Section : . Composer

Module : . Expressions in map composer

Using expressions in map composer

"In this module we explore how expressions can be used in map composer"

Expressions are just simple formulas that QGIS will calculate on the fly. Expressions in QGIS provide a powerful way to include dynamic content in your layers, in an analysis, and various other places. They can also be used to make your map composers change their appearance dynamically. In this module, we will explore the expression system in QGIS Composer to provide some examples of how they may be used to define labels, colours and so on. If you have not already completed our worksheet on basic expression use, you should probably do that first since we will assume that you already know how to use expressions. There are two main ways in which expressions can be used:

1. **Expressions that provide data-driven properties:** For example to specify a feature filter in atlas, or to specify the extents of a map.
2. **Expressions that produce text for labels.** For example, you can use an expression to insert a label on your map which contains the version of QGIS that was used to create the map.

You try:

Goal: To learn how to use expressions in QGIS Composer labels

- Add the provided wards layer and then **recreate the composer map** as shown in the image below.
- Use the specifications in the table below to set the label contents that are shown in the image in red.
  Note that there is a special syntax that QGIS Composer uses when you want to mix expressions with labels it looks something like this:

```
Normal text [% expression % ] more normal text
```

You will need to use this approach for labels which combine standard text and the contents of expressions (such as in the last requirement in the table below). Note that the last requirement only provides hints, you need to work out the complete syntax.

Name	Expectation
Expression for feature count for wards layer The layer_property function allows you to retrieve various properties about a layer by passing the layer's name (as shown in the legend) and the property of that layer you want to display.	[% layer_property('Wards2011','feature_count') %]

Expression for operating system The upper function forces the operating system name to upper case.	[%upper(@qgis_os_name) %]
Expression for QGIS Version Variables are prefixed with @. You also need to use the approach described above to mix in text and expressions.	See variables qgis_release_name and qgis_version

More about using expressions in map composer

The real value of expressions in your map composition is that these expressions are dynamic. Taking our exercise example above, you may have noticed that your operating system box contains different text to the image provided above. That is because when the composer layout draws itself, it checks to see what the current operating system is and then displays that in the label. Similarly, the feature count label will update itself automatically should you add or remove a feature from the wards layer. Try removing one of the wards now and then commit the changes to see how the label **updates itself**.

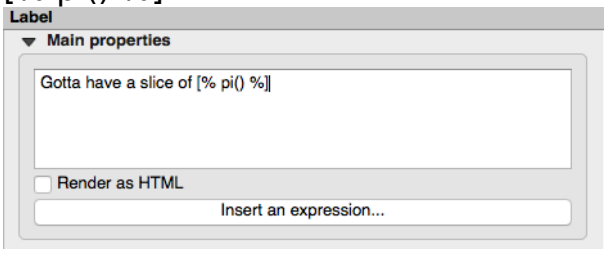
Using **variables** with composer expressions provides a very powerful way to dynamically change the appearance of your map. For example, you can create a variable (in the QGIS Setting dialogue) for your project colour (e.g. `project_colour` set to #BB7364) and then use that variable for data-driven properties for accent colours in your report. By the way, in the specific example given in the last sentence, you might be unfamiliar with the way that colours are described e.g. #BB7364. This is a hexadecimal colour notation, and you can easily obtain these hexadecimal colours from the colour picker widgets in QGIS or from a website like https://htmlcolorcodes.com.

Check your knowledge:

1. When should you use expressions in composer?:
   a. *When you need the contents of text to change dynamically*

b. *When you need to change the properties of a map element based on a project property or other dynamic property.*

2. What would the label produced from this expression look like? Gotta have a slice of [% pi() %]:



a. *Gotta have a slice of pi*
b. *Gotta have a slice of 3.14159265358979*
c. *Gotta have a slice of pi()*

Module video tutorial:

- https://youtu.be/hdKxH5bZq1k

Further reading:

- QGIS Expressions Documentation https://docs.qgis.org/2.14/en/docs/user_manual/working_with_vector/expression.html

Download the sample data for the lesson from http://changelog.qgis.org/media/images/lesson/worksheet/external_data/2237c9e22bbc305f54565bc211a29173b96a8314.zip.