

List Processing: Make a Personal Weather Map with Popfly

Many web sites, news broadcasts, and newspapers use weather maps to display temperature and forecast data on a map for selected cities.

In this lesson you will learn two different techniques for building your own customized weather map, on which you can plot temperature and forecast information for cities you choose.



Prepared by Mark Frydenberg
Computer Information Systems Department
Bentley College, Waltham, MA
mfrydenberg@bentley.edu

List Processing: Make a Personal Weather Map with Popfly



Professor Popfly Mashups Referenced in this Lesson:

- WeatherMap1 (<http://www.popfly.com/users/ProfessorPopfly/WeatherMap1>)
- WeatherMap2 (<http://www.popfly.com/users/ProfessorPopfly/WeatherMap2>)



Learning Outcomes

After completing this lesson, you should be able to:

- Create and process lists of data
- Explain the difference between static and dynamic content
- Split a string at the spaces to create a list of items
- Identify lists of information for which Popfly processes each of its items
- Append text to form longer character strings from shorter ones
- Plot latitude and longitude information on a map

Overview

Many web sites, news broadcasts, and newspapers use weather maps to display temperature and forecast data on a map for selected cities. In this lesson you will learn two different techniques for processing data to build your own customized weather map, on which you can plot temperature and forecast information for cities you choose.

Creating a personalized weather map requires gathering weather data for each of the cities to be plotted, and then “mashing” that data with a map.

Here is an example of such a map from weather.com:



http://www.weather.com/maps/activity/travel/usweathertoday_large.html

Weather.com makes their weather maps available as a service to other information providers. For example, USA Today uses weather maps from weather .com: Visit this page at USA Today <http://www.usatoday.com/weather/temppic/wtempusa.htm> and look closely at the lower right corner of the map. You will notice that the USA Today web site gets the temperature map directly from weather.com.

You can make a customized weather map with Popfly by mashing up weather information from your favorite cities and plotting it on a map. Here’s what the finished Popfly mashup might look like:



Processing Lists

The Weather Map mashup will gather temperature values for several different cities, and store them in a list (sometimes also called an array). A list or array describes a collection of items all of which contain the same type of information. It is common to process each item of a list in the same way: imagine adding up a list of test scores, finding the largest value of all the stocks in your portfolio, accessing one volume from a set of encyclopedias, or printing a form letter customized for each person in your address book. The processing is the same for each item. The only thing that is different is the specific information in each item.

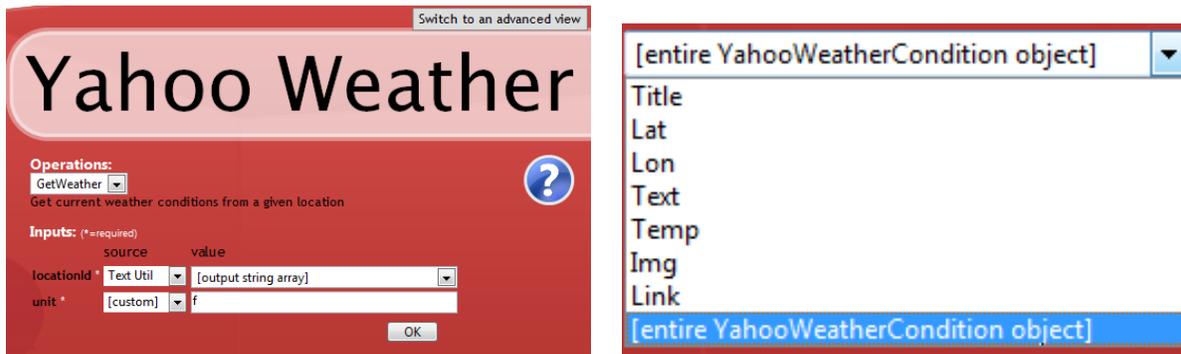
Popfly is smart enough to know that if a block receives an array of items as an input, the block will process all of the items in that array in the same way.

Data Providers

Building the weather map as a Popfly mashup requires combining weather data, latitude/longitude data, and a map.

This implementation makes use of the Yahoo Weather block contributed by Popfly user SebaGomez. (In order to find it, type *weather* in the block search box. There is more than one block named *Yahoo Weather Block*, so be sure you select the right one!) This block has both a GetWeather and a GetForecast operation.

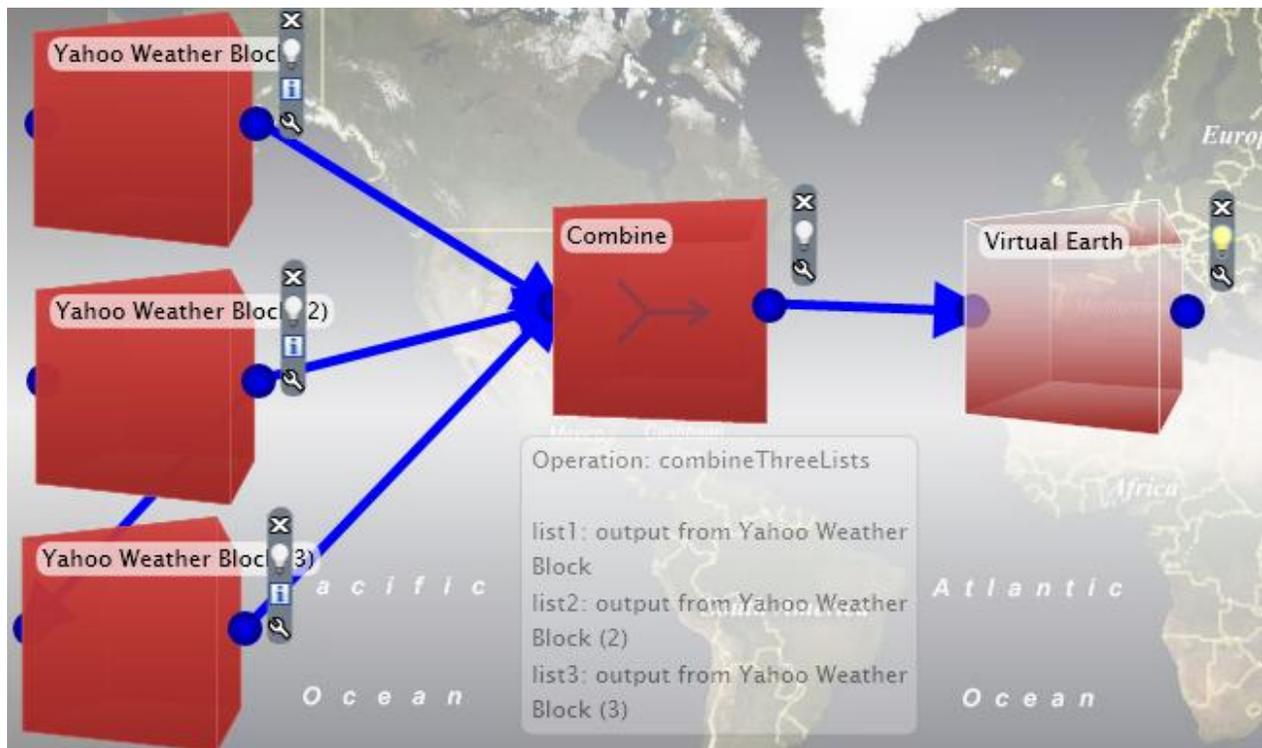
The GetWeather operation returns a YahooWeatherCondition object that contains a title, lat (latitude), lon (longitude), text, temperature, a weather image (clouds, sun, etc.), and a hyperlink to the web site at Yahoo.com with the forecast for this city. The other operation in this block, GetForecast, returns the current conditions and tomorrow's weather forecast for the specified city, along with a hyperlink to the web site at Yahoo.com with the forecast for this city. This operation is not used in the Weather Map mashup.



Because the GetWeather operation also provides latitude and longitude information, that makes it a good choice to use for mapping forecast data. Plotting data on a map requires specifying these coordinates. Some blocks provide latitude and longitude values for you; otherwise, you have to obtain the latitude and longitude yourself for a given location (street address, city, state, or zip code) by using Popfly's GeoNames or YahooGeoCoding blocks in your mashup.

Mashing Up the Weather

One way to design this mashup is to use a separate Yahoo Weather block for each city to plot on the map, and then use the Combine block to combine the weather results, and then pass the output of the Combine block to Virtual Earth. See <http://www.popfly.com/users/professorpopfly/WeatherMap1> for this version of Professor Popfly's Weather Map mashup.



This approach is relatively straightforward: if you want a map with three cities, you need three weather blocks. If you want a map with weather data for six cities, you need six weather blocks.

The Virtual Earth block will add a pushpin for each city whose weather information was combined (into the list) using the Combine block.

Note: Popfly's Combine block has operations to combine up to three blocks whose output all contain the same type of information. The CombineUpto10 block has operations to combine any number of blocks, up to ten, into a single list.

We could summarize the processing of this mashup as follows:

1. Look up weather information for city #1.
2. Look up weather information for city #2.
3. Look up weather information for city #3.
4. Combine these results into a list.
5. For each city in the combined results,
 - a. Place a pushpin and weather information on the map.

Static vs. Dynamic Data

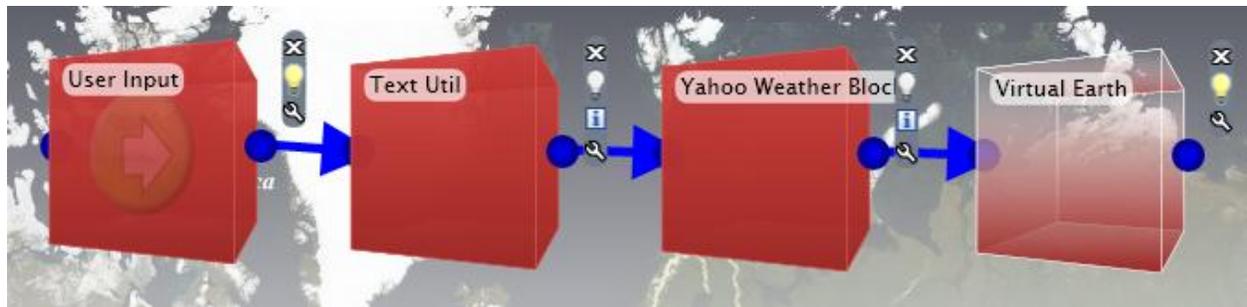
The first example uses a different Yahoo Weather block for each city. This requires knowing how many cities would be plotted on the map before previewing the mashup.

This solution is “static” – while the forecasts are live, the cities for which the forecasts are obtained do not change and cannot be changed when you preview the mashup. Furthermore, the number of cities to plot on the map is determined when the mashup is designed, as it is specified by the number of Yahoo Weather blocks that the mashup combines.

A more general solution, as shown in Professor Popfly's WeatherMap2 mashup, allows the user to enter as many zip codes as desired, and plots all of those cities on the map. In this case, neither the number of cities, nor the cities themselves, are known until the mashup actually runs. In this way, the results are dynamic, as the mashup provides information back to the user based on information from the user.

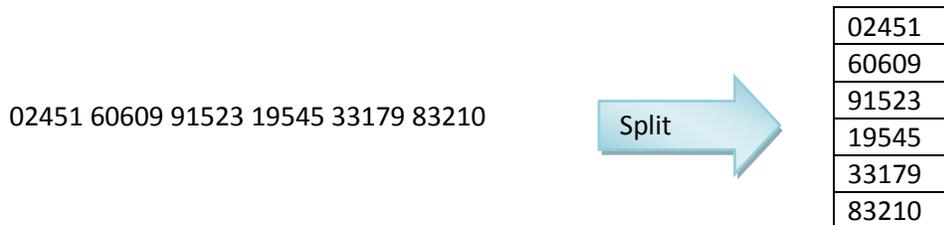
Generalizing the Solution

Consider the Popfly mashup below for creating a weather map.



As shown here, the `getText` operation of the User Input block allows the user to enter a series of zip codes, each separated by a space. All Popfly knows is that the user entered a string of characters, but the mashup designer knows that those spaces perform a special duty: they separate one zip code that the user entered from another.

In order for Popfly to interpret the spaces this way, the mashup uses the Split operation in the Text Util block to split the user's input at the spaces, creating a list of zip codes as a result.



Because the output of the Split operation is a list, the blocks that follow will affect each item in the list. The mashup will iterate over the entire list, applying the same processing steps to each item. Because Popfly dynamically creates the list of zip codes when running the mashup, it can accommodate any number of zip code values that the user specifies. Popfly knows the number of items in the list, so it knows how many times to repeat the steps that follow, once for each zip code in the list.

This summarizes the processing used in WeatherMap2:

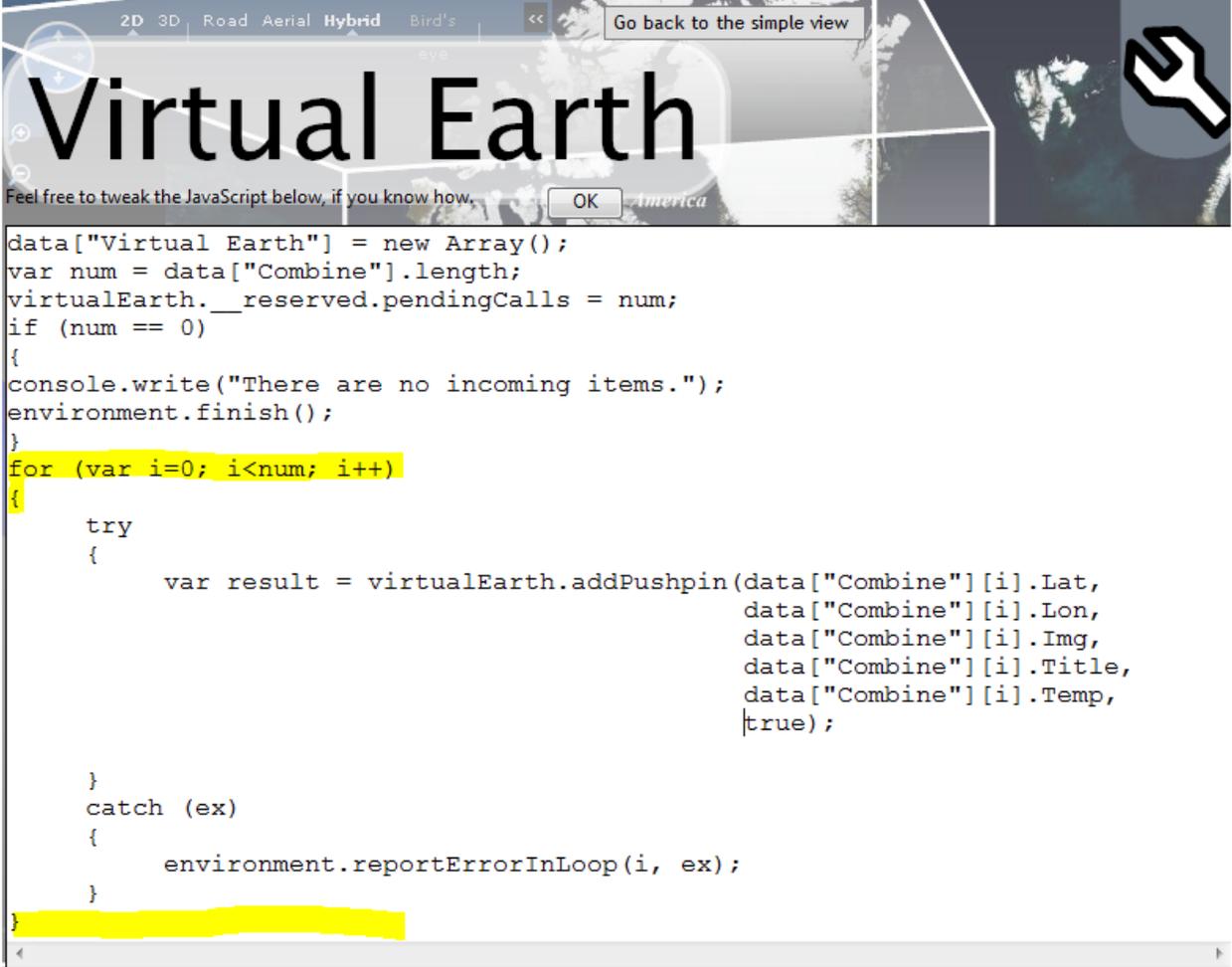
1. Get the zip codes as a string of characters from the user's input.
2. Split the zip codes at the spaces, placing each individual zip code into a list.
3. For each zip code in the list:
 - a. Look up weather information for that zip code.
 - b. Place a pushpin and weather information on the map.

Extra: Add a Hyperlink to the Callout

If you know the HTML tag for a hyperlink, you can include a hyperlink to the Yahoo Weather forecast page in the callout on the map. To do so, connect the output of YahooWeather to a TextAppender block, and use the settings below to create the HTML for a hyperlink. Use the output string from the Text Appender as the title in the callout on the map.

A Look Inside: Where does the looping happen?

Steps 3 a and b in the description of the WeatherMap2 mashup form the body of a loop (a set of repeated instructions) that describes how to process each item. If you're familiar with computer programming, you may have learned of a "for loop" – which describes how to iterate through each item in the list. Popfly hides the "for loop" in the advanced view of the source code of any block that accepts a list as input. For example, click on the *Switch to an Advanced View* button of the Virtual Earth block and you will see that the for loop is implemented in the block's source code.



```
data["Virtual Earth"] = new Array();
var num = data["Combine"].length;
virtualEarth.__reserved.pendingCalls = num;
if (num == 0)
{
  console.write("There are no incoming items.");
  environment.finish();
}
for (var i=0; i<num; i++)
{
  try
  {
    var result = virtualEarth.addPushpin(data["Combine"][i].Lat,
                                          data["Combine"][i].Lon,
                                          data["Combine"][i].Img,
                                          data["Combine"][i].Title,
                                          data["Combine"][i].Temp,
                                          true);
  }
  catch (ex)
  {
    environment.reportErrorInLoop(i, ex);
  }
}
```

Popfly PopQuiz

- Name three different services whose content are mashed in the WeatherMap mashups.
- Name the Popfly blocks used in WeatherMap1 and WeatherMap2 and their operations that create lists from original data that is not in a list.
- How can you combine information from more than ten blocks in WeatherMap 1?
- What is a limitation of the design of WeatherMap 1?
- Suppose you want to add ten more cities to your Weather Map. Which implementation will make it simpler to do so? Why?

Try It in Popfly

Modify or build these Popfly mashups to demonstrate your understanding of this lesson. The more ducks, the bigger the challenge.



1. In WeatherMap2, modify the User Input block's default values to contain zip codes for at least six of your favorite cities. Run the mashup.
2. Suppose you always want to display the same cities in WeatherMap 2. Modify that mashup to remove the User Input block. Where would you place the zip codes?



Popfly's Stock Quotes block contains operations to obtain a current stock quote and display a graph showing the performance of a given stock.

1. Following the format of the Weather Map 1 mashup, create your own mashup to display the performance of at least five of your favorite stocks. Use a separate Stock Quotes block for each stock and combine them. Display the graphs using an image display block such as PhotoStack.
2. Following the format of the Weather Map 2 mashup, create your own mashup to display the performance of at least five of your favorite stocks. Incorporate a User Input block for the user to enter stock symbols, split the input string at the spaces, and use a single Stock Quotes block in your solution.



Modify the Stocks mashup to display the stock symbol and current value as the title or caption of each graph image. Hint: Use the Text Appender block.



1. Create an original mashup that follows the format of Weather Map 2 by selecting another information provider and a display block.
2. Create a Popfly custom data block with zip codes of your favorite cities. Modify Professor Popfly's WeatherMap2 to use input from your data block rather than the User Input block.