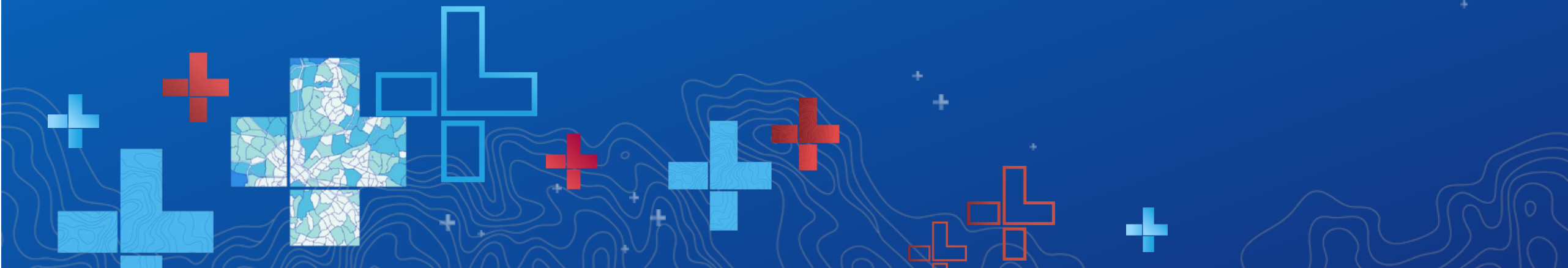
An Introduction to ArcGIS Arcade

Matt Berra, David Attaway, Jason Smith (EPA)

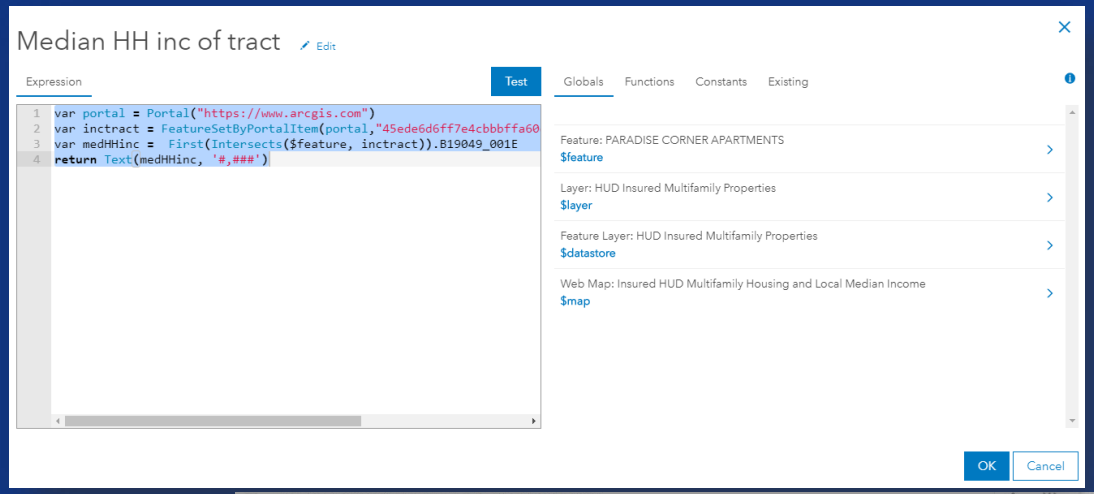
2020 ESRI FEDERAL GIS CONFERENCE | WASHINGTON, D.C.

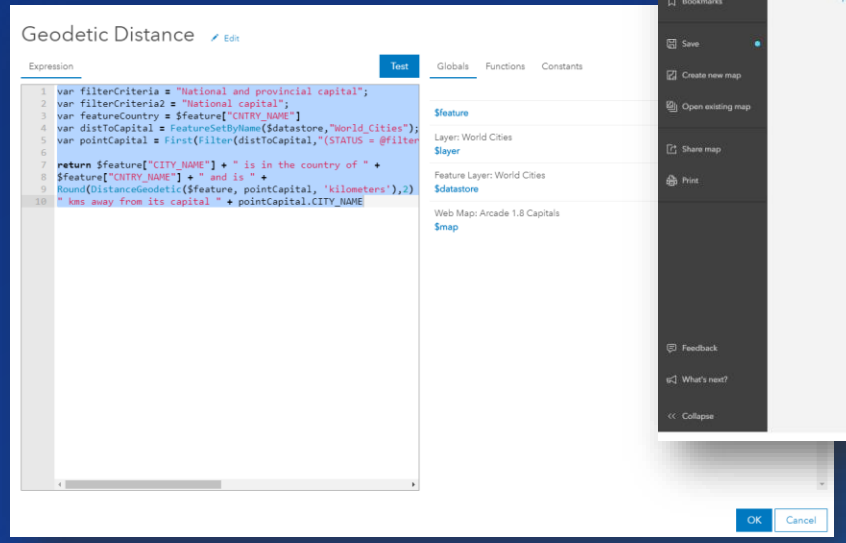


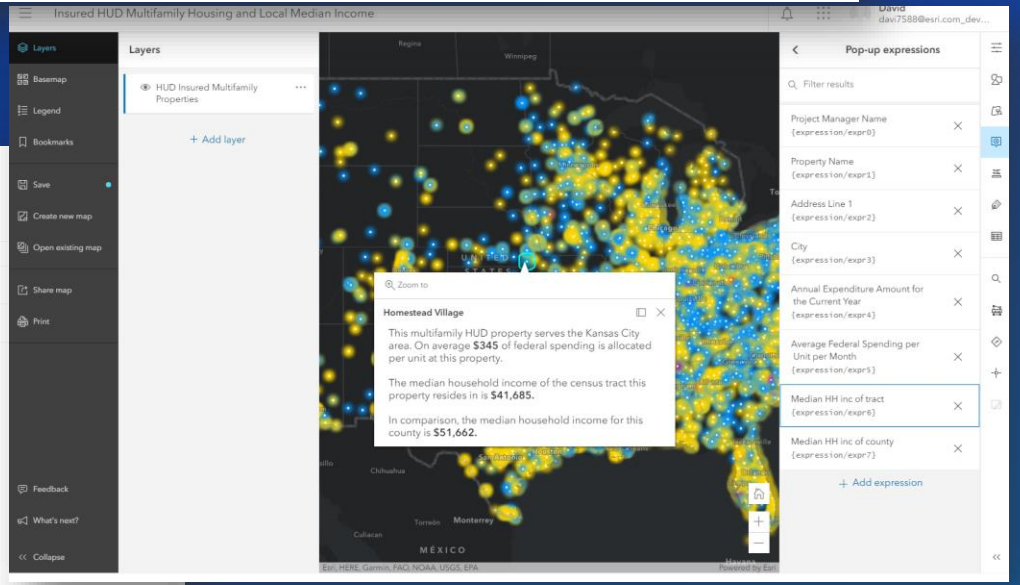
Topics

What we're doing here…

- Overview of Arcade
  - What is Arcade
  - Why use Arcade
- Arcade Languages
  - Variables, Functions, Loops, Conditional Statements
- Arcade
  - Using Arcade

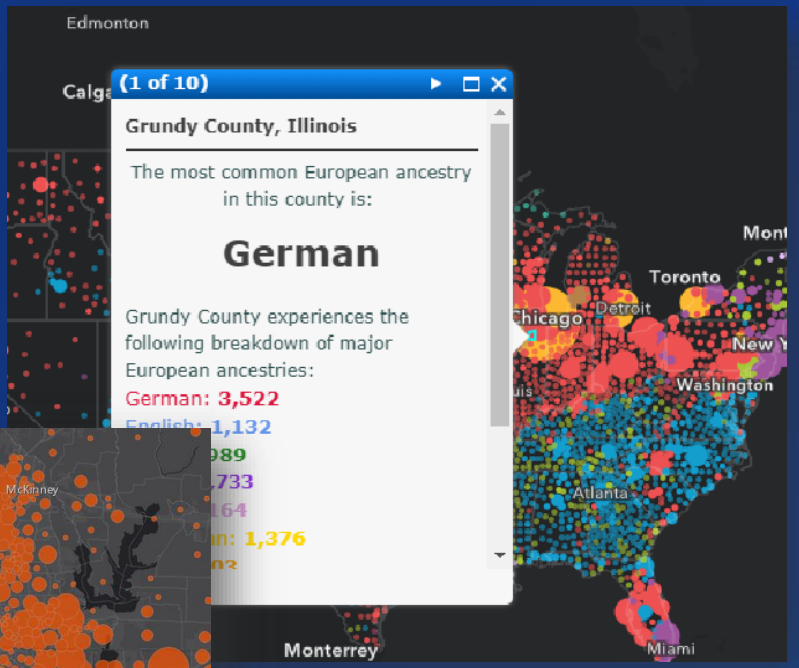


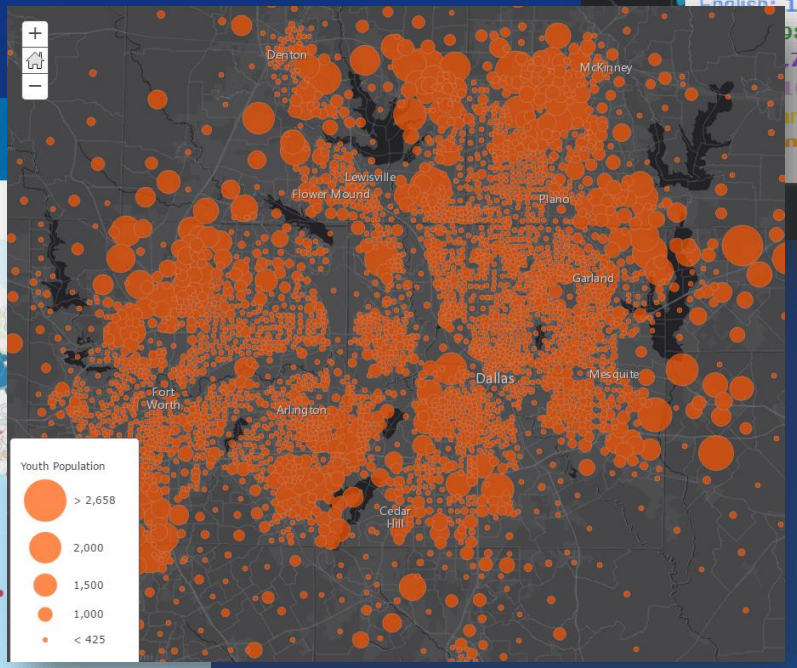


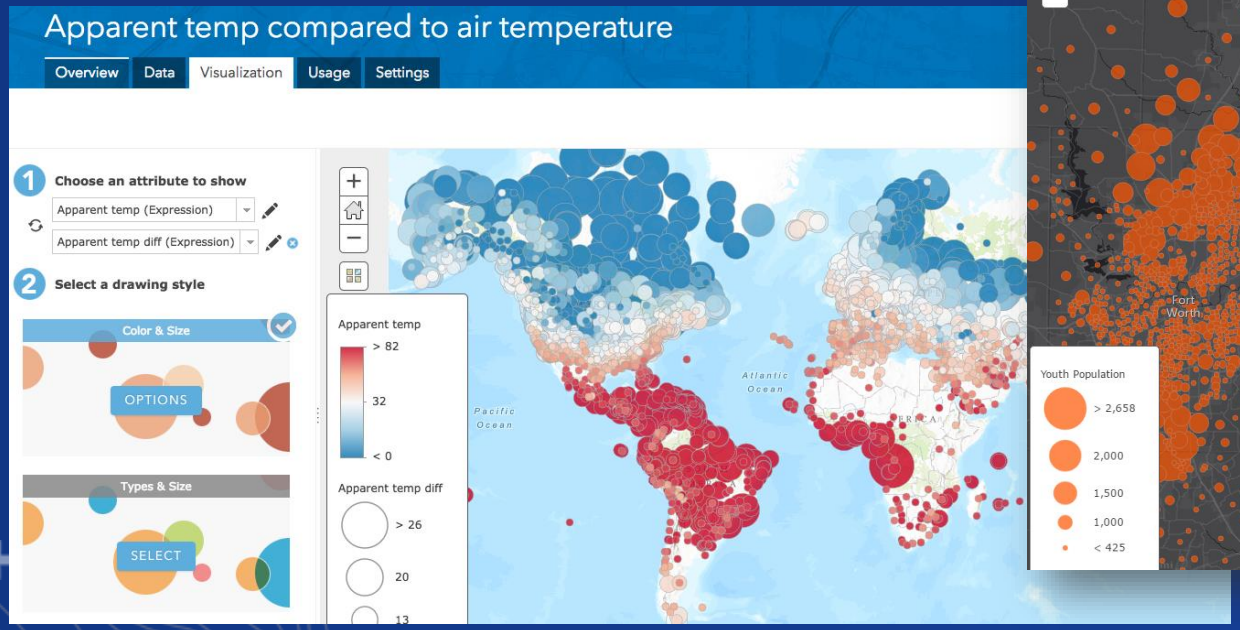


Arcade

Arcade is an expression language that can be used across the ArcGIS Platform. Whether writing simple scripts to control how features are rendered, or expressions to control label text, Arcade provides a simple scripting syntax to deliver these capabilities.







Arcade

- Arcade is NOT intended to be a Full Programming / Scripting Language
  - Lightweight and Simple
  - Similar to spreadsheet calculations
- Embedded Expressions
  - Labeling, Rendering, Symbol Variations, …
  - Sharing

Not a replacement for geoprocessing and automation

Arcade Goals

- Portable
  - Write an expression in ArcGIS Pro and have it work across the platform
- Secure
  - Expression has no impact on security
- Lightweight
  - The language should be small and fast
- Geospatial
  - GIS data is the main use case

Arcade Language

What's available

Functions

Data Functions

- Attachments
- Console
- Count
- **Dictionary**
- **Distinct**
- DomainCode
- DomainName
- Feature
- FeatureSet
- FeatureSetById
- FeatureSetByName
- **FeatureSetByPortalItem**
- **FeatureSetByRelationshipName**
- Filter
- First
- **GroupBy**
- Guid
- HasKey
- IndexOf
- IsNan
- NextSequenceValue
- Number
- OrderBy
- **Portal**
- Reverse
- Sort
- Text
- Top
- TypeOf

Date Functions

- Date
- DateAdd
- DateDiff
- Millisecond
- Second
- Minute
- Hour
- Month
- Weekday
- Year
- Day
- Now
- Today
- Timestamp
- ToLocal
- ToUTC

Geometry Functions

- Angle
- Area
- AreaGeodetic
- Bearing
- Buffer
- BufferGeodetic
- Centroid
- Clip
- Contains
- Crosses
- Cut
- Difference
- Disjoint
- Distance
- **DistanceGeodetic**
- Equals
- Extent
- Geometry
- Intersection
- Intersects
- **IsSelfIntersecting**
- Length
- LengthGeodetic
- MultiPartToSinglePart
- Multipoint
- Overlaps
- Point
- Polyline
- Polygon
- RingIsClockwise
- SetGeometry
- SymmetricDifference
- Touches
- Union
- Within

Logical Functions

- IsEmpty
- DefaultValue
- When
- Decode
- Iif
- Boolean

Text Functions

- Concatenate
- Find
- Lower
- Left
- Mid
- Proper
- Replace
- Right
- Split
- Trim
- Upper
- UrlEncode

Mathematical Functions

- Abs
- Acos
- Asin
- Atan
- Atan2
- Average
- Ceil
- Constrain
- Cos
- Exp
- Floor
- Log
- Mean
- Min
- Max
- Pow
- Random
- Round
- Sin
- Sqrt
- Stdev
- Sum
- Tan
- Variance

Global Variables and Profiles

Alias

Allows the map author to write an expression to evaluate a numeric value and return a string alias representing that value.

Global Variables
- $value

Attribute Rules

Attribute Rule Calculation
Attribute Rule Constraint
Attribute Rule Validation

Global Variables
- $feature
- $datastore

Feature Z

Use expressions to calculate z values for features in a 3D scene.

Global Variables
- $feature

Field Calculate

Use expressions to update a field value based on expression logic.

Global Variables
- $feature
- $layer
- $datastore

Labeling

Use expressions to determine the label to show on the map for each feature.

Global Variables
- $feature

Popup

Use expressions to return values for display in the popup.

Global Variables
- $feature
- $layer
- $map
- $datastore

Visualization

Use expressions to evaluate a value used to drive the visualization.

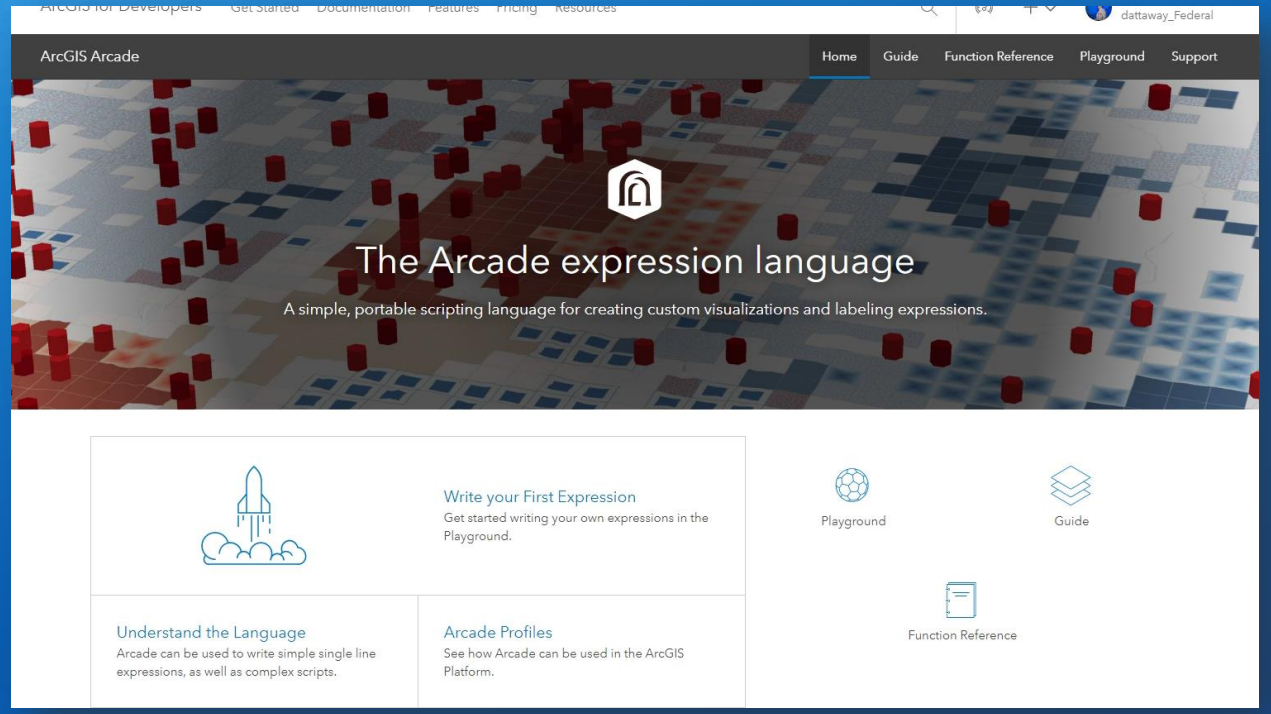
Global Variables
- $feature
- $view.scale

Future Profiles

Arcade 2019 recap

- Arcade continues to grow as a platform wide story in 2019
- Four major releases in 2019
  - New GeoAnalytics profile
  - Facilitate working with UN data
  - Improved functionality (new functions and updates to existing)
    - FeatureSets supported on more existing functions
    - New functions to work with area's and distance
    - Easier access to related data, living atlas content
    - GroupBy, Distinct functions

Arcade	ArcGIS Pro	ArcGIS Enterprise portal	ArcGIS JS API (4.x)	ArcGIS JS API (3.x)	ArcGIS Runtime
1.0	1.4	10.5.1	4.2	3.19	100
1.1	2.0[1]	n/a	4.4	3.21	100.1
1.2	n/a	10.6	4.5	3.22	n/a
1.3	2.1	10.6.1	4.6[2]	3.23	100.2
1.4	2.2	n/a	4.8[2]	3.25	100.3
1.5	2.3	10.7	4.10[3]	3.27	n/a
1.6	n/a	n/a	4.11	3.28	100.5[3]
1.7	2.4	10.7.1	4.12	3.29	100.6[3]
1.8	n/a	n/a	4.13	3.30	n/a
1.9	2.5	10.8	n/a	n/a	n/a

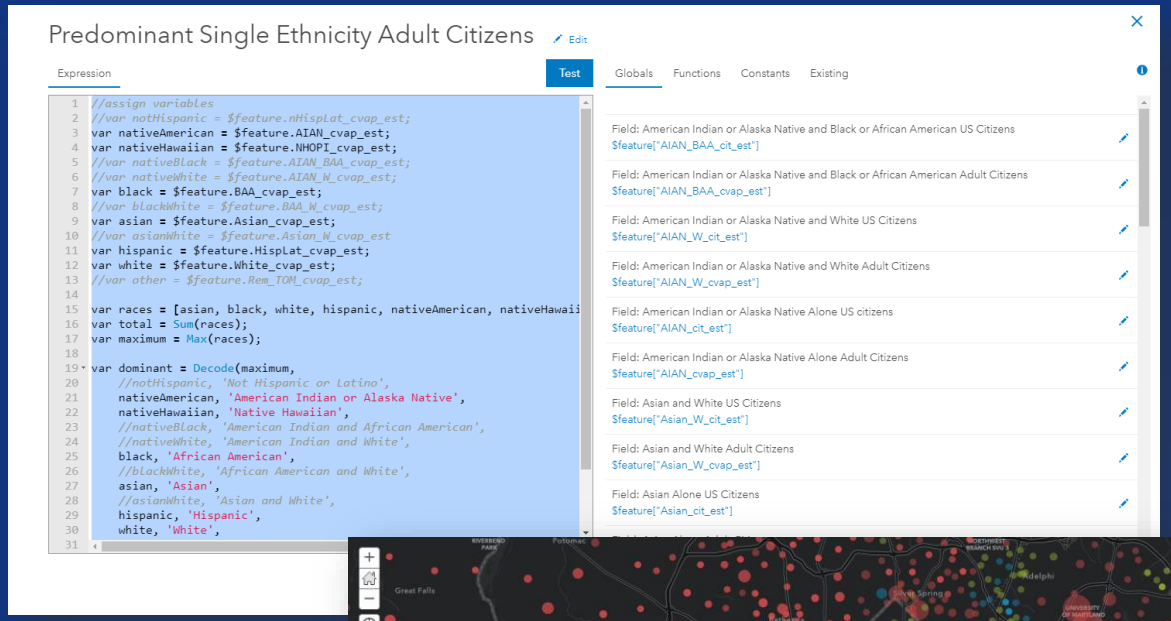


Hello World: Flooding

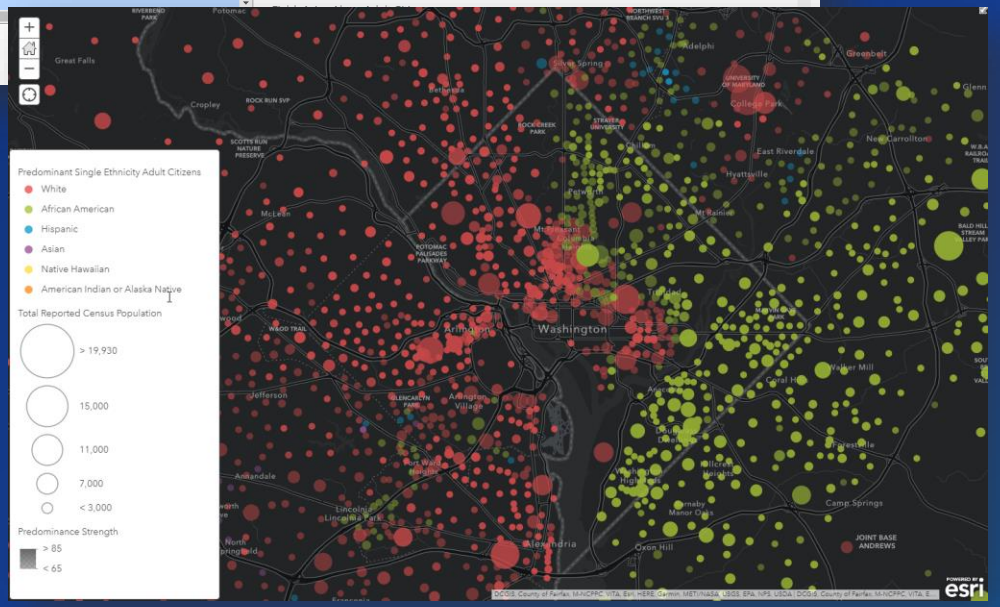
Matt Berra

Language Features

- Designed for Simple Expressions
  - Single Line – Implicit Returns
  - Case insensitive Language
  - Dictionary Syntax for Field Access



- Has language features to support more complex scripts
  - Type system
  - Implicit and Explicit type casting
  - Flow control statements: if, for
  - Variable declaration and assignments
  - User defined functions



- Arcade scripts run and return a value. They are self contained and cannot alter their environment.

Case Insensitive

- Language is case insensitive
  - Quicker to author
  - Simpler for working with data and field names

```
UpPeR(‘Hello World’)
$feAtuRe.FiElD_nAmE
```

Type System

- Simple Types
  - Numbers
  - Booleans
  - Dates
  - Strings
- Object Types
  - Dictionary
  - Feature
  - Array
  - Point
  - Line
  - Polygon
  - Multipoint

```
var myNumber = 10;
var myText = "Hello";
var myDate = Date(2015,1,1);
var myBool = true;
var myDictionary = {"key1": 10};
var myFeatures = Feature({"geometry":{...},
                          "attributes":{"key1": 10}});
var myArray = [1, 2, 3];
var myPoint = Point({...});
```

Dates, Dictionary, Feature, Point, Line….all have overloaded constructors.

Implicit and Explicit Type Casting

- Implicit Casting
  - For Function Parameters
  - For Expressions
- Explicit Casting
  - Functions cast between types
  - Number, Text, Date, Boolean

```
Implicit Casting:
return 10 + "Star"

Explicit Casting:
return text(10) + "Star"
```

if Statement

- if statements are supported
- Simple single line expressions
  - IIF
  - Decode
  - When

```
Traditional if Statement:
if (mapPoint){
    return ‘Star’;
} else {
    return ‘Circle’;
}

Equivalent:
IIF(mapPoint, ‘Star’, ‘Circle’);
```

for Statement

- for Statements
  - Same syntax as JavaScript
- for in Statements
  - Iterates over indices of an array, or field names of a Dictionary or Feature
- Break, Continue, Return Statements supported inside block

```
for(var i=1; i<100; i++) {
  if (i==3) continue;

  if (i==5) break;
    n+=i;
}

for(var k in myArray) {
    n+=myArray[k];
}
```

User Defined Functions

- Function keyword used to declare functions
  - Must be declared ahead of use
  - Variables defined inside function have local scope
  - Functions cannot be declared inside of functions

```
function MultiplyNumber(number) {
    return number * 50;
}

MultiplyNumber(10);
```

Variable Assignments

- Variables can have their values reassigned
  - Objects types may be immutable if passed into Arcade
  - Arrays are not Sparse. They must be sequential
  - Geometry types are immutable inside. You cannot change the points in a path of a line.

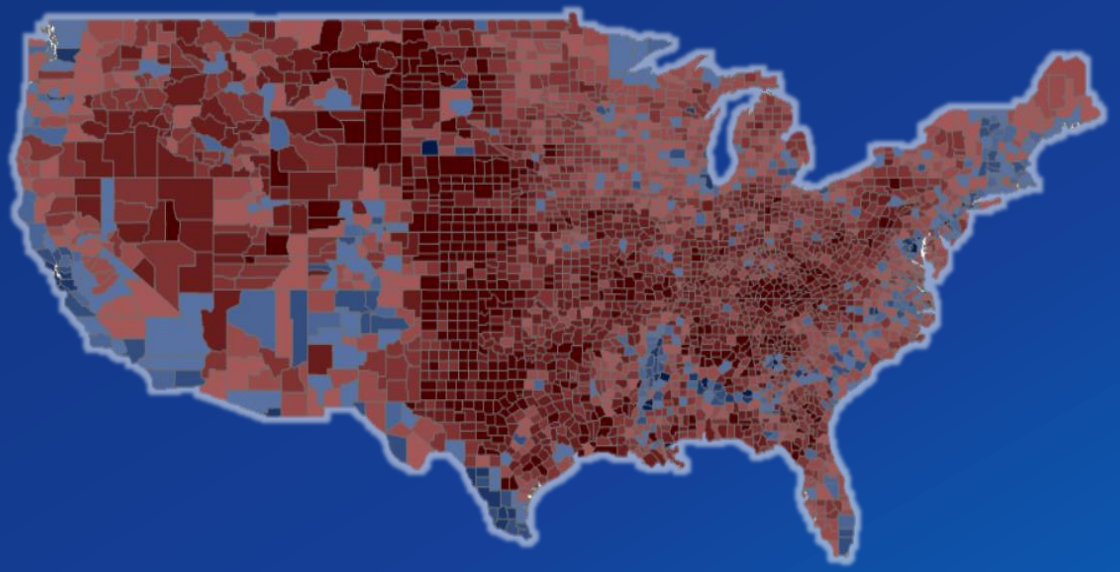
```
++myNumber
i = i + 1;
myArray[0] = 11;
myFeature.Field1 = 'arcgis';
```

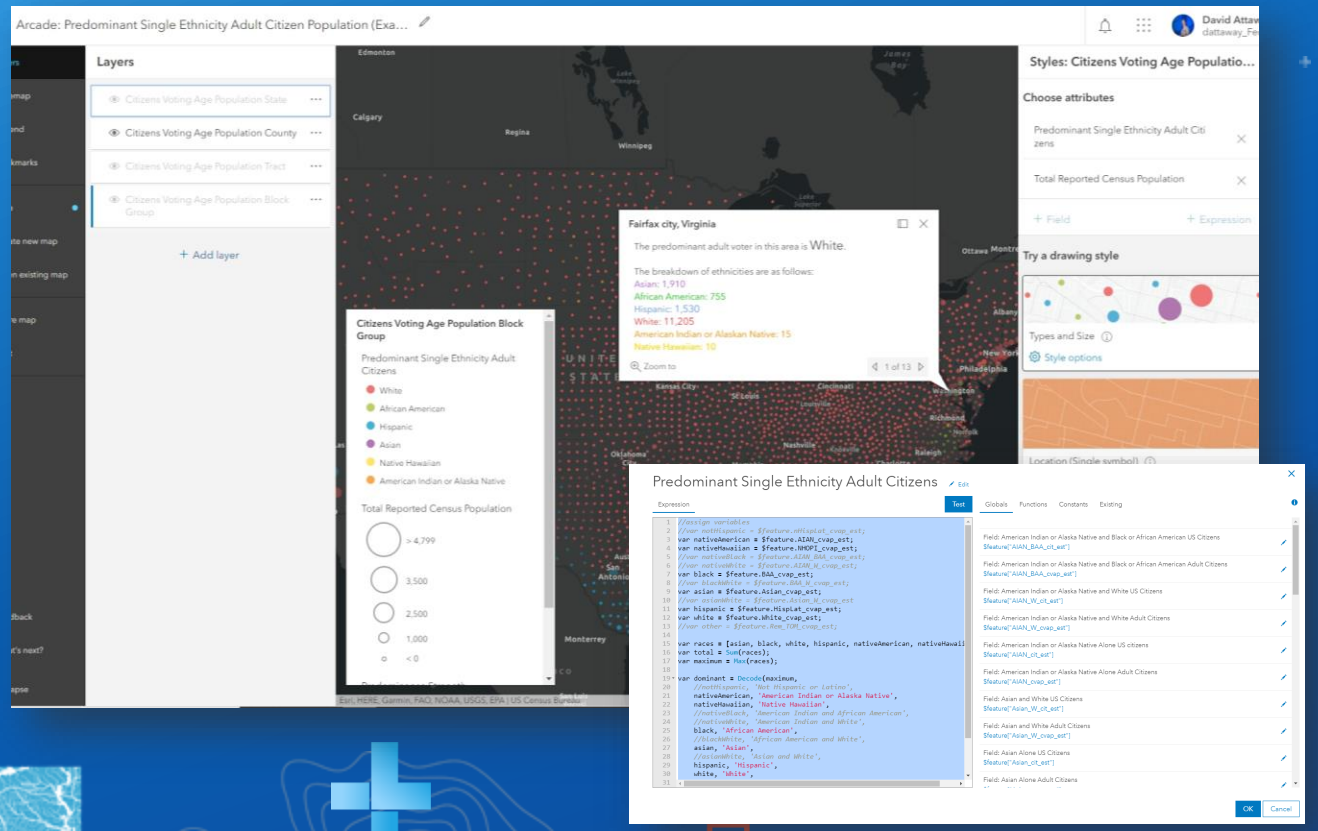
Rendering

Attribute Symbolization

The visualization profile allows the map author to write an expression that evaluates to a value used to drive the visualization.

```
        var dem = $feature.votes_dem;
        var rep = $feature.votes_gop;
        var total = $feature.votes_total;
    Color:
        if (dem > rep) {
            return "Democrat“;
        } else {
            return "Republican“;
        }
    Intensity:
        if (dem > rep) {
            return dem/total;
        } else {
            return rep/total;
        }
```



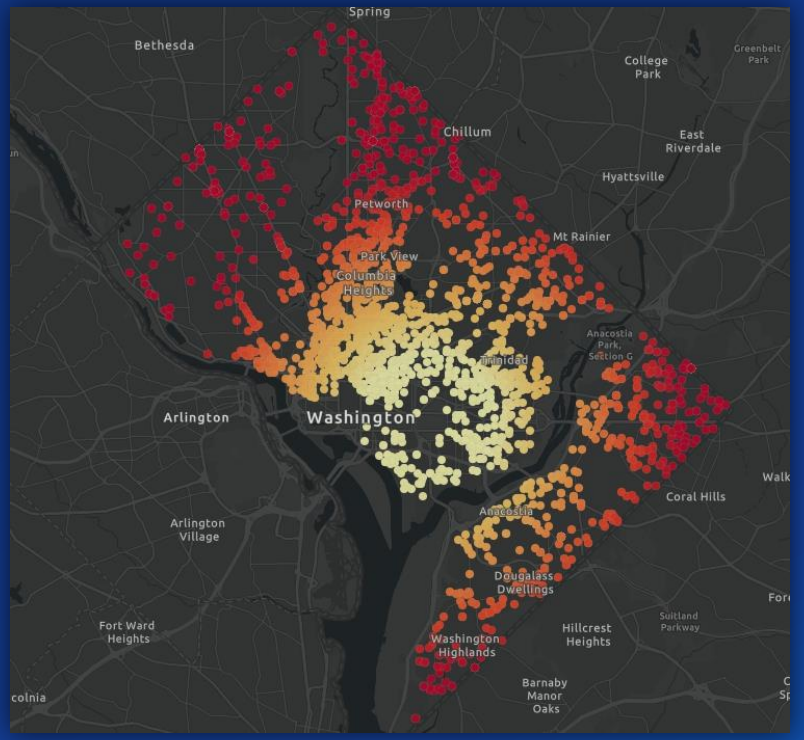


Predominant Ethnicity: Arcade

David Attaway

Geometry Functions

```
var json = {"x":-8572632.50,"y":4705891.49 ,"spatialReference":{"wkid":3857}};
var source_point = Point(json);
var feature_point = Geometry($feature);
var dist = Distance(feature_point, source_point, 'meters');
return dist;
```



Geometry Functions

Warning

Since Arcade expressions execute for each feature, using multiple geometry operations within the context of the labeling and visualization profiles can be expensive and severely impact the performance of the application. Also note that geometries fetched from feature services, especially polylines and polygons, are generalized according to the view's scale resolution. Be aware that using a feature's geometry (i.e. `Geometry($feature)`) as input to this function will yield results only as precise as the view scale. Therefore, results returned from geometry operations in the visualization and labeling profiles may be different at each scale level. Use these functions at your discretion within these contexts.

FeatureSets

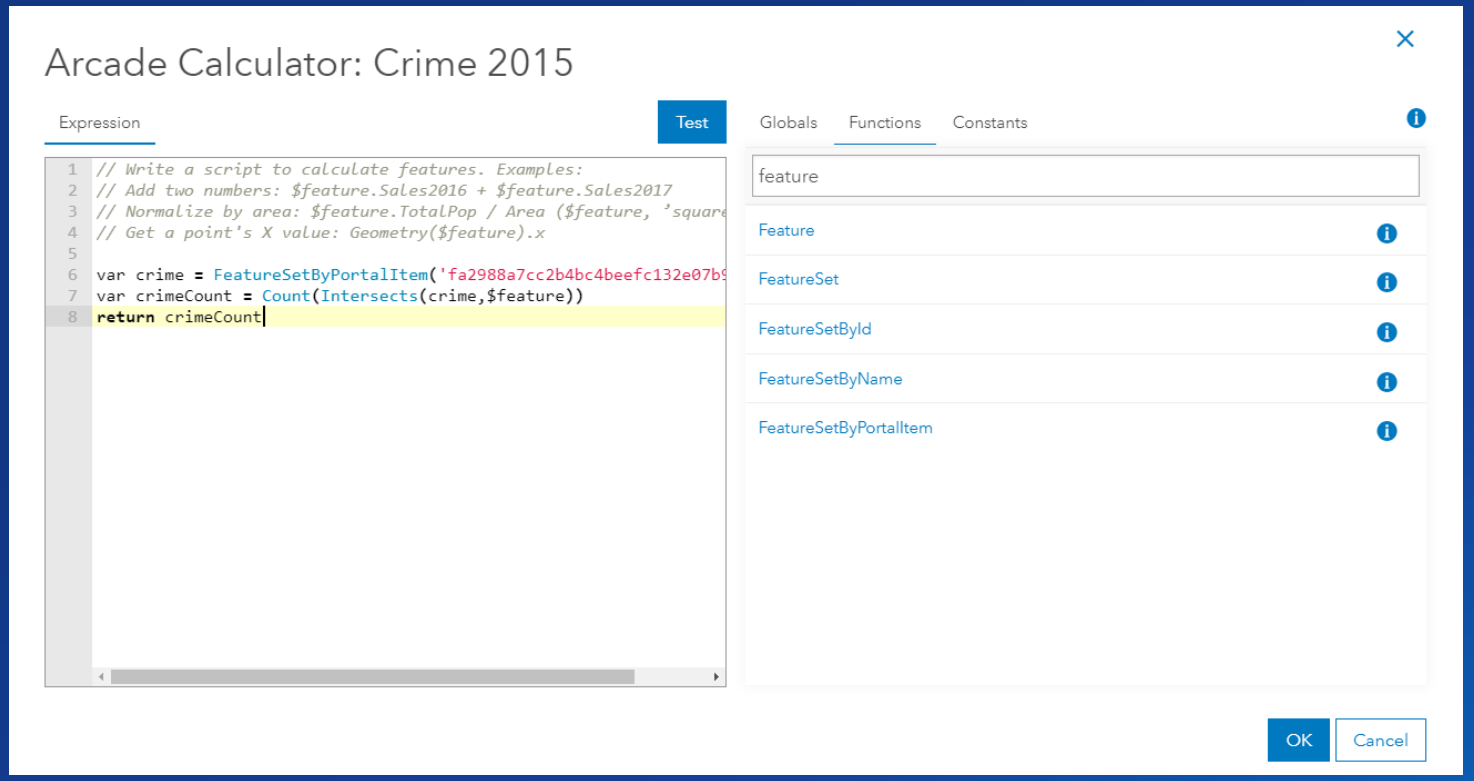
- At the heart of much of the 2019 work
- Allows you to work with multiple features instead of a single feature.
- Opens the door to many workflows
  - Summarize information about multiple features
  - Find a specific feature within a set of features
  - Combine information into a single popup
  - Supported with spatial and attribute functions

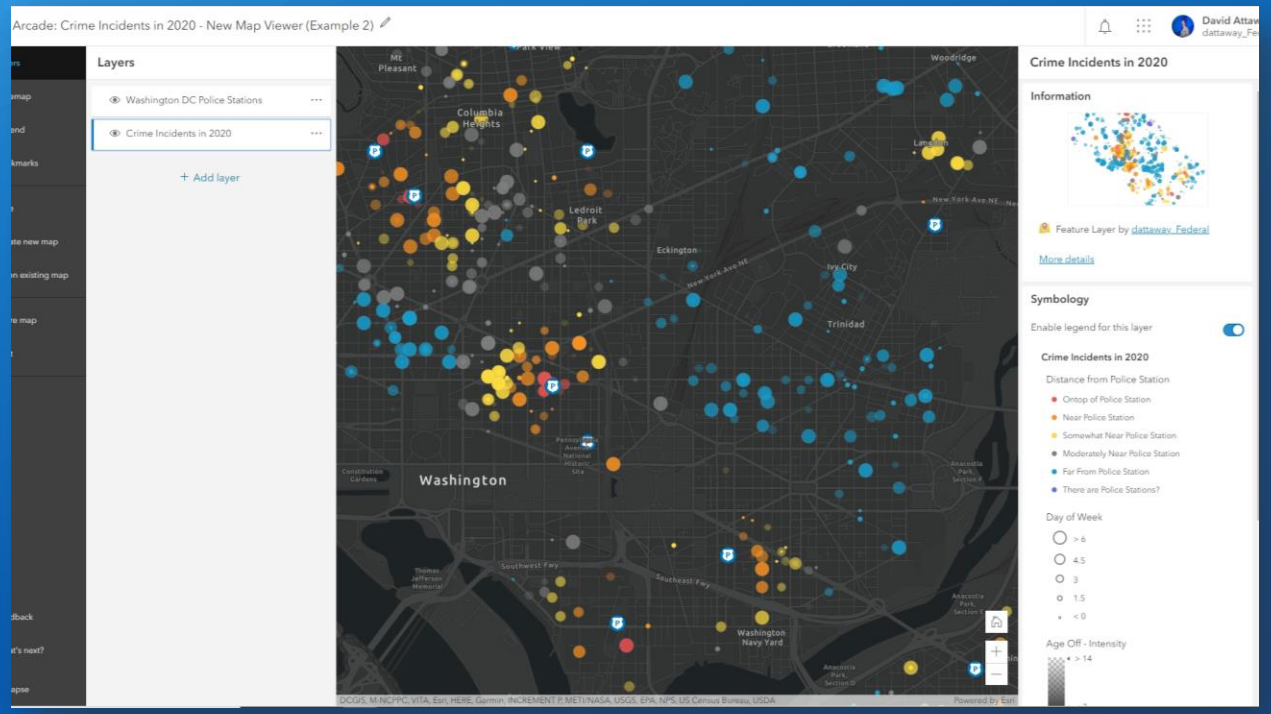
```
var intersectArea = Intersects(FeatureSetByName($map,"Urban_Forestry"), $feature);
var uniqueTreeList = Distinct(intersectArea, "COMMON_NAM");

var treeList = '';
for (var tree in uniqueTreeList) {

    treeList += tree.COMMON_NAM  + TextFormatting.NewLine;

}
return treeList;
```

Feature Sets

Field Calculations

```
var crime = FeatureSetByPortalItem('fa2988a7cc2b4bc4beefc132e07b9d80', 0);
var crimeCount = Count(Intersects(crime, $feature));
return crimeCount;
```



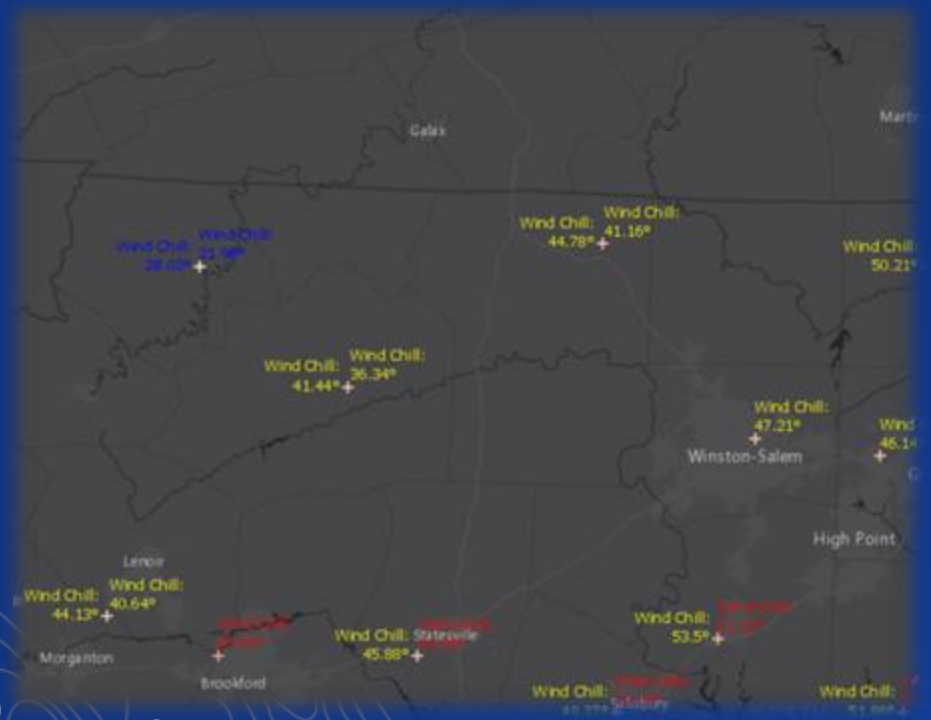


Arcade: Crime Data

David Attaway

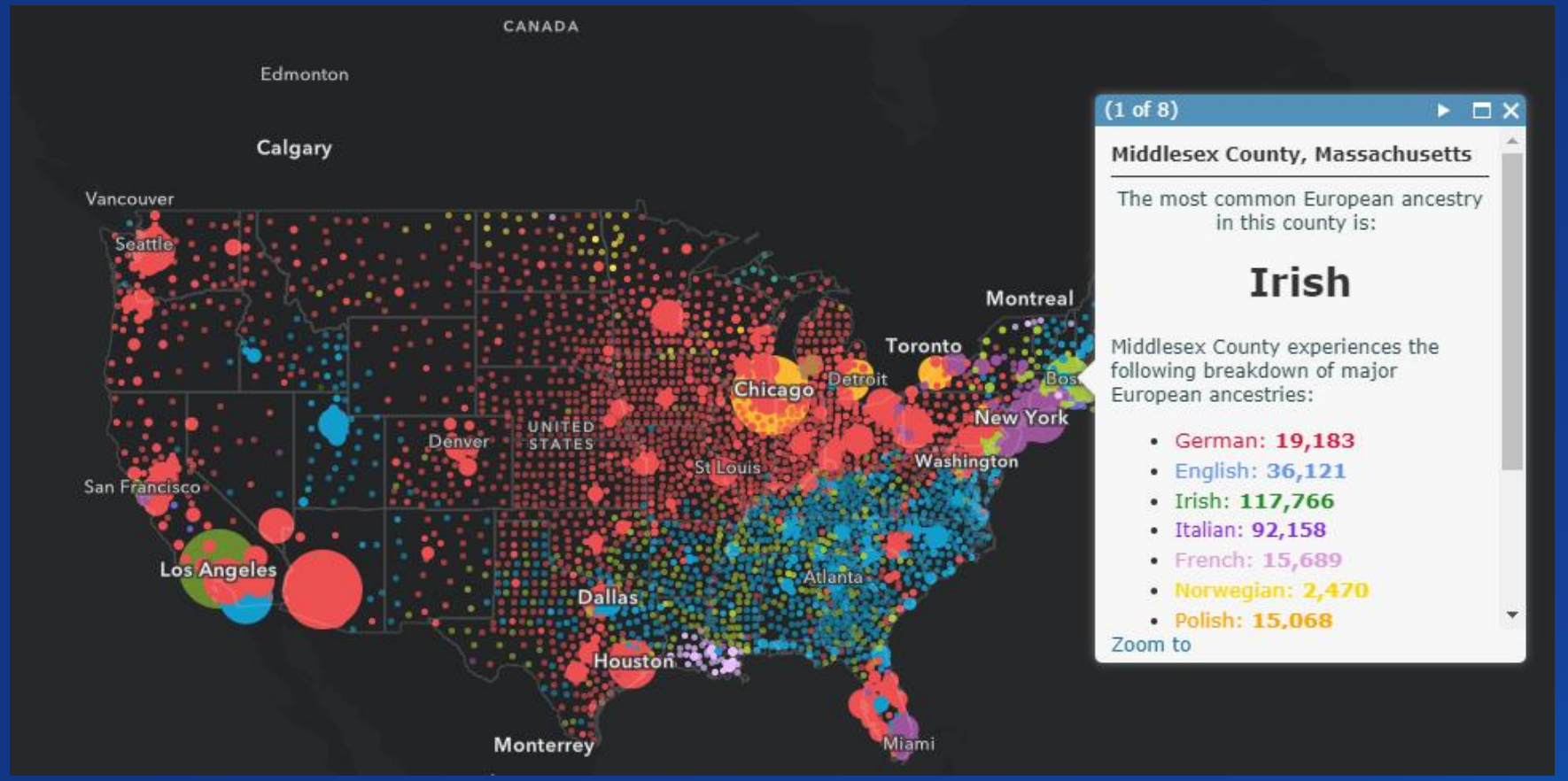
Labeling

When the label is to be drawn, the script will be evaluated. It will be passed the feature that is about to be labeled. It is expected that the script returns a text string, comprising the label to be drawn.

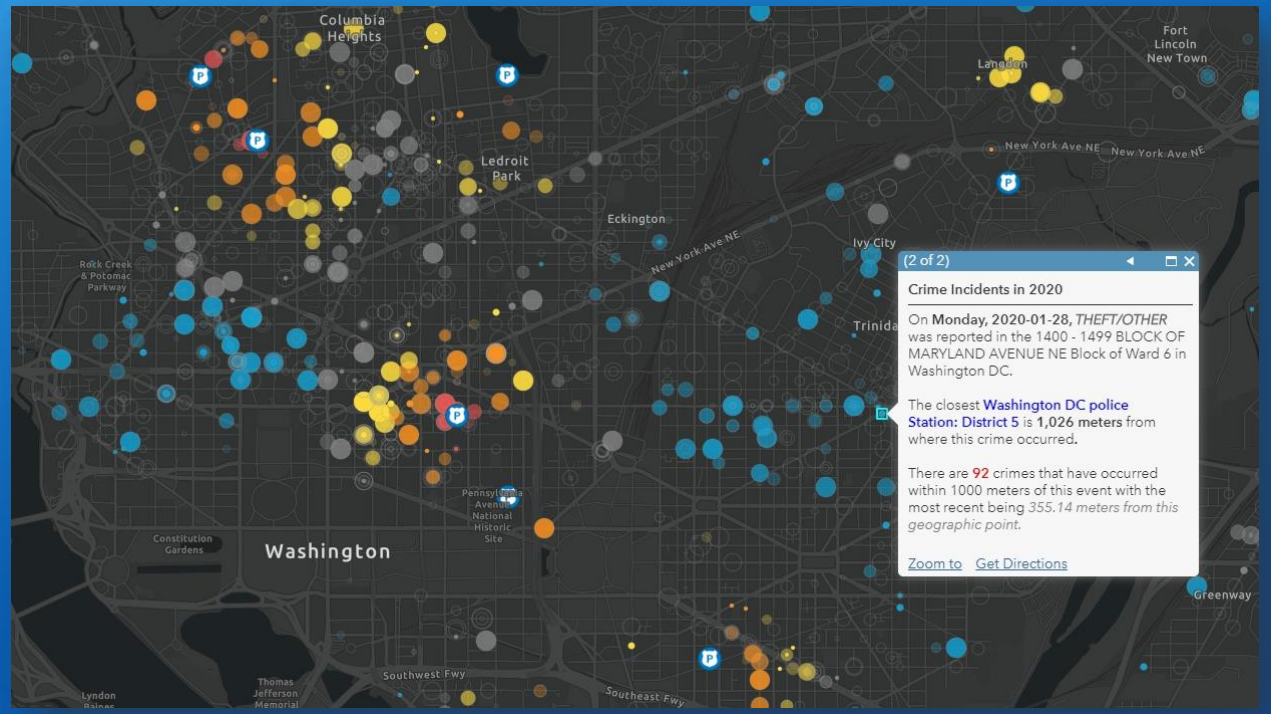



```
var windSpeed = $feature.WIND_SPEED;
var temp = $feature.TEMP;
var windChill =  Round(35.74 + (0.6215*temp) -
35.75*POW(windSpeed,0.16) + 0.4275*temp*POW(windSpeed,0.16), 2);


if (windChill <= 32) {
    return '<CLR blue = "255">Wind Chill: '+ windChill + '° </CLR>';
} else if (windChill >32 && windChill <= 55) {
    return '<CLR yellow = "255">Wind Chill: '+ windChill + '° </CLR>';
} else {
    return '<CLR red = "255">Wind Chill: '+ windChill + '° </CLR>';
}
```

Popups




```
// return the value to the map, to be mapped
IIf(total > 0 && domAttainment == "Irish", domAttainment, null);
```

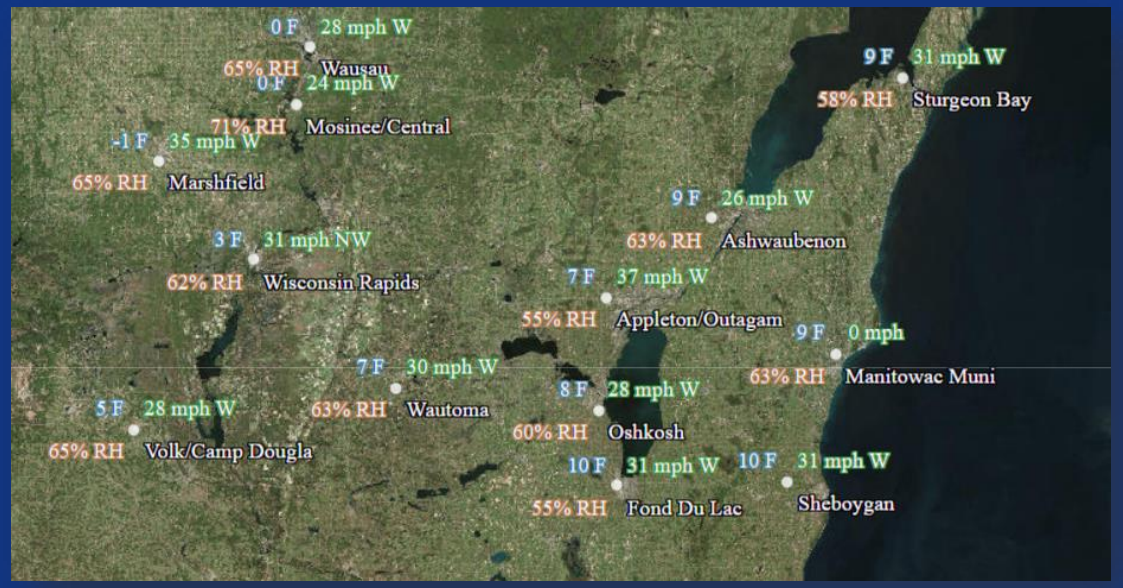


Arcade Pop-ups

David Attaway

JavaScript

Using Arcade

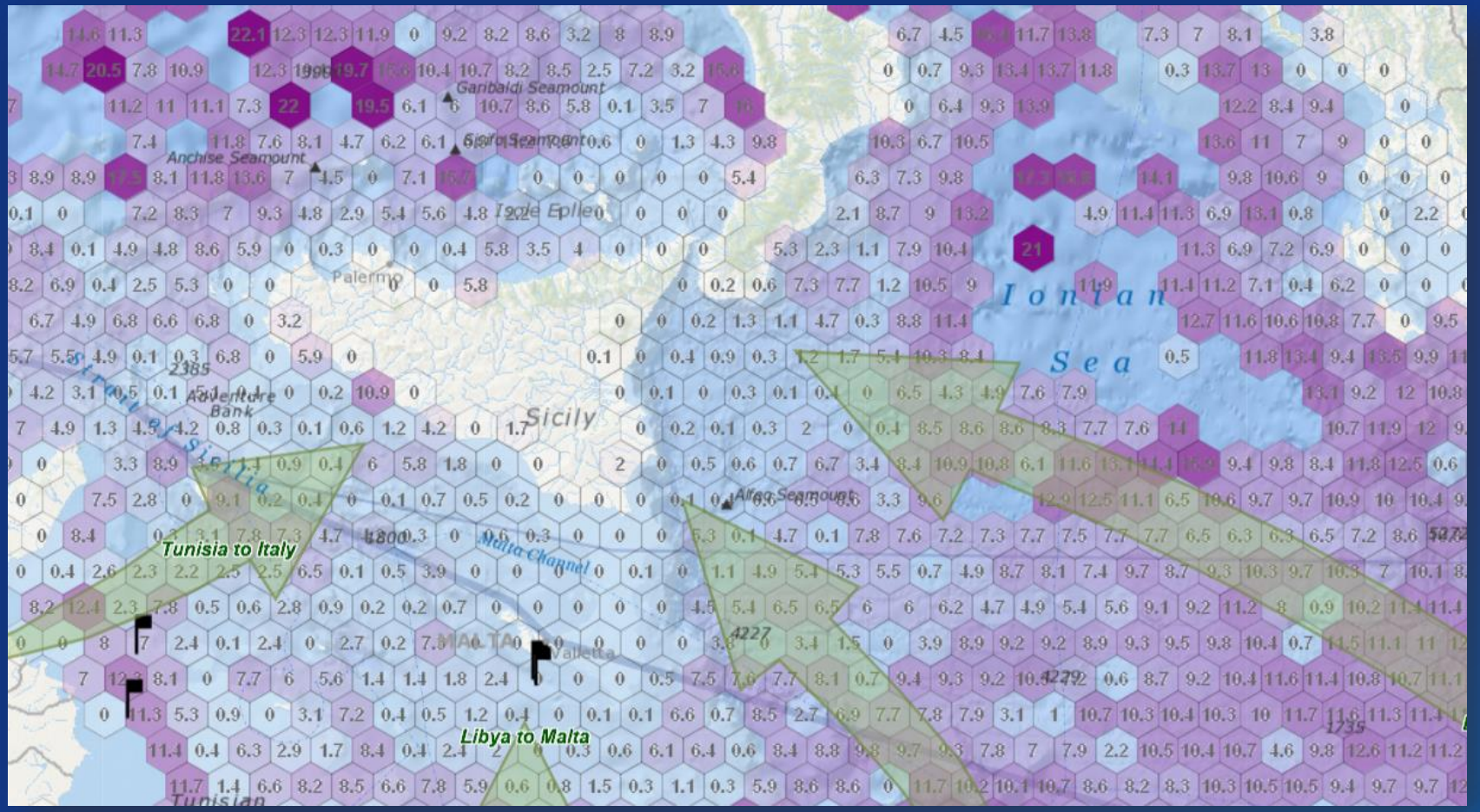


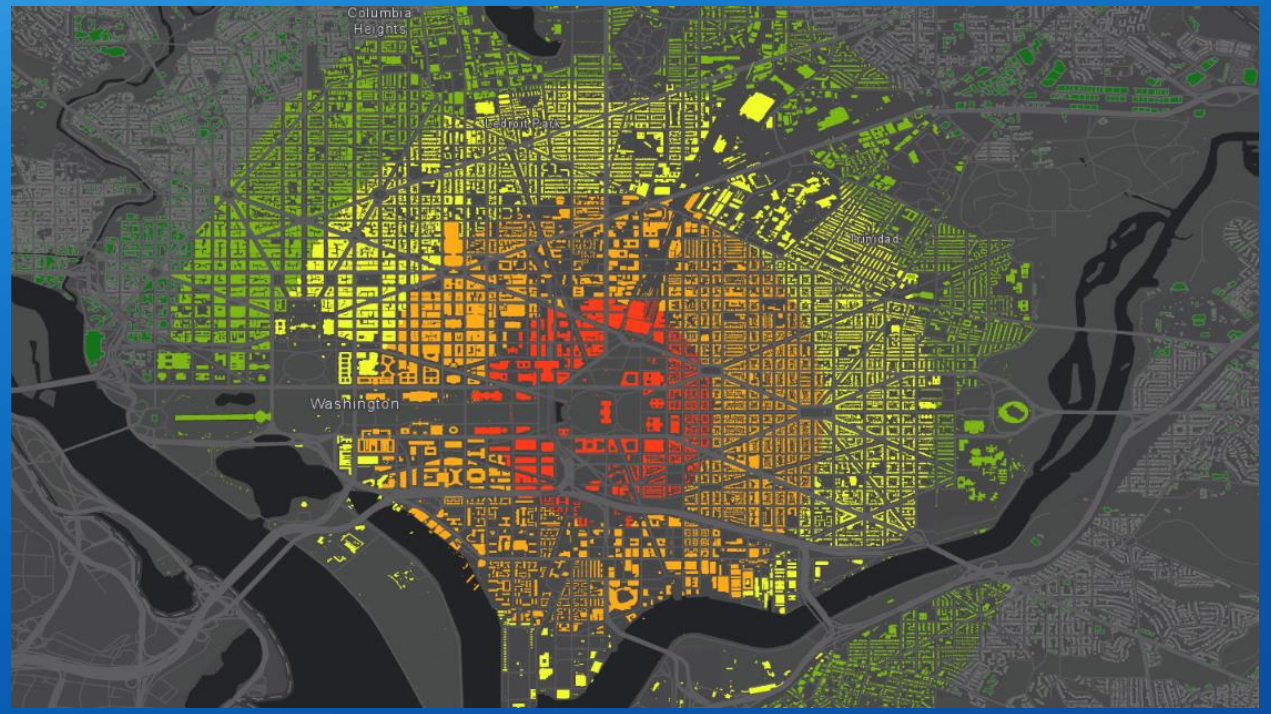

```
<script type="text/plain" id="wind-direction">
  // WIND DIRECTION

  var DEG = $feature.WIND_DIRECT;
  var SPEED = $feature.WIND_SPEED;
  var DIR = When( SPEED == 0, null,
    (DEG < 22.5 && DEG >= 0) || DEG > 337.5, 'N',
     DEG >= 22.5 && DEG < 67.5, 'NE',
     DEG >= 67.5 && DEG < 112.5, 'E',
     DEG >= 112.5 && DEG < 157.5, 'SE',
     DEG >= 157.5 && DEG < 202.5, 'S',
     DEG >= 202.5 && DEG < 247.5, 'SW',
     DEG >= 247.5 && DEG < 292.5, 'W',
     DEG >= 292.5 && DEG < 337.5, 'NW', null );
  var WIND = SPEED + ' mph ' + DIR;
  return WIND;
</script>

<script>
 var windArcade = document.getElementById("wind-direction").text;

    var windClass = new LabelClass({
      labelExpressionInfo: {
        expression: windArcade
      },
      labelPlacement: "above-right",
      minScale: 2500000
    });
    windClass.symbol = createTextSymbol("#3ba53f", 11, { x: 3, y: 3 });
</script>
```

GeoAnalytics

Arcade in analysis

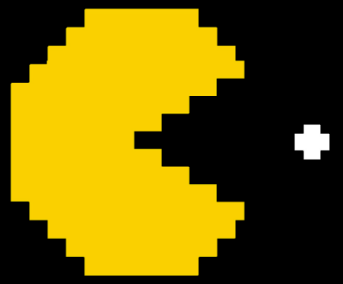
- Create Buffers
- Reconstruct Tracks
- Join Features
- Detect Incidents
- Calculate Field





Arcade: Weather Station

Platform Language



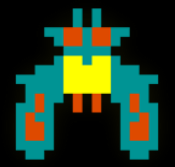
Jason Smith - US EPA

ESRI's

Arcade

The easy to learn scripting language…

Example 1

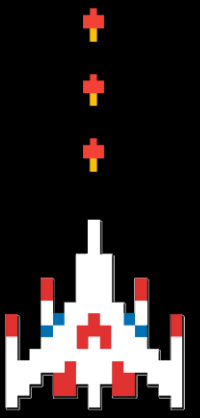


Data Updated: Fri 04/19/2019 12:00 PM CDT

Pollutant	NowCast AQI	oncentration
Ozone	25	29 ppb

Data Updated: Fri 04/19/2019 12:00 PM CDT

Pollutant	NowCast AQI	oncentration
Ozone	87	70 ppb



Example 1

Ozone AQI Background Color

Expression

Test

```
1  Var AQI = $feature.OZONE_AQI
2
3  When (
4  IsEmpty(AQI), '',
5  When (
6  AQI >=0 && AQI <=50, '#00E400',
7  AQI >=51 && AQI <=100, '#FFFF00',
8  AQI >=101 && AQI <=150, '#FF7E00',
9  AQI >=151 && AQI <=200, '#FF0000',
10 AQI >=201 && AQI <=300, '#8F3F97',
11 AQI >=301, '#7E0023','#000000'
12 ))
```

Example 2

(1 of 2)

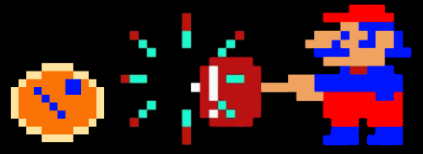
South DeKalb

Monitor 130890002 - Georgia Department of Natural Resources

Current Ozone Air Quality is **Good**

Data Updated: Fri 04/19/2019 01:00 PM EDT

Pollutant	NowCast AQI	Concentration
Ozone	26	28 ppb



Example 2

Ozone AQI Category

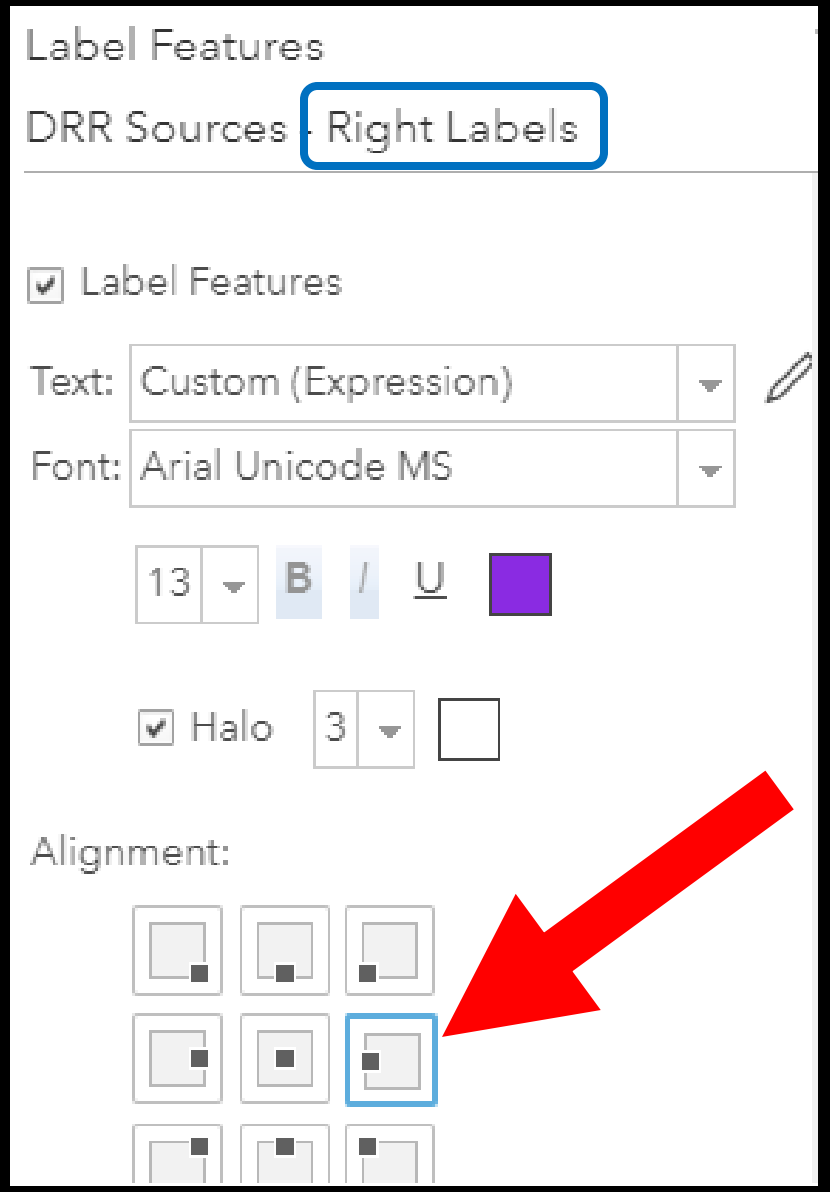
Edit

Expression

Test

```
1  Var AQI = $feature.OZONE_AQI
2
3  When (
4  IsEmpty(AQI), '',
5  When (
6  AQI >=0 && AQI <=50, 'Good',
7  AQI >=51 && AQI <=100, 'Moderate',
8  AQI >=101 && AQI <=150, 'Unhealthy for Sensitive G
9  AQI >=151 && AQI <=200, 'Unhealthy',
10 AQI >=201 && AQI <=300, 'Very Unhealthy',
11 AQI >=301, 'Hazardous', 'Undefined')
12 )
```

Example 3


Label Features
DRR Sources
Right Labels
Label Features
Text:
Custom (Expression)
Font:
Arial Unicode MS
13
B
I
U
Halo
3
Alignment:

Example 3

Show Certain Labels

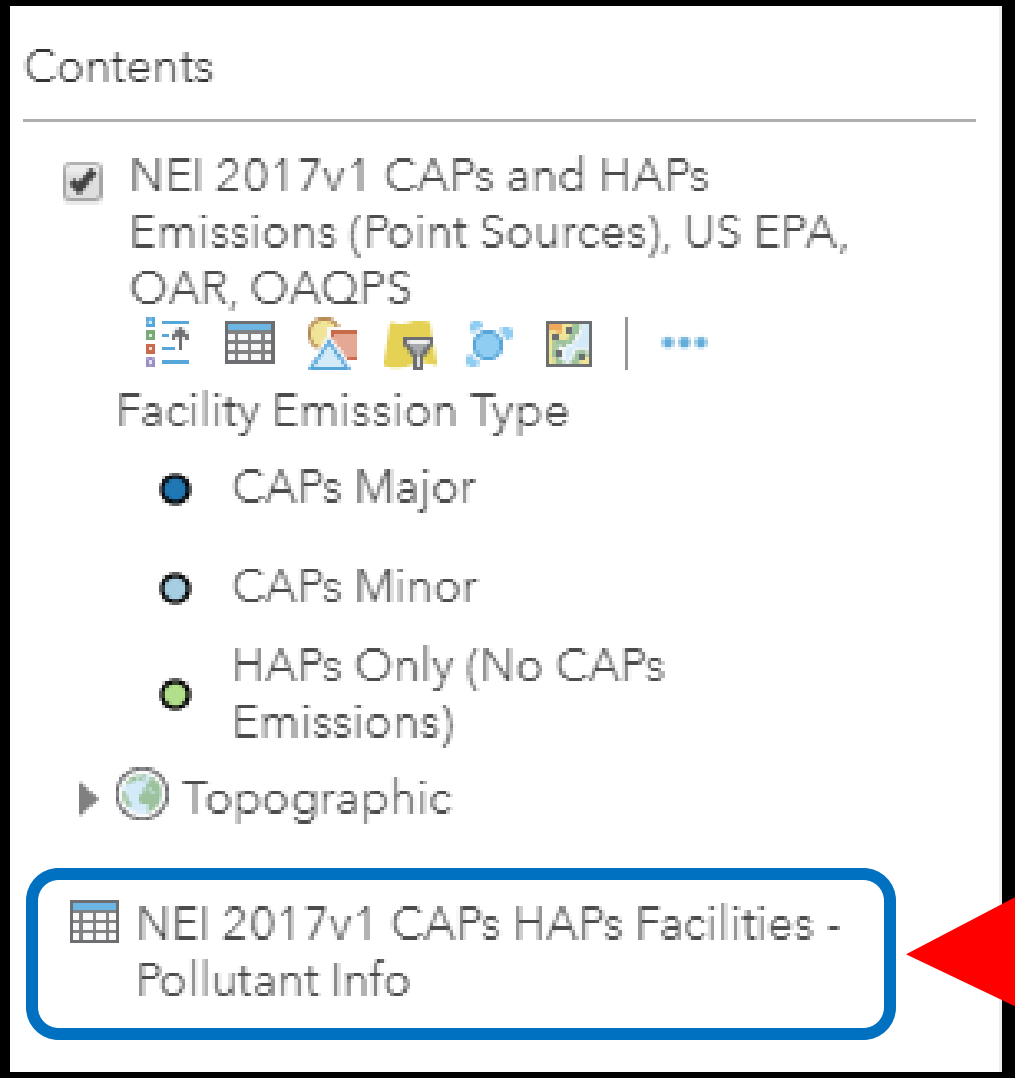
Edit

Expression

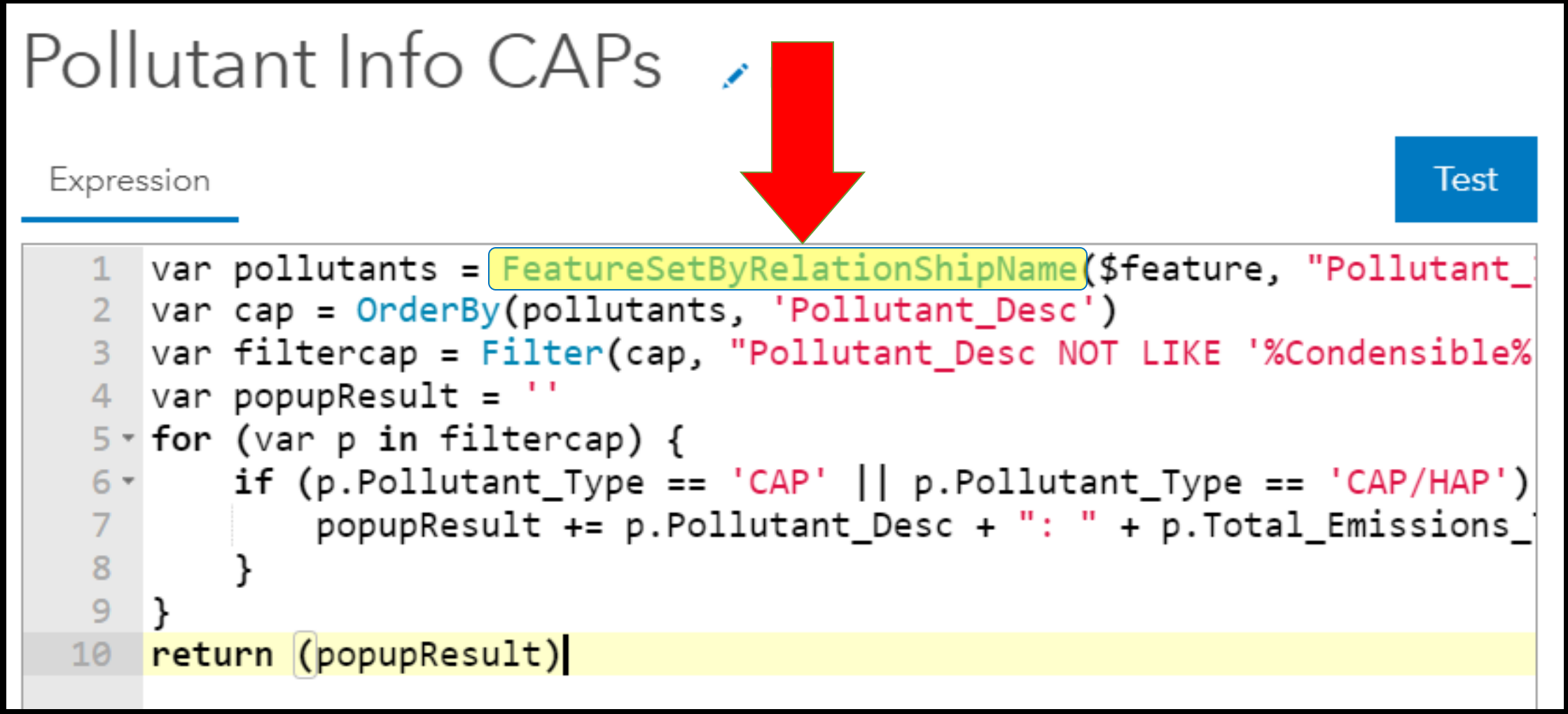
Test

```
1  if ($feature.Facility_short == "Flint Creek"||
2  $feature.Facility_short == "Entergy-Independence"||
3  $feature.Facility_short == "Plum Point"||
4  $feature.Facility_short == "Cabot - Ville Platte"||
5  $feature.Facility_short == "CLECO - Brame"||
6  $feature.Facility_short == "Oxbow - Baton Rouge")
7
8  {
9      return $feature.Facility_short
10 }
```

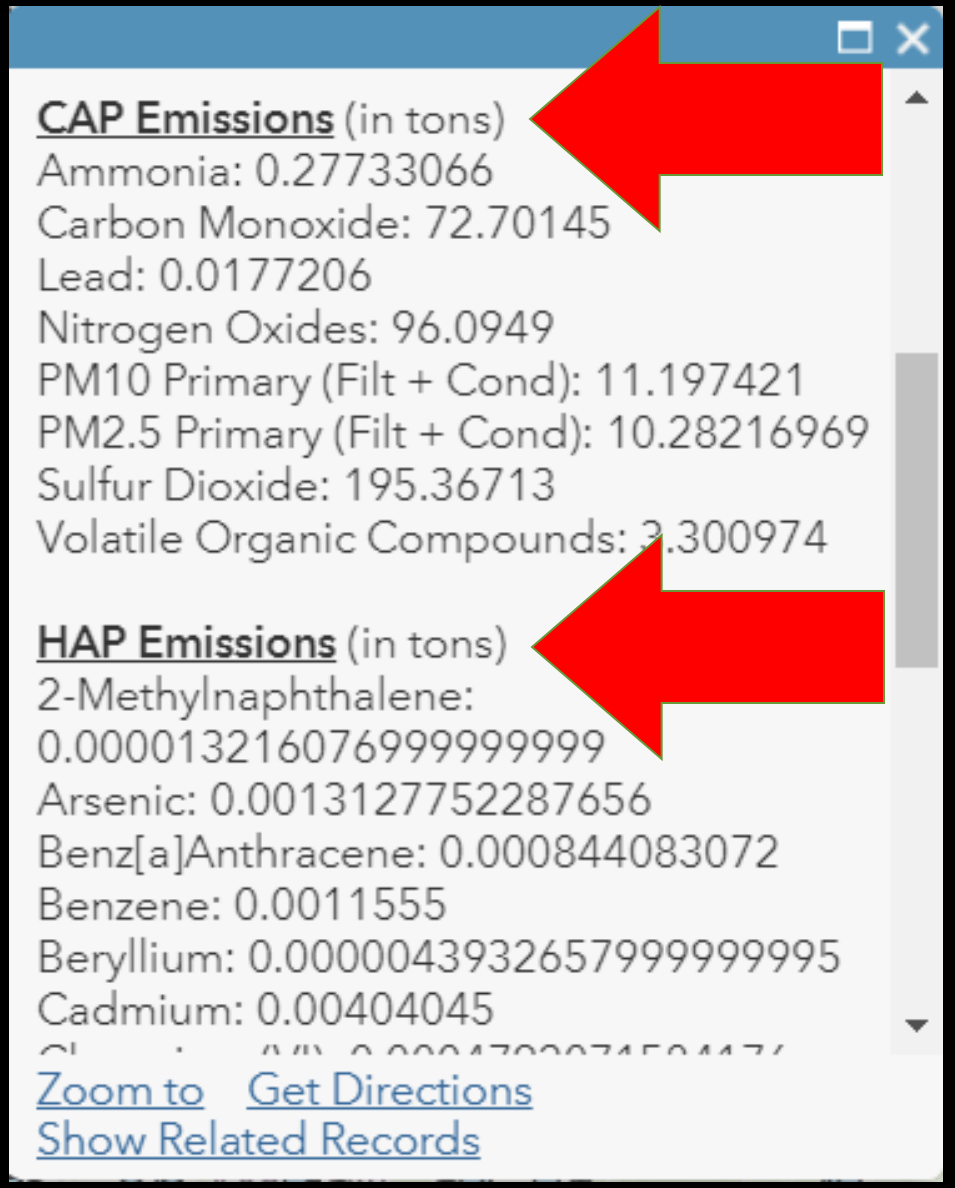
Example 4


Contents
NEI 2017v1 CAPs and HAPs Emissions (Point Sources), US EPA, OAR, OAQPS
Facility Emission Type
CAPs Major
CAPs Minor
HAPs Only (No CAPs Emissions)
Topographic
NEI 2017v1 CAPs HAPs Facilities - Pollutant Info

Example 4


Pollutant Info CAPs
Expression
Test
1 var pollutants = FeatureSetByRelationShipName($feature, "Pollutant_
2 var cap = OrderBy(pollutants, 'Pollutant_Desc')
3 var filtercap = Filter(cap, "Pollutant_Desc NOT LIKE '%Condensible%
4 var popupResult = ''
5 for (var p in filtercap) {
6 if (p.Pollutant_Type == 'CAP' || p.Pollutant_Type == 'CAP/HAP')
7 popupResult += p.Pollutant_Desc + ": " + p.Total_Emissions_
8 }
9 }
10 return (popupResult)

Example 4


CAP Emissions (in tons)
Ammonia: 0.27733066
Carbon Monoxide: 72.70145
Lead: 0.0177206
Nitrogen Oxides: 96.0949
PM10 Primary (Filt + Cond): 11.197421
PM2.5 Primary (Filt + Cond): 10.28216969
Sulfur Dioxide: 195.36713
Volatile Organic Compounds: 3.300974
HAP Emissions (in tons)
2-Methylnaphthalene:
0.00001321607699999999
Arsenic: 0.0013127752287656
Benz[a]Anthracene: 0.000844083072
Benzene: 0.0011555
Beryllium: 0.00000439326579999999995
Cadmium: 0.00404045
Zoom to
Get Directions
Show Related Records


ARCADE
GAME OVER
ON!

Resources

- Arcade Documentation
  - *https://developers.arcgis.com/arcade/*
- Arcade Documentation on Github
  - *https://github.com/esri/arcade-expressions*
- Playground
  - *https://developers.arcgis.com/arcade/playground/*
- Using Arcade in JS API
  - *https://developers.arcgis.com/javascript/latest/guide/arcade/index.html*
- Using Arcade in GeoAnalytics Server
  - *https://enterprise.arcgis.com/en/server/latest/get-started/windows/geoanalytics-calulate-field-expression.htm*

Examples Given During Presentation:

- Hello World & Flooding Example – Matt
  - *Map Viewer*
- Predominant Ethnicity): Map Viewer & Map Viewer Beta – David
  - *Map Viewer (Click this one if you want to see the code)*
  - *Map Viewer Beta (Click this one if you have and AGOL account and want to see in new Map Viewer Beta)*
- Crime Data – Temporal, Intensity, & Distance to Police Stations - David
  - *Map Viewer (Also the Arcade Popup & Label Example)*
- Weather Station Example (Multi-Function Label) – Matt
  - *Map Viewer Beta*
- Also, If you want the code please send an email to dattaway@esri.com and mberra@esri.com and we'll be happy to pass along the Arcade code