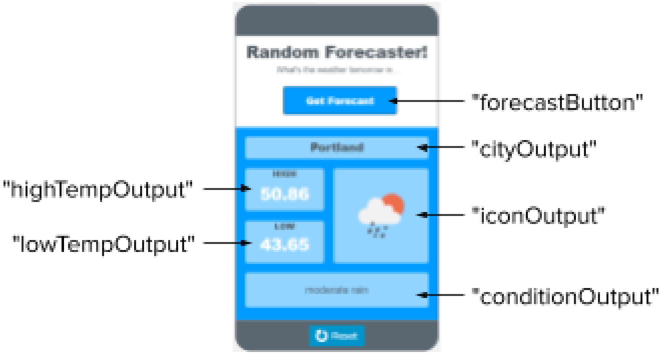
Name(s)____ Period ______ Date ___________________

Activity Guide - Traversals Make

Step 1 - Try the app

- Click on "Get Forecast"
- Click the button several times to see how the display changes



Discuss with a Partner

- What information is needed to create this app?
- What list filtering patterns might be used?

Step 2 - Plan

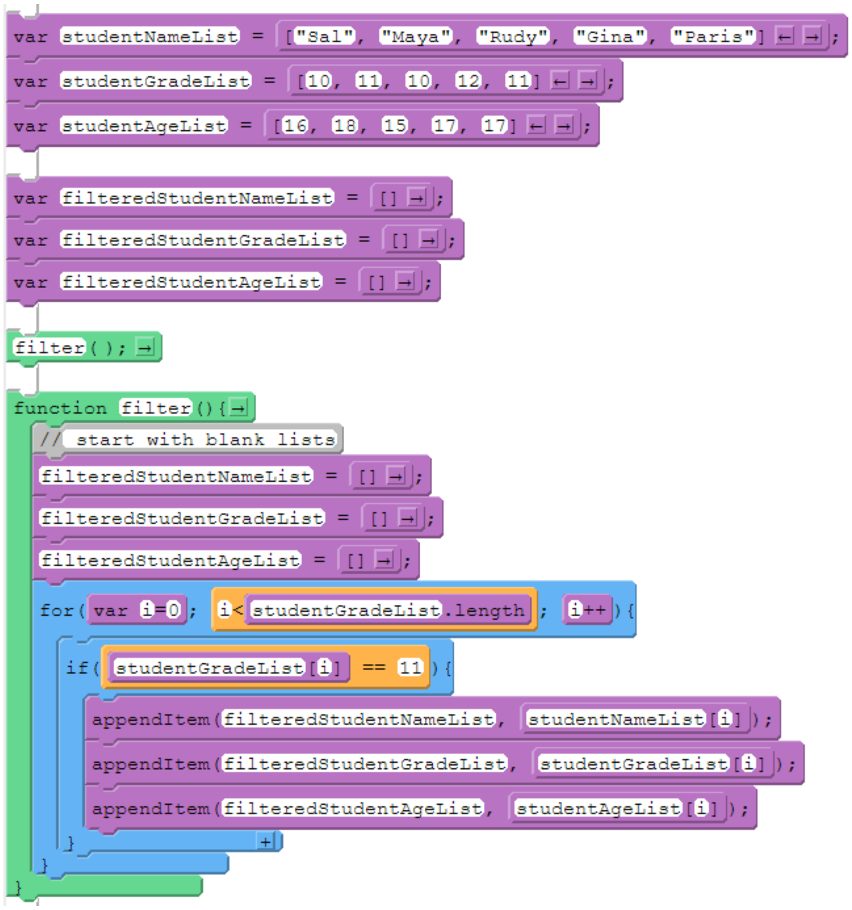
Lists: This app uses the **Daily Weather** table. Open the data tab and click on the table. Which columns do you think you will use in this app? The columns will be stored as lists in your program. There are **6 columns** used in this app.

Column	Name of List	What is stored
City	cityList	A list of cities
Forecast Number	forecastNumberList	The number of days from today, starting from 1 (for example: 1 = today, 2 = tomorrow)
	conditionList	

Filtered Lists: What lists will be filtered? What list will be used to filter these lists? Remember, we want to only display the forecast for tomorrow. There are **5 filtered lists** and one list which is traversed to filter the other lists.

Original List	Filtered List	Filtered by
cityList	filteredCityList	
conditionList	filteredConditionList	

Traversal: Review the **List Filter Pattern: Filtering Multiple Lists**. Consider how you will use this in your app.



```
var studentNameList = ["Sal", "Maya", "Rudy", "Gina", "Paris"];
var studentGradeList = [10, 11, 10, 12, 11];
var studentAgeList = [16, 18, 15, 17, 17];

var filteredStudentNameList = [];
var filteredStudentGradeList = [];
var filteredStudentAgeList = [];

filter();

function filter(){
  // start with blank lists
  filteredStudentNameList = [];
  filteredStudentGradeList = [];
  filteredStudentAgeList = [];
  for(var i=0; i<studentGradeList.length; i++){
    if(studentGradeList[i] == 11){
      appendItem(filteredStudentNameList, studentNameList[i]);
      appendItem(filteredStudentGradeList, studentGradeList[i]);
      appendItem(filteredStudentAgeList, studentAgeList[i]);
    }
  }
}
```

All the lists must be the same length for this pattern - ideally they are columns pulled from a table

How does it work?

- Create a variable for each list
- Create blank lists to store the filtered lists
- In a function, first reset all filtered lists to blank lists. Every time the function is called, the filtered lists will reset.
- Use a for loop to access each item in the list that contains the element you are filtering by
- If the element is found, append to each filtered list the element at the index in the original lists where the element was found

Output: Think about a function that updates the screen. This app displays a random city's forecast. How will the random city be selected? How will that information be used to display data from the filtered lists?

How will the random city be selected:

How will that information be used to display data from the filtered lists:

Step 3 - Write Your Code

- Write the code for the app, using your plan above and the comments provided in Code Studio to help
- Steps you can follow:
  - Create all the lists from your tables above.
  - Give your variables a starting value using the assignment operator (=). For lists that are created from columns in the table, use `getColumn()` to populate the lists.
  - Create a function to filter the lists. Refer to the List Filter Pattern.
  - Create a function that updates the screen.
  - Update the output elements on the screen inside of the function that updates the screen.
  - Create an `onEvent()` for the user input and call the function that updates the screen.
  - Use your debugging skills to identify unexpected behavior and fix your program
  - Comment your code as you go, explaining what each event handler and function does
- Extension Ideas
  - Add sound to indicate what the weather will be. For example, you can add a brief rain sound if it is rainy or a clip from a song talking about sunshine if it is going to be a sunny day.
  - Create an algorithm that would judge the weather in some way. For example, you might be able to tell a user if the weather was good weather for a specific activity such as going for a run outside.

Step 4 - Submit

Before you submit, check the rubric below to make sure your program meets the requirements of the task.

Category	Extensive Evidence	Convincing Evidence	Limited Evidence	No Evidence
Input	onEvents are created for all the required inputs.	onEvents are created for most of the inputs.	onEvents are created for some of the inputs.	onEvents are not created for any inputs.
Storage: Variables and Lists	Variables and lists are created and appropriately used for all pieces of information used in the app.	Variables and lists are created and appropriately used for most pieces of information used in the app.	Some information is stored in variables and lists and appropriately updated throughout the app.	There are no variables or lists which store the necessary information for the app to work correctly.
Processing: Lists	The program correctly processes the list for all user interface elements.	The program correctly processes the list for most user interface elements.	The program correctly processes the list for some of the user interface elements.	The program does not include or does not process a list.
Code: Functions	A function is used which correctly updates all output elements. The function is called in the appropriate onEvents.	A function is used which correctly updates most of the output elements. The function is called in the appropriate onEvents.	A function is used which updates some of the output elements or the function is only called in some of the appropriate onEvents.	There is no function which updates the screen.
Output	The screen correctly displays a random city's forecast.	The screen displays most of a random city's forecast.	The screen displays some of a random city's forecast.	The screen does not correctly display any of a random city's forecast.
Code runs without errors	No errors are present in the required code.	One or two errors are present in the required code.	Three or four errors are present in the required code.	More than four errors are present in the required code.
Coding Comments	Comments are used to correctly explain the purpose and function of all onEvents and functions.	Comments are used to explain the purpose and function of most onEvents and functions.	Comments are used to explain the purpose and function of some onEvents and functions.	Comments are not present.