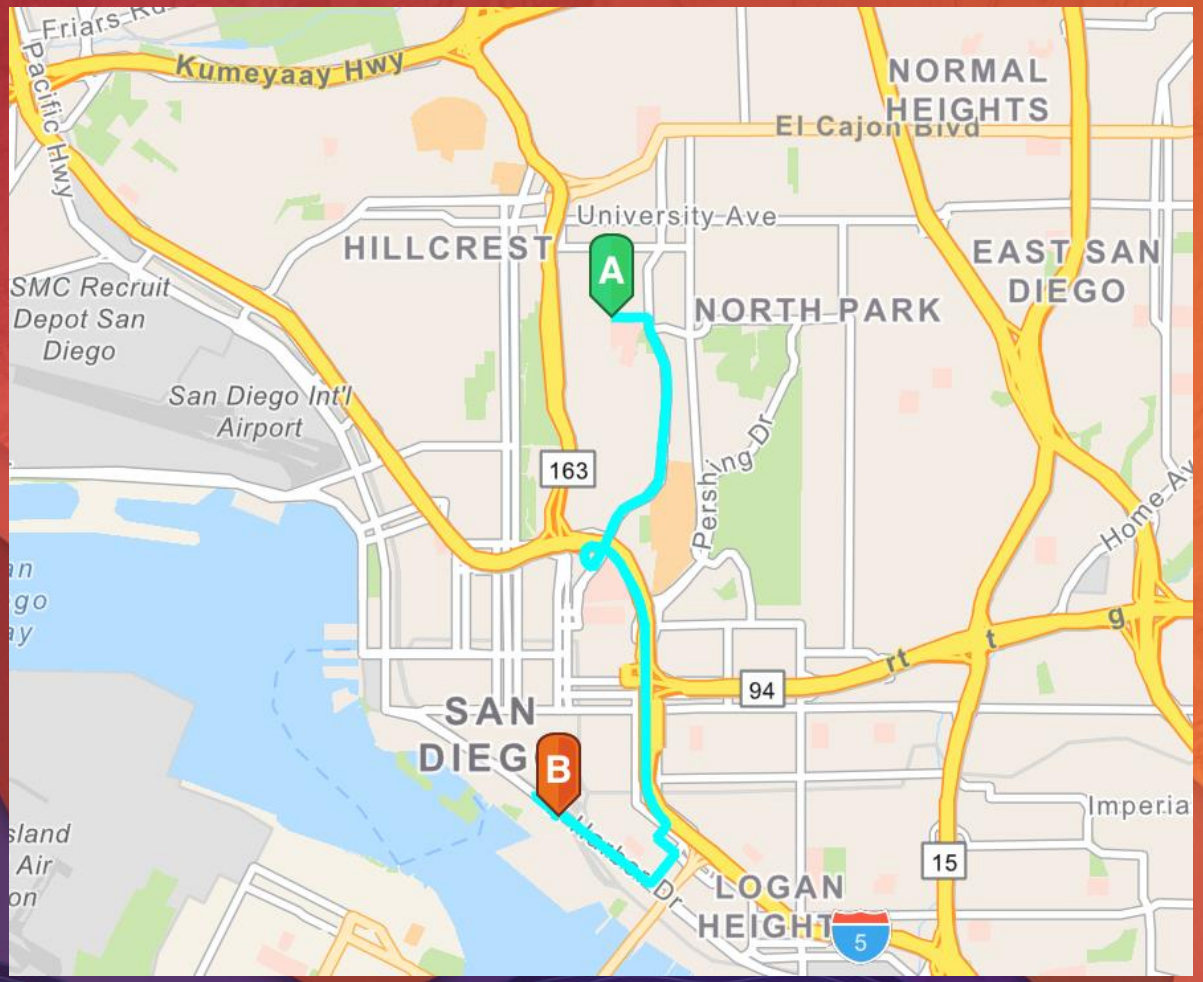


Asynchronous Programming with QML

Async programming with QML

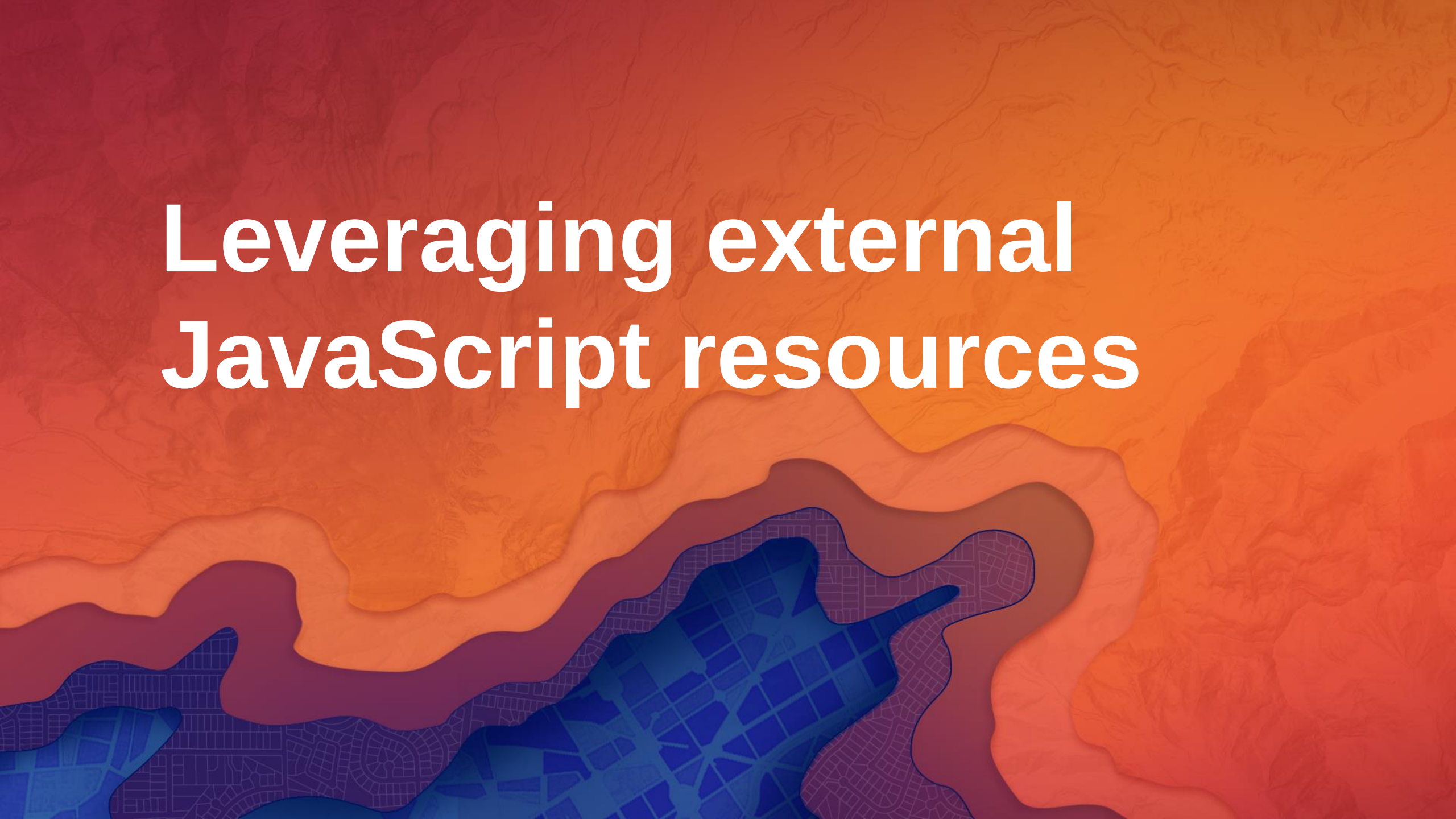
- **Signal and Handler Event System**
- **The event is a *signal* (ex: mouseClicked)**
- **The signal is responded to through a *signal handler***
  - **Signal handler is the name of the signal with the “on” prefix (ex: onMouseClicked)**
  - **Write JavaScript to perform some procedure when a signal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

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



Using async JS functions

Leveraging external JavaScript resources



Importing standalone JavaScript

- Separate non-trivial logic into external JavaScript files
- Make your code reusable
- How?
- Place your functions inside a .js file
- Import to QML files with import statement:

```
import "../Resources/DataDownload.js" as JS_Download
```

- Execute your external functions from QML

```
JS_Download.downloadData(dataNeeded);
```

- Include external JavaScript files from other JavaScript files with Qt.include()

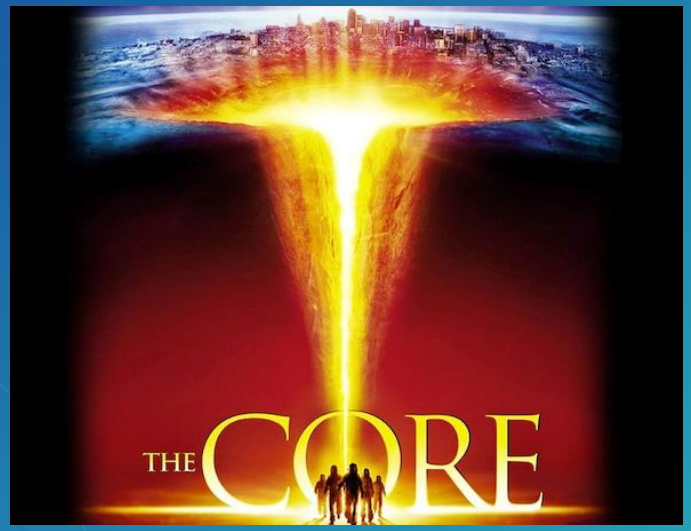
How is Esri using QML?

Esri & QML

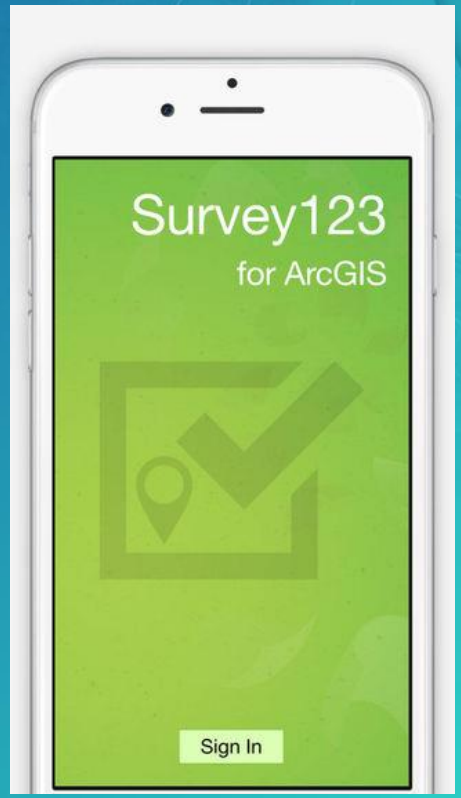
ArcGIS Runtime SDK for Qt

AppStudio for ArcGIS

Runtime Core (for testing)



Survey 123



Recap

- Native apps. Why?
- Overview of Qt and QML
- How to use JS skills to build native apps
  - Property binding with JavaScript expressions
  - JavaScript functions
  - External JavaScript resources

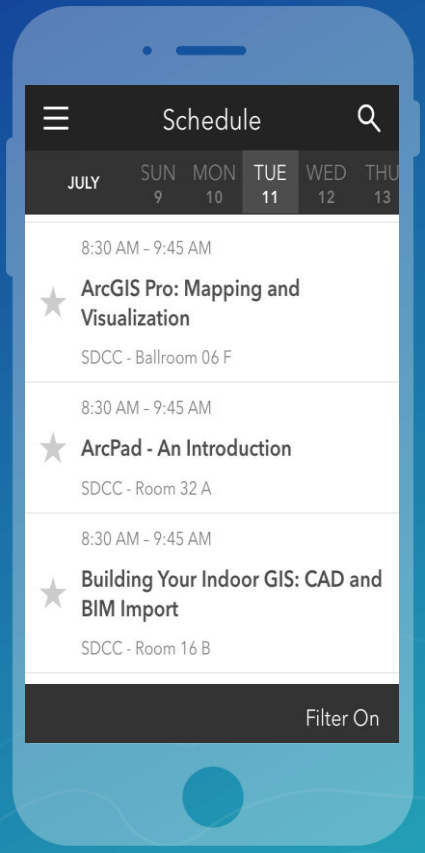
How to get started?

- Visit the Developer's page – http://developers.arcgis.com/qt
- Download the sample viewer
- Install Qt and ArcGIS Runtime, and start building apps!
- Talk to us on GeoNet, Slack, Twitter, etc
- Come visit us at the showcase!

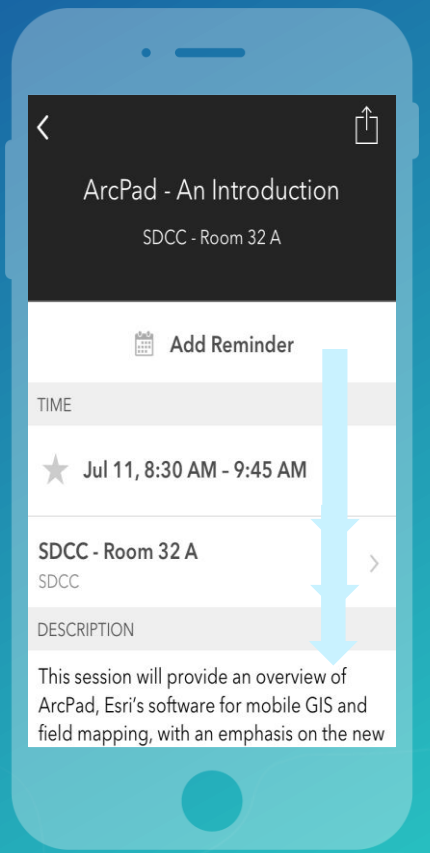
Please Take Our Survey on the **Esri Events App!**

Download the Esri Events app and find your event

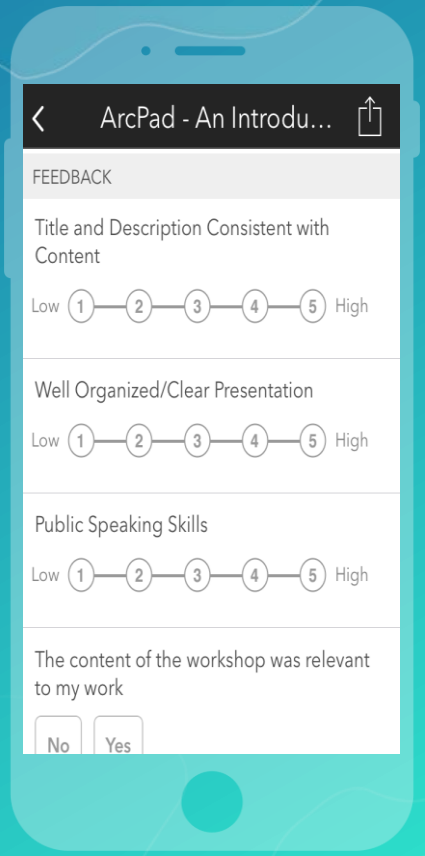
Select the session you attended



Scroll down to find the survey



Complete Answers and Select "Submit"




esri
THE
SCIENCE
OF
WHERE