GEO TUTORIAL

UNLOCK A HIDDEN POTENTIAL OF ARCADE EXPRESSIONS IN ARCGIS ONLINE MAP VIEWER

Kate Grala
Andrew Nagel
John Cartwright

Geosystems Research
Institute Mississippi
State University

MAY 2024

This work was supported through funding by the National Oceanic and Atmospheric Administration Regional Geospatial Modeling Grant, Award # NA19NOS4730207.

Geosystems Research Institute, Mississippi State University, Box 9627, Mississippi State, MS 39762-9652

The Geospatial Education and Outreach Project (GEO Project) is a collaborative effort among the Geosystems Research Institute (GRI), the Northern Gulf Institute (a NOAA Cooperative Institute), and the Mississippi State University Extension Service. The purpose of the project is to serve as the primary source for geospatial education and technical information for Mississippi.

The GEO Project provides training and technical assistance in the use, application, and implementation of geographic information systems (GIS), remote sensing, and global positioning systems for the geospatial community of Mississippi. The purpose of the GEO Tutorial series is to support educational project activities and enhance geospatial workshops offered by the GEO Project. Each tutorial provides practical solutions and instructions to solve a particular GIS challenge.

UNLOCK A HIDDEN POTENTIAL OF ARCADE EXPRESSIONS IN ARCGIS ONLINE MAP VIEWER

Kate Grala[1] (kgrala@gri.msstate.edu)
Andrew Nagel[1] (andrewn@gri.msstate.edu)
John Cartwright[1] (johnc@gri.msstate.edu)

[1] Geosystems Research Institute,
Mississippi State University

REQUIRED RESOURCES

- ArcGIS Online Map Viewer (tutorial last tested on April 09, 2024).
- Access to an ArcGIS Online account (at minimum, the User role needed).

FEATURED DATA SOURCES

https://services1.arcgis.com/URmJp8f6MBqak7GG/arcgis/rest/services/WoodyWetlands2001/FeatureServer

OVERVIEW

The primary objective of this tutorial is to outline the assessment of multiple attribute fields in ArcGIS Online (AGOL) Map Viewer. At the same time, you will learn how to incorporate an Arcade conditional statement into a symbology expression. The goal is to symbolize each point using one of two characteristics based on similarities and differences between the existing fields. When you complete the tasks listed below, you will be able to synthesize information from multiple attribute fields and display it on the map without creating an additional field. To implement this tutorial, basic familiarity with the AGOL Map Viewer is needed, but no prior Arcade knowledge is necessary.

Let's assume you are working on a group project that assesses wetland conditions in coastal Mississippi. During the team meeting, you and your friend offered to make an online map illustrating change in woody wetlands cover while the rest of the group agreed to design a dashboard featuring your map. The following tutorial will help you accomplish your task by teaching you how to create a unique symbology for point locations using an Arcade expression.

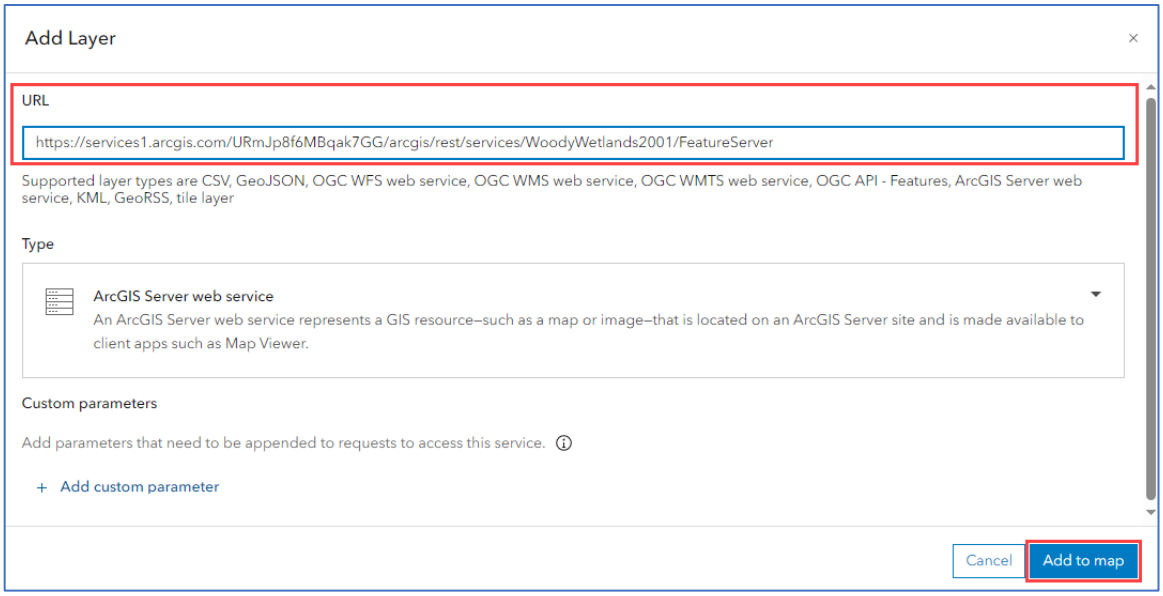
STEP 1. CREATE A WEB MAP AND REVIEW THE TUTORIAL DATA

Your friend prepared the wetlands feature layer and sent you its URL address. He called you to verify that you can access the data and asked if you would not mind completing the map without him. He apologized for the short notice and explained that he was joining emergency response efforts in a hurricane-impacted area. He promised

to check his messages, but he anticipated to be out of reach for a couple of days. You assured him that designing the map would be your top priority.

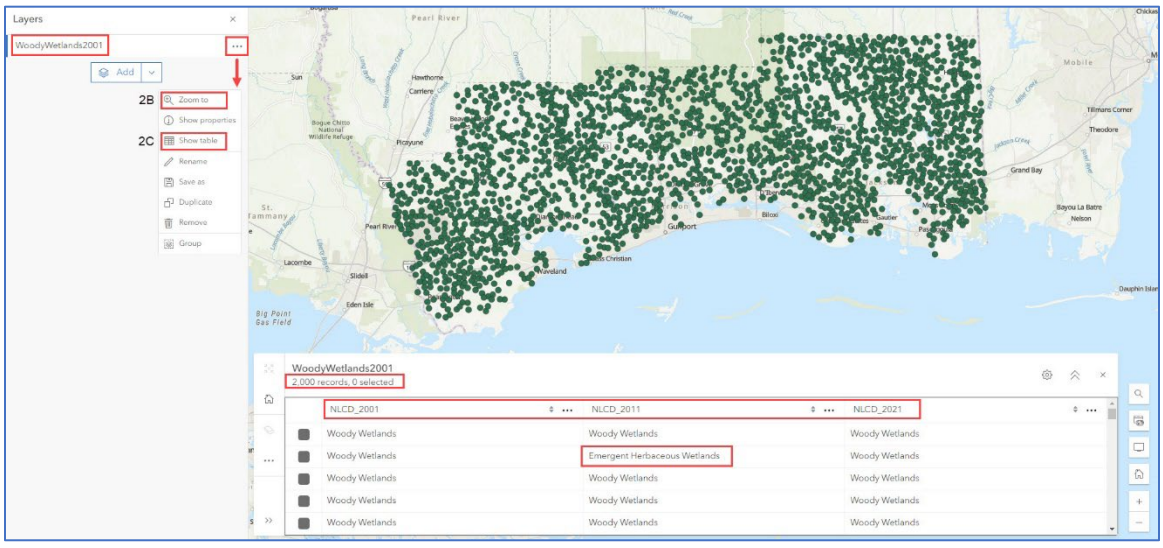
Now, it is up to you to get this task finished. In this step, you are going to create a web map and add a layer from the provided URL address.

A. Sign In to your AGOL account, open the Map Viewer, and add a Feature Layer to your map using the link provided in the Feature Data Sources section (Layers> Add > Add layer from URL).



B. The map should automatically focus on the extent of the inserted layer. If not, click the ellipsis button adjacent to the layer name and select the Zoom to option. The green points you see on the screen represent the Woody Wetlands category extracted from the 2001 NLCD Land Cover dataset and cover the extent of Mississippi coastal counties.

C. Click the ellipsis button again to access the context menu for the WoodyWetlands2001 layer and select the Show table option. Verify the record count and examine the current fields. The table consists of 2,000 records, each described by three attribute fields representing land cover classification from three different time periods. The NLCD_2001 field has only one category: Woody Wetlands. The remaining two fields classify most points as Woody Wetlands as well. However, sporadically, there are other land cover classes, such as Emergent Herbaceous Wetlands, as shown in the figure below.



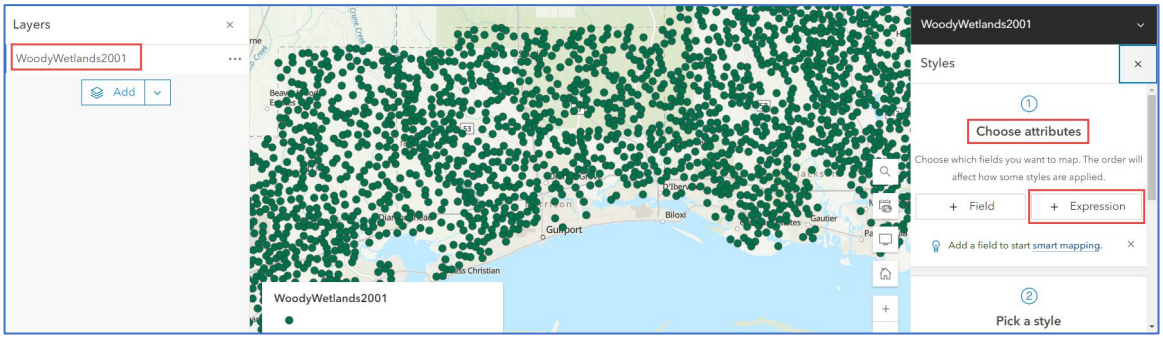
D. Scroll down the table and examine the NLCD_2011 and NLCD_2021 fields. What other land cover categories do you see in the table? Remember, your objective is to detect points that, according to either the 2011 or 2021 classification, do not represent Woody Wetlands. As you can expect, a visual evaluation of 2,000 records for multiple different fields is not practical. You must find a different way of completing this task. Take a minute and think what the best way would be to evaluate the existing NLCD-related fields.

E. Up until now, you're delivering on your promise. You have successfully verified that you can add the provided feature layer to the map and that you can view the associated records. While contemplating a more efficient analysis approach, you can close the attribute table and save the web map to your AGOL account.

STEP 2. BUILDING AN ARCADE EXPRESSION TO CHANGE LAYER DISPLAY

After a short deliberation, you settled on the idea of creating a new attribute field and calculating its values based on already existing fields. More specifically, you decided to create a calculation expression that would help you to determine how many points represented Woody Wetlands in 2001, 2011, as well as 2021. This is a valid and commonly used approach. However, you noticed that you are not able to add a new field to the attribute table, and more importantly, you can't make any edits to the provided dataset. Since your friend is out of reach, you contacted the other team members and asked for advice. They confirmed your concern that the ability to add fields is available for hosted feature layers, but only the owner is permitted to make edits or enable this option for other users. However, they offered an alternative solution of using Arcade to change the symbology of the feature layer, and the good news is, you don't have to be the owner of the dataset to make this work. Here, you will find a detailed explanation of how you can accomplish this.

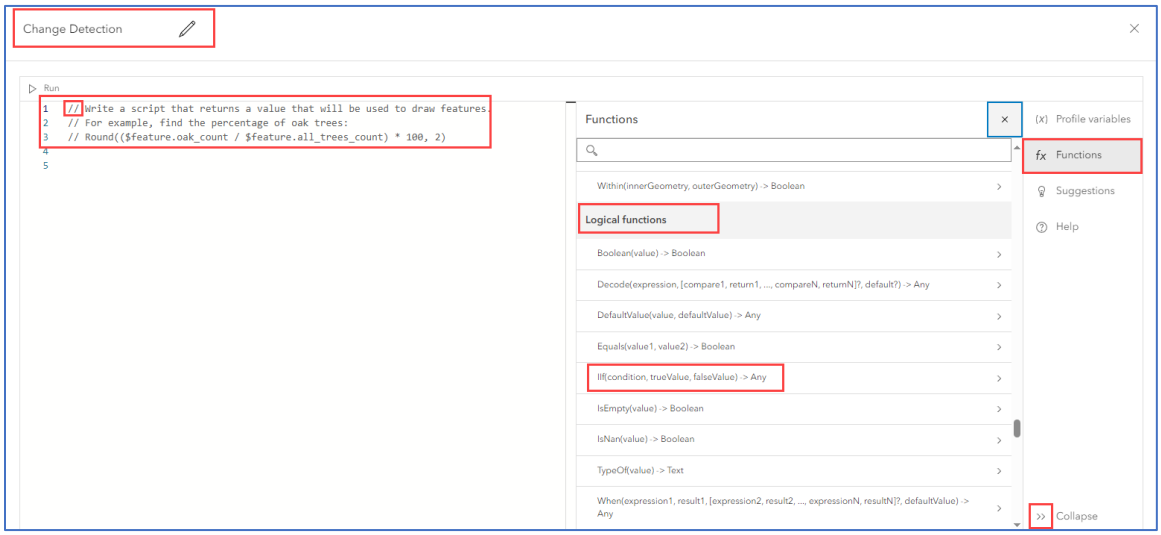
A. You will change the layer's symbology properties, so make sure that the Woody Wetlands_2001 is the active layer. From the panels on the right, select Styles, and click on the + Expression button located under the Choose attributes options.



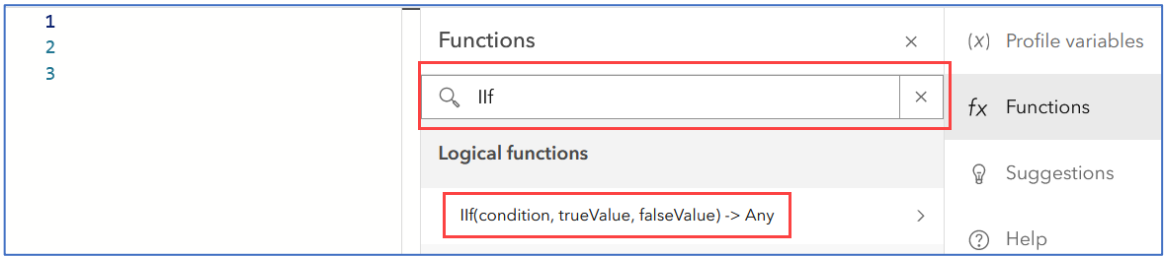
B. This will open the Arcade expression editor. Start by renaming the expression name to Change Detection. It is beneficial to assign meaningful names to your expressions, as this will help you keep your AGOL map well organized. You can think of the expression names as you would of the field names, and although the values resulting from the expressions will not be saved in the layer's attribute table, they will be saved in your map file. This way, you will be able to recognize what each expression represents, especially when you create numerous Arcade expressions in your map project. Also, the expression name will generate a heading in the symbology legend once the expression is implemented. Make a note of this step, circle back to it later in the tutorial, and determine if this minor effort was worth your while.

C. Explore the Arcade editor options and read the provided code example. Note the two forward slashes (//) at the beginning of each code line. In Arcade, they indicate comments and are used to explain code in plain language. Comments provide an easy way to describe code functionality and are particularly important if you intend to share your expression with others. However, they are ignored when the expression is run. This means the provided 3-line sample is not an executable code and does not produce any results. You will type your expression in the main window, where the numbered comment lines now appear. None of the elements of the provided sample code will be incorporated, so feel free to delete all three comments.

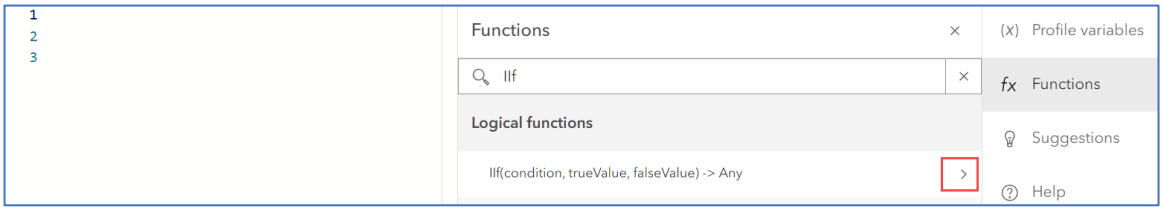
D. Expand the **Options** panel available on the right (**<<**). If needed, you have an option to **Collapse** this panel (**>>**). This is helpful, especially if some of the longer expression lines do not fit into the existing window. From the available menu choices, click on the ***fx* Functions** option and scroll down through the displayed list till you reach the **Logical functions** section. Review the code for the **IIf()** function. This is how the conditional if() statement is constructed in Arcade. You will use this function to generate the symbology expression for the **WoodyWetlands2001** layer. If you are familiar with JavaScript, you may have seen a very similar code structure already since Arcade was created using similar programming principles.



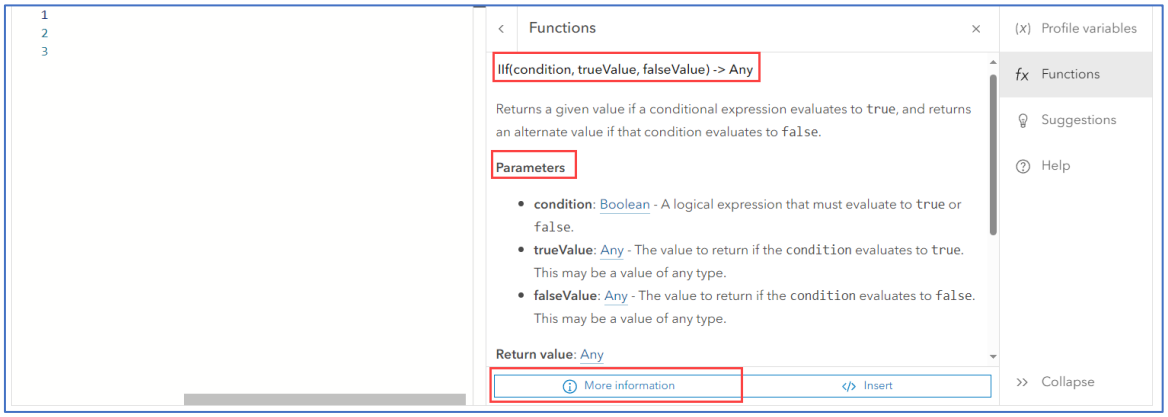
E. Go back to the ***fx* Functions** menu and scroll down to the bottom of the list. As you can see, the list of available Arcade built-in functions is rather extensive, so scrolling through the list is not the most efficient way of finding the function you are looking for. As described above, we will be using the **IIf()** function, so instead of scrolling up and down the function menu, in the search window, start typing **IIf,** and the function you need will promptly appear.



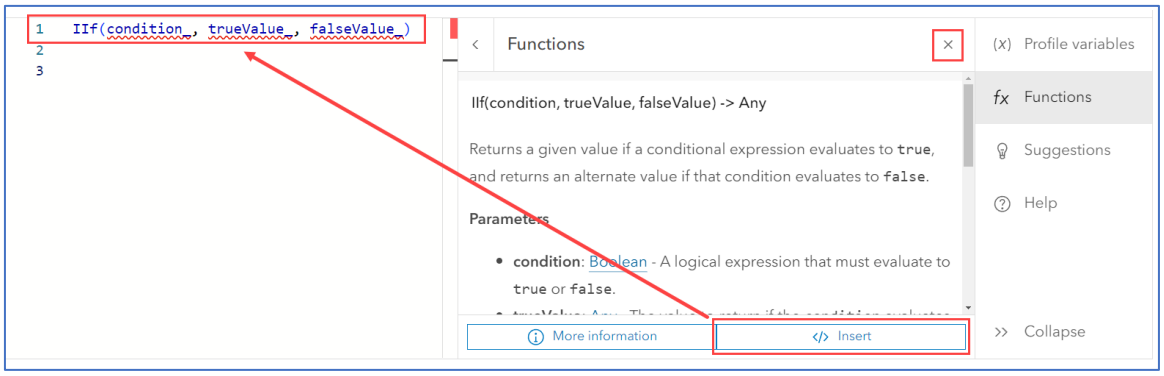
F. It is time to learn how the conditional statement is constructed in Arcade. Click on the arrow symbol shown at the end of the function code (**>**) to access the existing function documentation.



G. Read the displayed explanation and try to understand the provided code sample. Pay attention to the list of parameters and click on the available description links. If you would like to see additional clarification, select the ⓘ More information option.

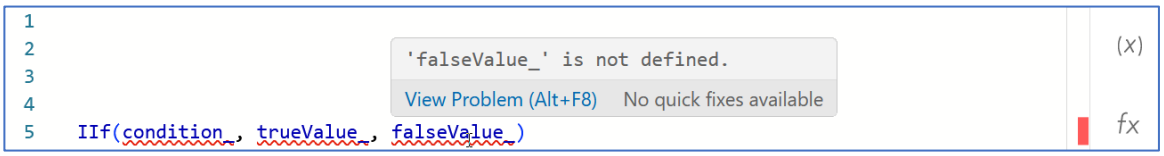


H. Once you familiarize yourself with the logic behind the conditionals in Arcade, click on the </> Insert button. This will enter the code syntax into the first line of the expression window. Close the function panel (x) to make more room for editing the expression.



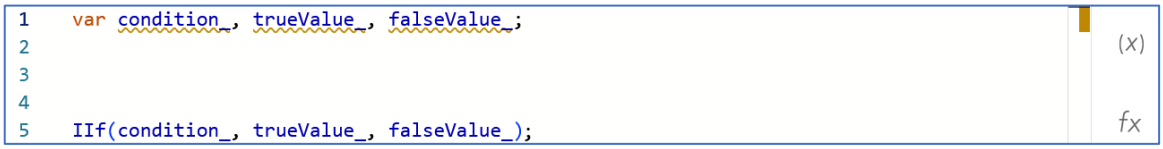
I. You could start modifying that line of code right away, but this is a good opportunity to learn how to define variables in Arcade. This way, you can keep the inserted line of code completely intact and make it a key element of your expression. Place the cursor at the beginning of the first line and press Enter at least four times. This should give you enough room to insert the missing expression elements.

J. Note that the Arcade editor automatically underlined all function parameters with red squiggly lines. If you hover your mouse over one of the parameters, you can learn more about the source of the problem. The listed error description should not be surprising, as none of the parameters are defined at this point.

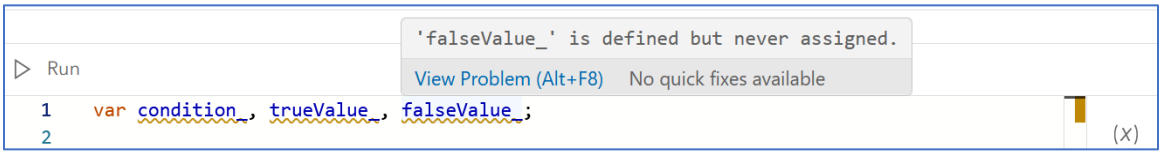


K. To define variables in Arcade, the var keyword is used. Place the cursor on the first line, type in var, and add a space. Next, you will name three empty variables, one for each of the main expression elements. Technically, the order and capitalization of the variable names are unimportant in Arcade, but it is a good practice to arrange variables in the order they appear in the expression. Name your variables condition_, trueValue_, and falseValue_. This way, they are listed in the same order and look identical to the code parameters you inserted in step 2H. In fact, you could copy their names directly from the automatically inserted line of code. Don't overlook the underscores at the end of each variable name, and make sure that they are comma-separated. Also, even though this is not required, it is customary in Arcade to add a

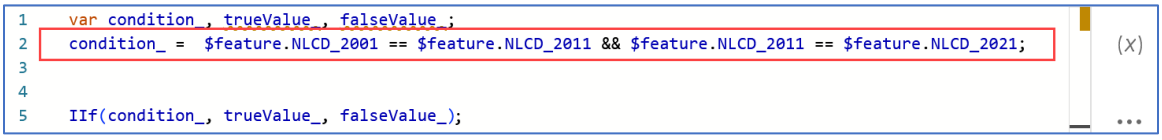
semicolon at the end of each line. Now, you should have some code listed in the first and fifth lines, as shown below.



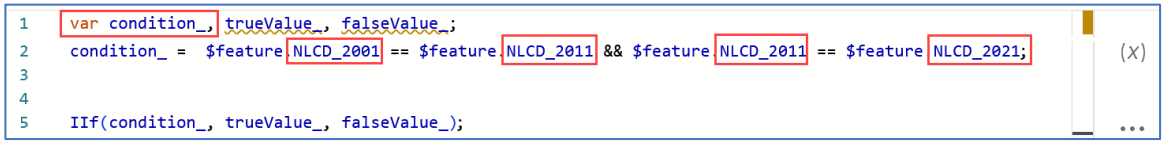
L. Note that inserting the first line fixed the errors in line five. However, a new type of error is now highlighted in the first line. This new error indicates that although the variables are correctly defined, they need to be assigned to a value.



M. Assigning values to variables means that each variable will represent something and that you can reuse this value later in the subsequent lines of your expression. In Arcade this is done with a single equal sign (=). Place the cursor on the second line and type the statement defining the condition_ variable as shown in the second line in the figure below.

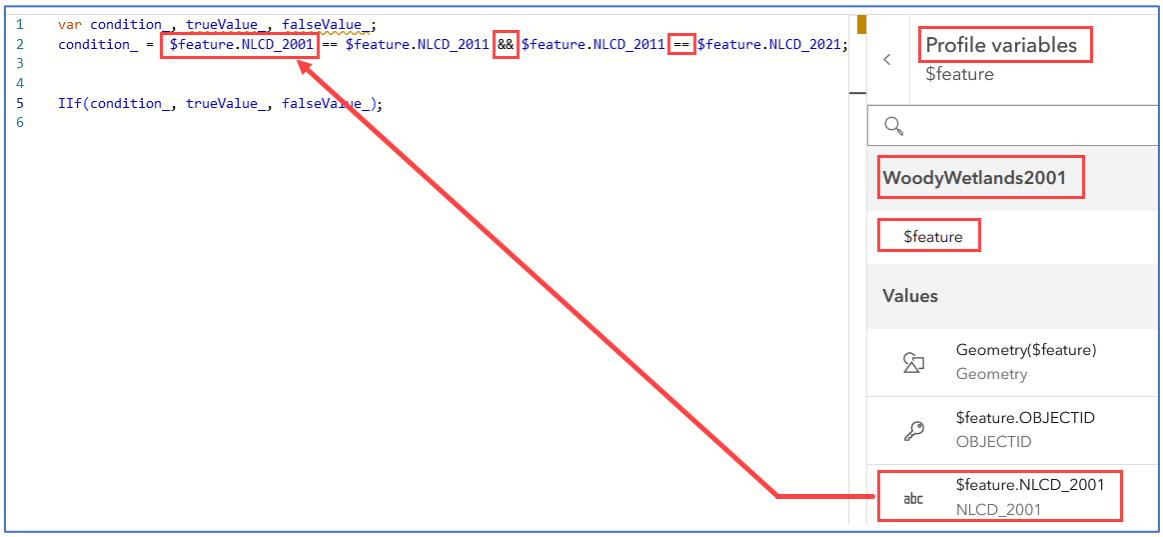


N. This line of code is the key element of the expression where you compare the attribute values in the available NLCD-related fields. When the statement assigned to the condition_ variable is evaluated as true, it identifies the point locations categorized as Woody Wetlands based on the examined attributes. This is the most important idea to understand in this tutorial. At this point, you may be puzzled about this line of code. However, the following three field names should be familiar: NLCD_2001, NLCD_2011, NLCD_2021. Before moving to the next step, note that the assignment error is now cleared for the condition_ variable. Also, if you are wondering if there are more efficient ways to enter the long line of codes in Arcade and what these extra characters in between the attribute field names represent, please keep reading.



O. While typing the code manually is a commonly used option, it is also an easy way to introduce syntax errors. Instead, you can copy and paste the code directly into the Arcade expression window. However, this will only work if you have the exact required code accessible in a text format. As shown above, that is not always the case. Therefore, it is helpful to know how this line of code can be constructed from scratch and to understand the key elements of its syntax. First, according to the documentation, the condition variable must be assigned to a Boolean value, which evaluates results as either true or false. Second, what is unique to Arcade are the Profile variables. They represent data values used as inputs in Arcade expressions and always start with the $ character. The profile variable $feature represents the attributes and geometry of the feature layer for which the expression is created. In this case, it is the WoodyWetlands2001 layer. To access the field attributes of this layer, you need to use a dot notation (i.e., $feature.NLCD_2011). You can access available Profile variables by clicking the (x) option on the panel to the right. Click on the arrow (>) at the end of the line options next to the $feature to display the profile variables for the WoodyWetlands2001 layer. Each of the listed Profile variables can be directly added to the expression window by simply clicking on it. Using this functionality greatly reduces the chances of introducing syntax errors. Finally, you need to be familiar with two types of Arcade operators. Double

equal sign (==) is a type of comparison operator and represents equality. In this case, you are using the equality operator to check if the values in the attribute fields are identical. Here, this needs to be done twice—first to check similarities between 2001 and 2011 fields and then to look for similarities between 2011 and 2021 fields. The operator connecting and evaluating these two conditions is known as logical and (&&). This means that conditions on each side of the $$ operator must be true, otherwise the entire statement will be considered false. Apply these concepts and try to recreate the second line of code on your own.



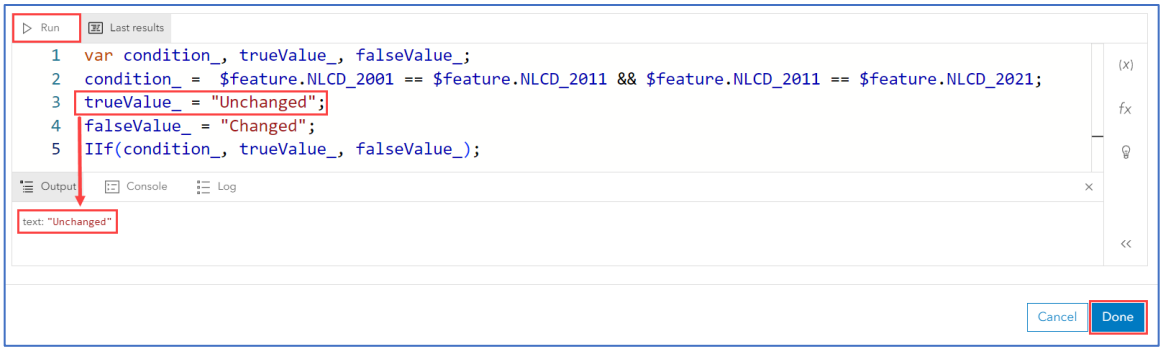
P. As noted in the documentation, the last two variables can be assigned to Any data type. For example, this can be a number, text, date, or time value. Here, you will assign a text value to both variables. Note that the assigned text must be placed in quotes. Both double and single quotes are acceptable in Arcade, but they must be straight quotes. The use of curly quotes will result in a syntax error. Place the cursor on the third line and type in the following: `trueValue_ = "Unchanged"`. This line of code represents values evaluated as true (points that are classified as Woody Wetlands in 2001, 2011, and 2021).

Q. Place the cursor on the fourth line and type in the following: `falseValue_ = "Changed"`. This line of code represents values evaluated as false (points that are not classified as Woody Wetlands either in 2001, 2011, or 2021). Be sure to add semicolons at the end of lines 3 and 4.

R. Verify that your expression matches the code shown in the figure below. Notice that all the error indicators are cleared.

```
1  var condition_, trueValue_, falseValue_;
2  condition_ =  $feature.NLCD_2001 == $feature.NLCD_2011 && $feature.NLCD_2011 == $feature.NLCD_2021;
3  trueValue_ = "Unchanged";
4  falseValue_ = "Changed";
5  IIf(condition_, trueValue_, falseValue_);
```

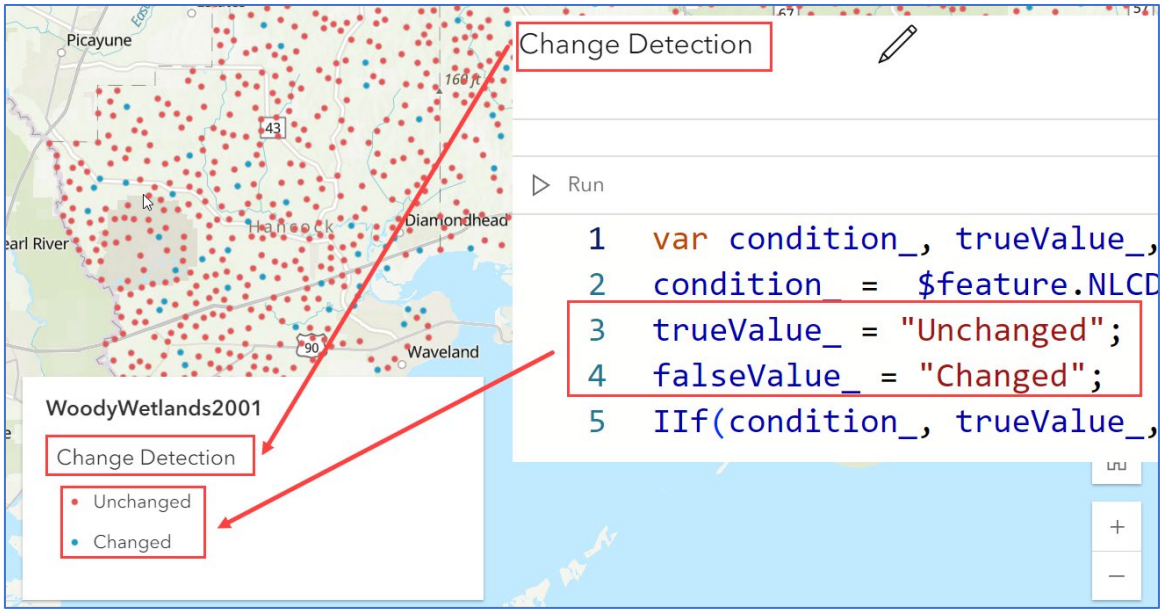
S. If you notice problems, repair the existing errors to the best of your abilities, and when corrected, click the Run button in the top left corner to execute the expression. This should produce an example of a text output just below the expression window that represents the value assigned to either the trueValue_ or falseValue_ . This is a good sign and a clear indication that your expression is constructed correctly. If that is the case, click the Done button.



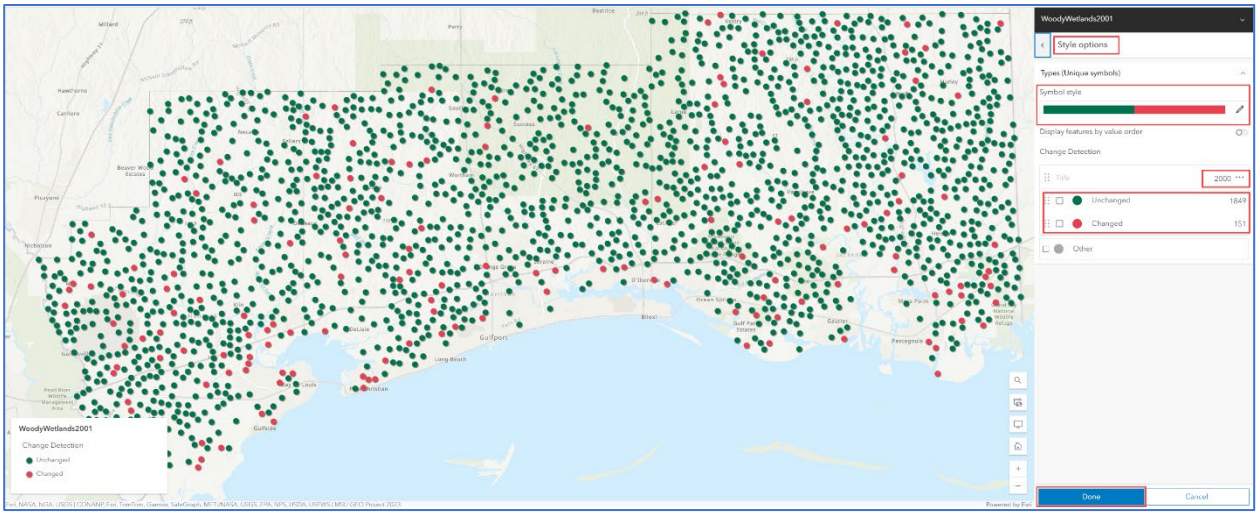
T. If you still encounter problems, delete the code you entered. Instead of starting from scratch, copy and paste the text below and run the expression again.

```
var condition_, trueValue_, falseValue_;
condition_ = $feature.NLCD_2001 == $feature.NLCD_2011 && $feature.NLCD_2011 == $feature.NLCD_2021;
trueValue_ = "Unchanged";
falseValue_ = "Changed";
IIf(condition_, trueValue_, falseValue_);
```

U. Once the operation is successful, review the results displayed. Note that the legend automatically added to your map reflects the text values you assigned to variables representing true and false in the expression and that the heading added to the legend reflects the name of your expression (review step 2B).



V. Modify the automatically generated style to better visualize the change in woody wetlands cover in the mapped area before sending the map to your teammates. Select your favorite colors and choose an appropriate symbol size. You can also send them a short note on how many points were affected by the change during the analyzed period. The results should be the same as the figure on the next page. Of the initial 2,000 points, 151 have changed the land cover class from woody wetlands at least once during the evaluated time frame. Save your map. You can message your friend to let him know the portion of the group project you both volunteered to work on is completed. You are now ready to share the map with your teammates, and they can start building a dashboard.



This completes our GEO Tutorial on building an Arcade expression to display data created by summarizing multiple fields. The main advantage of the generated expression is the ability to design a unique map utilizing a dataset that you don't own and for which you don't have the editing rights. The expression presented in this tutorial can be used to make visual comparisons between values in multiple (two or more) attribute fields. In this case, you compared text attributes, but the same workflow will work when evaluating numbers, calendar dates, or values in any other format. To further improve the functionality of your online map, you can use this expression to configure feature pop-ups and generate custom labels.

In addition to displaying changes in land cover, you can use the Arcade conditional function to compare the results of different classification analyses, evaluate datasets originating from various sources, or simply look for errors in the revised version of attributes. Consider a few geospatial team projects you worked on in recent years. Can you recall any tasks when incorporating an Arcade expression similar to the one presented above that could have helped you save the day?