

Unit 5 Lesson 1

Lists Explore

Resources

Unit 5 Lesson 2

Lists Investigate

Resources

Unit 5 Lesson 3

Lists Practice

Resources

Unit 5 Lesson 4

Lists Make

Resources

Name(s) _____ Period _____ Date _____

Activity Guide - Lists Make

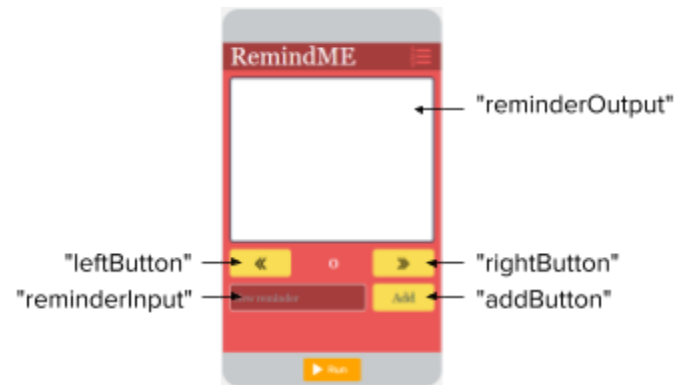


Step 1 - Try the app

- Add several reminders to the app.
- Scroll through your reminders.

Discuss with a Partner

- What programming patterns with lists do you think you'll need to use?



Step 2 - Plan

Lists: This app uses one list. Give it a name! What does it store?

List Name	What the List Stores

Other Variable: This app uses one extra variable in addition to the list. What does it store?

Variable Name	What the Variable Stores
<i>index</i>	

Conditionals: Review the List Scrolling Pattern. Consider how you will use this in your app.

```

var index = 0;
var myList = ["banana", "apple", "pear"];

onEvent(▼ "leftButton", ▼ "click", function(){
  if(index > 0){
    index--;
  }
  updateScreen();
});

onEvent(▼ "rightButton", ▼ "click", function(){
  if(index < myList.length - 1){
    index++;
  }
  updateScreen();
});

function updateScreen(){
  setProperty(▼ "fruitLabel", ▼ "text", myList[index]);
}
  
```

How does it work?

Many App Lab projects run in the following way.

1. The user is interested in seeing items in a list, like a list of favorite foods, or a list of reminders.
2. The app shows the user one item in the list at a time.
3. By clicking buttons (or otherwise interacting with the app) the user can move back and forth through the list, one item at a time, to see every item.

This pattern allows a user to "scroll" through all the items in the list. In order to make it work, create an index variable and a list. Then use the "Counter Pattern with Boundary" pattern to create event handlers to change the value of the index.

This pattern makes sense to use with the `updateScreen` pattern since you will need at least two buttons for scrolling left and right through the list but afterwards will use the same code to update the screen.

Function: What code may be repeated in your app? What should be included in your function(s)?

- Hint: Check out how the `updateScreen()` function is set up in the List Scrolling Pattern.

Inputs: What are the inputs for the app? These will all be turned into `onEvents`.

Input	Action	Result
"rightButton"	click	The reminder stored in the list at the next index is displayed.

Step 3 - Write Your Code

- Write the code for the app, using your plan above and the comments provided in Code Studio to help
- Step You Can Follow
 - Create all the variables (including the list!) from your tables above.
 - Give your variables a starting value using the assignment operator (`=`).
 - Create a function that updates the screen.
 - Update the output elements on the screen inside of the function.
 - Create event handlers (`onEvent`) for the inputs in your table above
 - Inside the event handlers, update the appropriate variable using the List Scrolling Pattern. Then call the update screen function.
 - 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
 - Create a button that deletes unwanted items from the list.
 - Change the list scrolling so that it wraps from the end of the list back to the beginning.

Step 4 - Submit

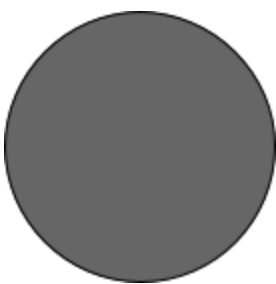
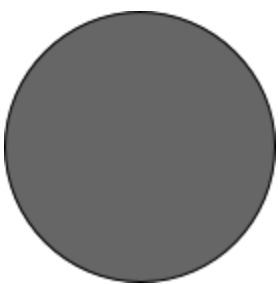
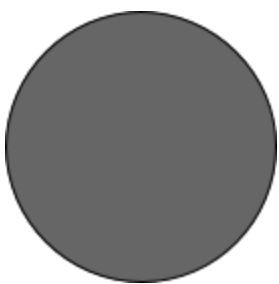
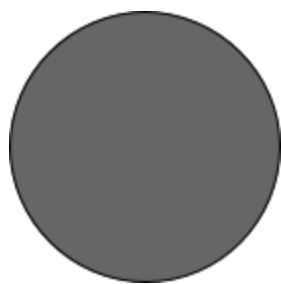
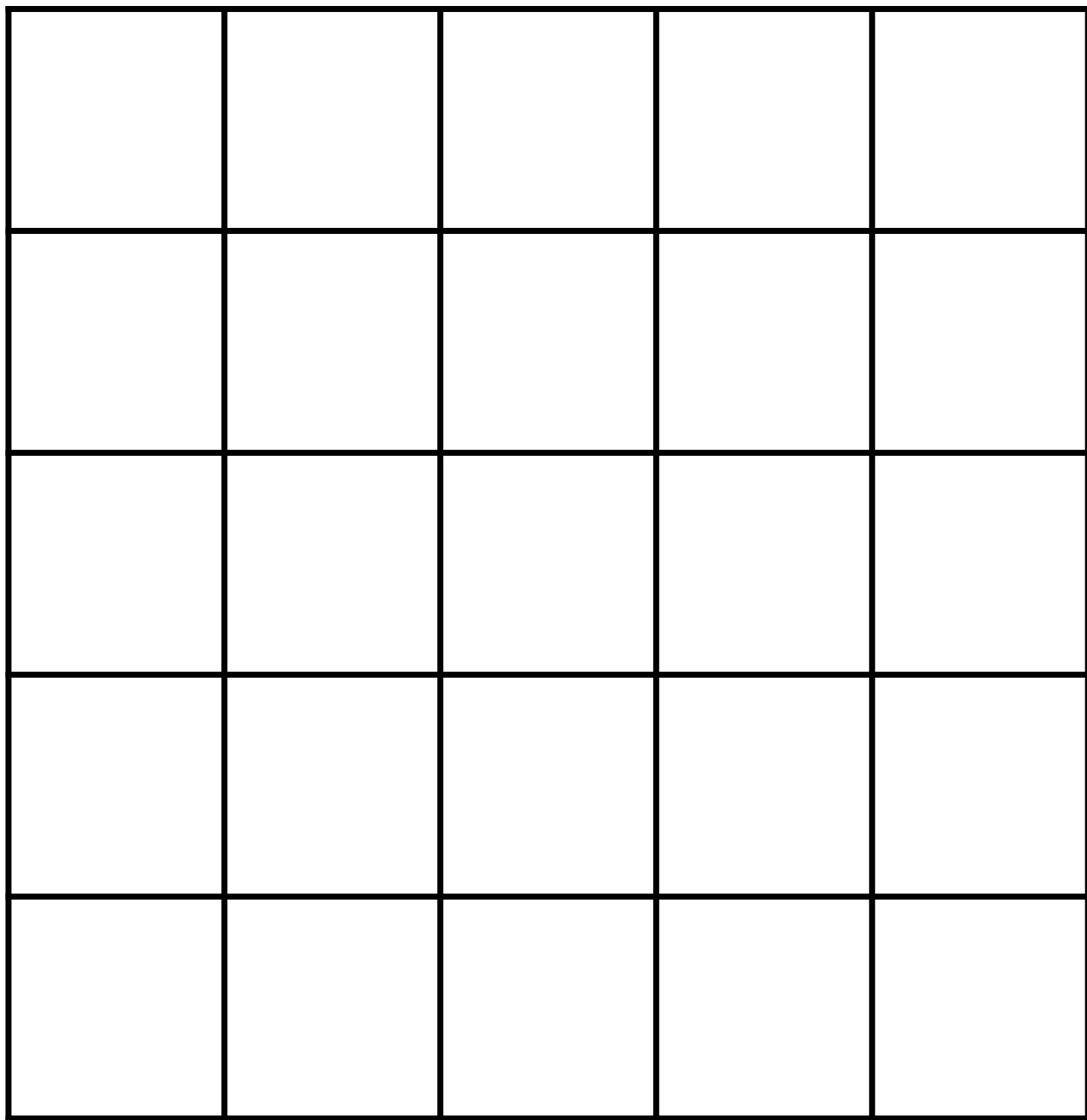
Before you submit check the rubric below to make sure your program meets the requirements of the task.

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 the current reminders in all instances.	The screen correctly displays the current reminders in most instances.	The screen correctly displays some but not all information.	The screen does not correctly display any stored information.
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.

Unit 5 Lesson 5

Loops Explore

Resources



Unit 5 Lesson 6

Loops Investigate

Resources

Unit 5 Lesson 7

Loops Practice

Resources

Unit 5 Lesson 8

Loops Make

Resources

Name(s) _____ Period _____ Date _____

Activity Guide - Loops Make



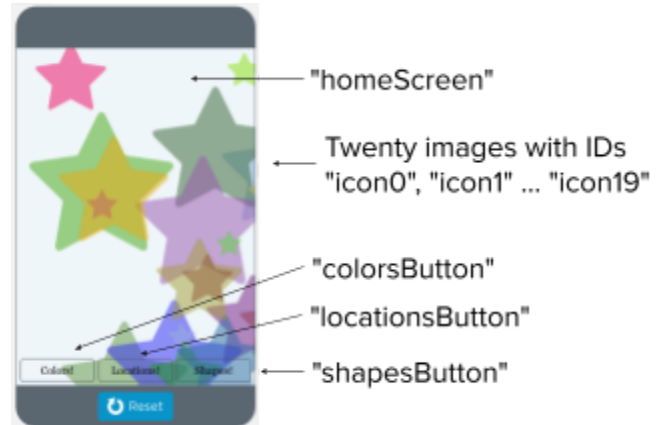
Step 1 - Try the app

Try using the different combinations of inputs.

- What does each button change and what does it not change?

Discuss with a Partner

- Where (if at all) do you think this app is using a list?
- Where (if at all) do you think this app is using a loop?



Step 2 - Plan

Fill in the information in the table below for each event handler you'll need to create

Element ID	Description of What the Event Handler will Do

Fill in the table below for each variable you'll need to create.

Variable Name	What the Variable Stores

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 variables from your table above.
 - Give your variables a starting value using the assignment operator (=).
 - Create blank event handlers (onEvent) for each screen element in your table above.
 - Write the code to make each of the three buttons work.
 - Hint: in order to create random colors use the rgba() block with randomNumber(). Read the documentation for those blocks if you need more help.
 - Use your debugging skills to identify unexpected behavior and fix your program.
 - Comment your code as you go, explaining what each event handler does.
- Extension Ideas
 - Create a way to hide the three buttons when you have a lock screen that you like
 - Add sounds to each button.
 - Change the code for the "shapes" button to guarantee that you never randomly get the same shape twice in a row.

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.
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.
Loops	The program correctly uses loops for all three buttons to generate the expected output.	The program correctly uses a loop for two of the buttons to generate the expected output.	The program correctly uses lists for one of the buttons to generate the expected output.	The program does not include or correctly use any loops.
Output	All three buttons work as expected to move icons, change their colors, and change their icon to a random new icon.	Only two of the buttons work as expected.	Only one of the buttons work as expected.	None of the buttons work as expected.
Code runs without errors.	No errors are present in the required code.	Some errors are present in the required code.	Many errors are present in the required code.	The code does not run.
Comments	Comments are used to correctly explain the purpose and functionality of both the function and event handlers.	Comments are used to explain the purpose and functionality of either the function or event handlers.	Comments are present, but are not used to explain the purpose or functionality of the function or event handlers.	Comments are not present.

Unit 5 Lesson 9

Traversals Explore

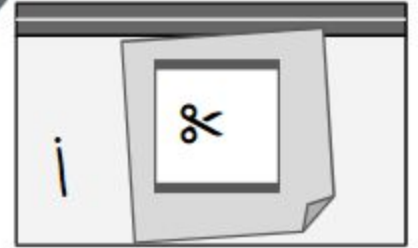
Resources

&

```
for ( var i= 0 ; i<list.length; i++ ) {
```

```
    var element= list[ & ];
```

```
}
```



%

	0	1	2	3	4	5	6	7
--	---	---	---	---	---	---	---	---

PULL ▲

%

i	0	1	2	3	4	5	6	7
---	---	---	---	---	---	---	---	---

Unit 5 Lesson 10

Traversals Investigate

Resources

Unit 5 Lesson 11

Traversals Practice

Resources

Unit 5 Lesson 12

Traversals Make

Resources

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	<i>cityList</i>	<i>A list of cities</i>
Forecast Number	<i>forecastNumberList</i>	<i>The number of days from today (for example: 1 = tomorrow)</i>
	<i>conditionList</i>	

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
<i>cityList</i>	<i>filteredCityList</i>	
<i>conditionList</i>	<i>filteredConditionList</i>	

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, 19, 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.

Unit 5 Lesson 13

Project - Hackathon Part 1

Resources

Hackathon Project Planning Guide

Project Description

For this project you will work with a partner. Together you will create an app that uses a dataset. There are two roles in this project: designer and programmer. You are both responsible for the overall project, but you will focus on different parts of the project at any given time. On the last day of the project, you will individually complete a written response that will help you practice for the Create Performance Task.

You will submit

- Your final app
- This completed project-planning guide
- A written response

App Requirements

- At least three screens
 - All screens can be easily navigated to through the user interface
- A dataset used in a meaningful way towards the program's purpose
- At least one list is traversed using: map, reduce, or filter (indicate which in a comment) in a meaningful way towards the program's purpose
- Programming constructs: variable(s), function(s), conditional(s), list(s), loop(s)
- All functions include comments that explain their purpose and how they work
- All element IDs have meaningful names
- No errors in the code

Steps

- Collaborate with your partner to pick a dataset
- Decide on an app to build which traverses a list from the dataset
- Create a paper prototype
- **Designer:**
 - Create the screens
 - Support the programmer with pair programming as needed
- **Programmer:**
 - Program the app
 - Support the designer as needed.

Investigate Phase

Step 1. Choose a Dataset: Open a project on Code Studio and look through the different datasets available. Choose one that looks interesting to both you and your partner.

Dataset: _____.

What column(s) of data will you use in your app?

-
-
-

Step 2. Brainstorm an App: Consider the columns of data that you are using. How will your app traverse this data? Circle one and explain below.

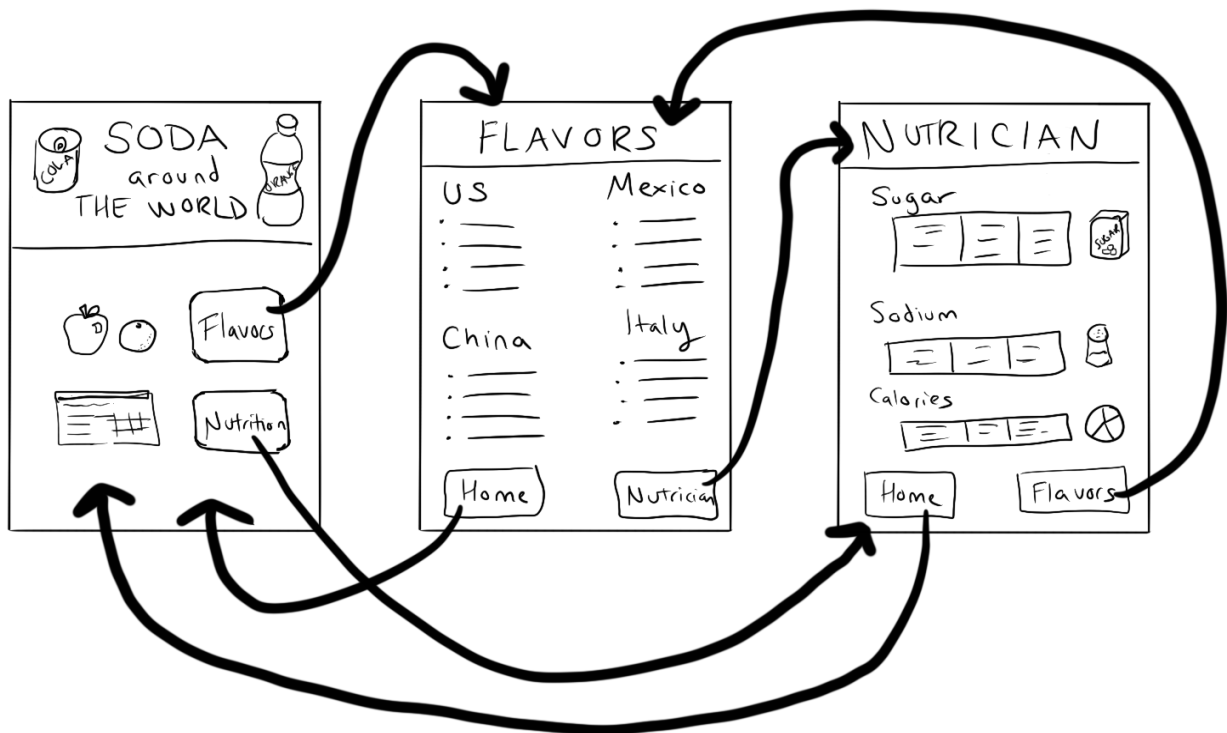
- **Filter** (most common option): use the list from one column to determine information that will be filtered from a list created by another column
 - Example: dogHeight filters dogNames, so only the names of small dogs are added to the filtered list
- **Map:** Add or change each item in a list
 - Example: map a list of numbers pulled from a column using Math.round - now each number is rounded
- **Reduce:** Reduce the data in a list to a single number
 - Example: find the smallest number in a list

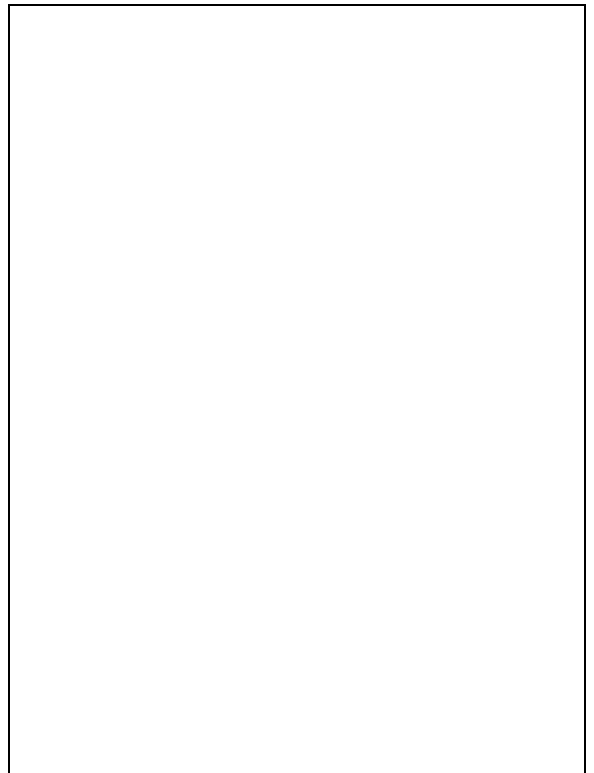
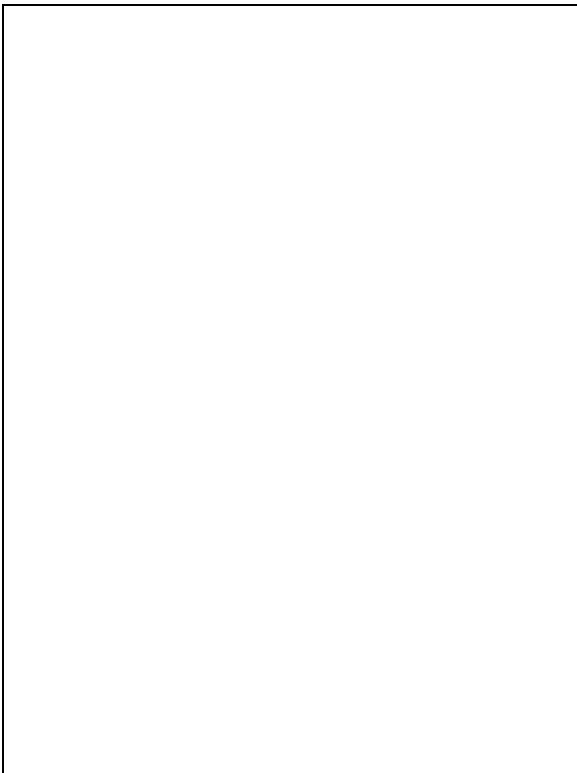
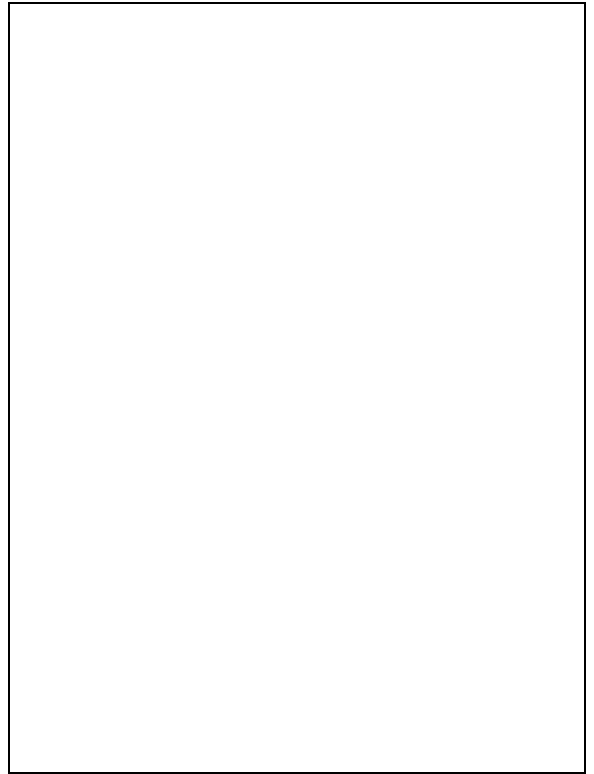
Your app will use (circle): MAP REDUCE FILTER

Explain in more detail:

Design Phase

Step 3. Create a Paper Prototype: On the next page you should draw a **prototype** that shows how your app will actually run. This means you should include all the buttons, text, and images that the user will be able to use and see. Write notes or draw arrows showing how different user interface elements should work. For example, if clicking a button takes me to another screen, I should draw an arrow from that button to the drawing of the screen.





Step 4. Prepare to Build Your App:

Before you begin to code, fill out the chart below. Add more rows (or use a scrap piece of paper) as needed.

Element IDs		
Name	Type	Notes
<i>nameOutput</i>	<i>label</i>	<i>Displays the name of the dog on the screen</i>

Variable(s)		
Name	Type	What it's storing
<i>dogSize</i>	<i>list</i>	<i>A list of dog sizes pulled from the "Max Height" column</i>
Conditional(s)		
Boolean Expression (pseudocode is ok)		Notes
<i>if dog size is greater than 15, append to a list</i>		<i>Inside a for loop that traverses the dogSize list</i>
Loop(s)		
For Loop (pseudocode is ok)		Notes
<i>for(var i=0; i<dogSize.length; i++)</i>		<i>Traverses dogSize list</i>

onEvent(s)		
Element ID	Action	What happens?
"dogButton"	"click"	A picture of a dog appears The background of the screen changes to green
function(s)		
Name	Purpose	How it works
updateScreen	updates what appears on the screen after the user selects a dog	Filters the lists and displays the images and names of dogs on the screen.

Build Phase

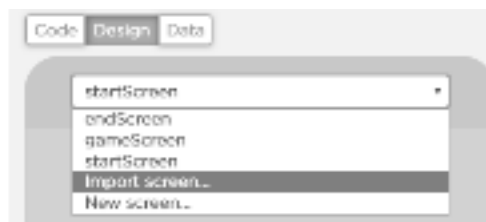
Step 5: Test Screens: Build a quick version of the screens which includes all elements with their proper element IDs. Do not worry about design at all. This is purely to allow the programmer the ability to test their code as they go along.

Step 6: Build

Programmers: Use the chart to guide you in adding programming statements to your program.

Designers: Use the chart to guide you in adding screen elements to your program. You can work on a separate computer from your partner.

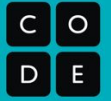
- When you have finished screens, your partner should delete the test screens and then import the finished screens.
- Note: If screens are not deleted before importing the project you will get an error message because element IDs will be the same. To avoid this, add a blank screen, delete the old project screens, import the finished screens, and delete the blank screen. Make sure to set your home screen to be the default.



Scoring Guidelines

Written Response (individual) - In the Written Response the student...	
Task	Point
Describes the purpose of the app	/1
Includes a program segment that shows how the initial data has been stored in a list	/1
Includes a program segment that shows how the data in this list is being processed	/1
Identifies the name of the list being processed in this response	/1
Identifies what the data contained in this list is representing in the program	/1
Explains how the list manages complexity by explaining how this program code would be written differently without using this list.	/1
Overall Project (both partners)	
Program uses a dataset in a meaningful way towards the program's purpose	/1
At least one list is traversed using: map, reduce, or filter (indicate which in comment) in a meaningful way towards the program's purpose	/1
Program contains the following programming constructs that are used in meaningful ways towards the program's purpose	
• variable(s)	/1
• function(s)	/1
• conditional(s)	/1
• list(s)	/1
• loop(s)	/1
The app has at least three different screens	
• screen #1	/1
• screen #2	/1
• screen #3	/1
All screens can be easily navigated to through the user interface	/1
All functions include comments that explain purpose and how they work.	/1
No errors in the code	/1
All element IDs have meaningful names	/1
Total	/20

Hackathon Project Written Response



Question 1: Provide a written response that describes the overall purpose of the program

Write your response here

Question 2: Insert screenshots of two program code segments which contain a list being used in your program.

The first program code segment should show how the data has been stored in the list.

Insert the screenshot of the first program code here

The second program code segment should show the data in the same list being used, such as creating new data from the existing data or accessing multiple items in the list.

Insert the screenshot of the second program code here

Write a written response that:

- Identifies the name of the list being processed
- Identifies what the data contained in the list is representing in your program
- Explains how the list manages complexity in your program by explaining how your program code would be written differently if you did not use the list

Name of the list	
Data contained in the list	
The list manages complexity by...	

Unit 5 Lesson 14

Project - Hackathon Part 2

Resources

Hackathon Project Planning Guide

Project Description

For this project you will work with a partner. Together you will create an app that uses a dataset. There are two roles in this project: designer and programmer. You are both responsible for the overall project, but you will focus on different parts of the project at any given time. On the last day of the project, you will individually complete a written response that will help you practice for the Create Performance Task.

You will submit

- Your final app
- This completed project-planning guide
- A written response

App Requirements

- At least three screens
 - All screens can be easily navigated to through the user interface
- A dataset used in a meaningful way towards the program's purpose
- At least one list is traversed using: map, reduce, or filter (indicate which in a comment) in a meaningful way towards the program's purpose
- Programming constructs: variable(s), function(s), conditional(s), list(s), loop(s)
- All functions include comments that explain their purpose and how they work
- All element IDs have meaningful names
- No errors in the code

Steps

- Collaborate with your partner to pick a dataset
- Decide on an app to build which traverses a list from the dataset
- Create a paper prototype
- **Designer:**
 - Create the screens
 - Support the programmer with pair programming as needed
- **Programmer:**
 - Program the app
 - Support the designer as needed.

Investigate Phase

Step 1. Choose a Dataset: Open a project on Code Studio and look through the different datasets available. Choose one that looks interesting to both you and your partner.

Dataset: _____.

What column(s) of data will you use in your app?

-
-
-

Step 2. Brainstorm an App: Consider the columns of data that you are using. How will your app traverse this data? Circle one and explain below.

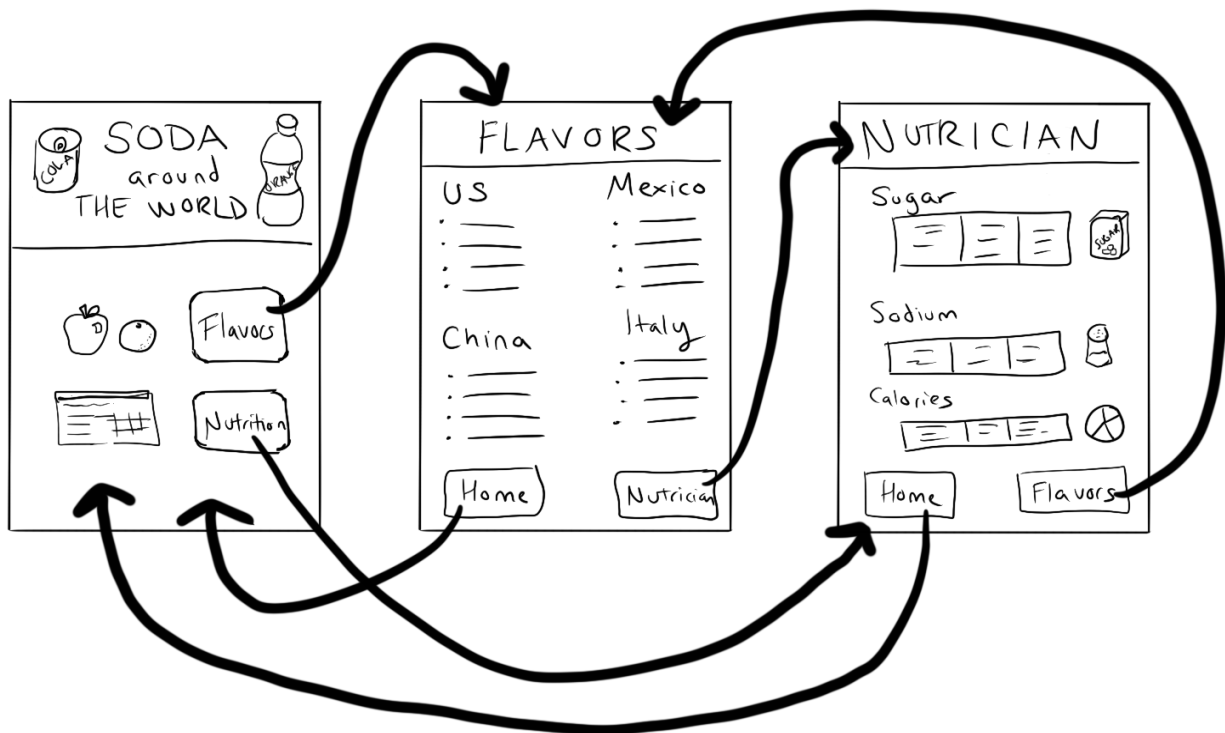
- **Filter** (most common option): use the list from one column to determine information that will be filtered from a list created by another column
 - Example: dogHeight filters dogNames, so only the names of small dogs are added to the filtered list
- **Map:** Add or change each item in a list
 - Example: map a list of numbers pulled from a column using Math.round - now each number is rounded
- **Reduce:** Reduce the data in a list to a single number
 - Example: find the smallest number in a list

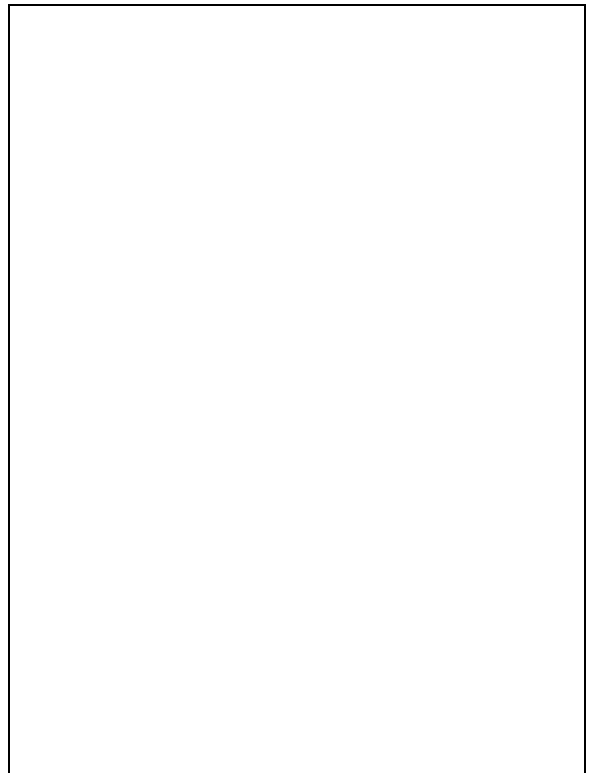
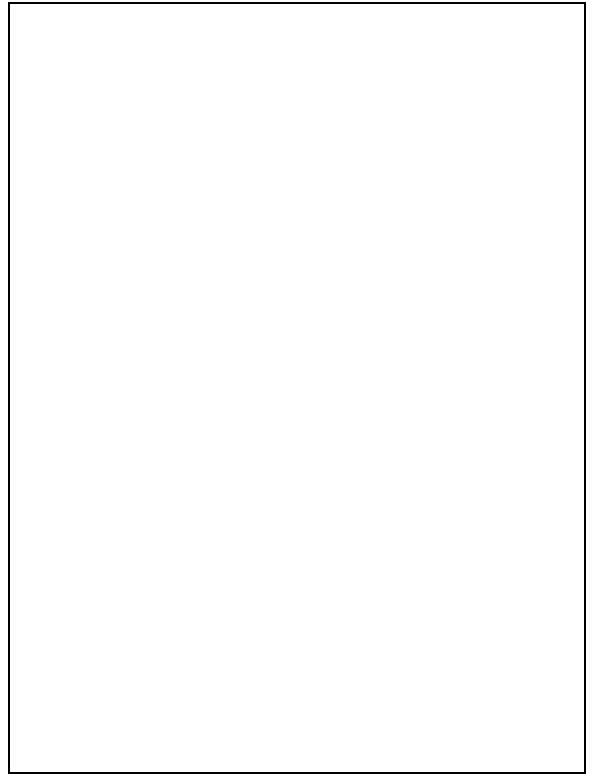
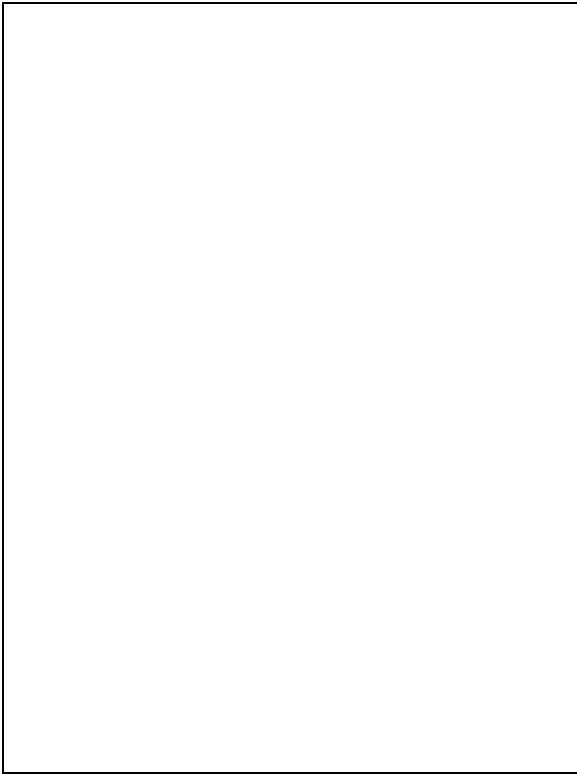
Your app will use (circle): MAP REDUCE FILTER

Explain in more detail:

Design Phase

Step 3. Create a Paper Prototype: On the next page you should draw a **prototype** that shows how your app will actually run. This means you should include all the buttons, text, and images that the user will be able to use and see. Write notes or draw arrows showing how different user interface elements should work. For example, if clicking a button takes me to another screen, I should draw an arrow from that button to the drawing of the screen.





Step 4. Prepare to Build Your App:

Before you begin to code, fill out the chart below. Add more rows (or use a scrap piece of paper) as needed.

Element IDs		
Name	Type	Notes
<i>nameOutput</i>	<i>label</i>	<i>Displays the name of the dog on the screen</i>

Variable(s)		
Name	Type	What it's storing
<i>dogSize</i>	<i>list</i>	<i>A list of dog sizes pulled from the "Max Height" column</i>
Conditional(s)		
Boolean Expression (pseudocode is ok)		Notes
<i>if dog size is greater than 15, append to a list</i>		<i>Inside a for loop that traverses the dogSize list</i>
Loop(s)		
For Loop (pseudocode is ok)		Notes
<i>for(var i=0; i<dogSize.length; i++)</i>		<i>Traverses dogSize list</i>

onEvent(s)		
Element ID	Action	What happens?
"dogButton"	"click"	A picture of a dog appears The background of the screen changes to green
function(s)		
Name	Purpose	How it works
updateScreen	updates what appears on the screen after the user selects a dog	Filters the lists and displays the images and names of dogs on the screen.

Build Phase

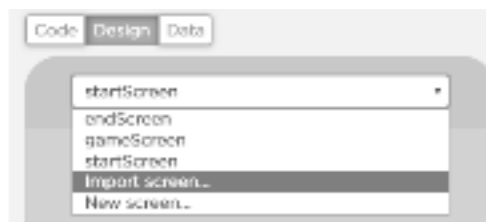
Step 5: Test Screens: Build a quick version of the screens which includes all elements with their proper element IDs. Do not worry about design at all. This is purely to allow the programmer the ability to test their code as they go along.

Step 6: Build

Programmers: Use the chart to guide you in adding programming statements to your program.

Designers: Use the chart to guide you in adding screen elements to your program. You can work on a separate computer from your partner.

- When you have finished screens, your partner should delete the test screens and then import the finished screens.
- Note: If screens are not deleted before importing the project you will get an error message because element IDs will be the same. To avoid this, add a blank screen, delete the old project screens, import the finished screens, and delete the blank screen. Make sure to set your home screen to be the default.



Scoring Guidelines

Written Response (individual) - In the Written Response the student...	
Task	Point
Describes the purpose of the app	/1
Includes a program segment that shows how the initial data has been stored in a list	/1
Includes a program segment that shows how the data in this list is being processed	/1
Identifies the name of the list being processed in this response	/1
Identifies what the data contained in this list is representing in the program	/1
Explains how the list manages complexity by explaining how this program code would be written differently without using this list.	/1
Overall Project (both partners)	
Program uses a dataset in a meaningful way towards the program's purpose	/1
At least one list is traversed using: map, reduce, or filter (indicate which in comment) in a meaningful way towards the program's purpose	/1
Program contains the following programming constructs that are used in meaningful ways towards the program's purpose	
• variable(s)	/1
• function(s)	/1
• conditional(s)	/1
• list(s)	/1
• loop(s)	/1
The app has at least three different screens	
• screen #1	/1
• screen #2	/1
• screen #3	/1
All screens can be easily navigated to through the user interface	/1
All functions include comments that explain purpose and how they work.	/1
No errors in the code	/1
All element IDs have meaningful names	/1
Total	/20

Unit 5 Lesson 15

Project - Hackathon Part 3

Resources

Hackathon Project Planning Guide

Project Description

For this project you will work with a partner. Together you will create an app that uses a dataset. There are two roles in this project: designer and programmer. You are both responsible for the overall project, but you will focus on different parts of the project at any given time. On the last day of the project, you will individually complete a written response that will help you practice for the Create Performance Task.

You will submit

- Your final app
- This completed project-planning guide
- A written response

App Requirements

- At least three screens
 - All screens can be easily navigated to through the user interface
- A dataset used in a meaningful way towards the program's purpose
- At least one list is traversed using: map, reduce, or filter (indicate which in a comment) in a meaningful way towards the program's purpose
- Programming constructs: variable(s), function(s), conditional(s), list(s), loop(s)
- All functions include comments that explain their purpose and how they work
- All element IDs have meaningful names
- No errors in the code

Steps

- Collaborate with your partner to pick a dataset
- Decide on an app to build which traverses a list from the dataset
- Create a paper prototype
- **Designer:**
 - Create the screens
 - Support the programmer with pair programming as needed
- **Programmer:**
 - Program the app
 - Support the designer as needed.

Investigate Phase

Step 1. Choose a Dataset: Open a project on Code Studio and look through the different datasets available. Choose one that looks interesting to both you and your partner.

Dataset: _____.

What column(s) of data will you use in your app?

-
-
-

Step 2. Brainstorm an App: Consider the columns of data that you are using. How will your app traverse this data? Circle one and explain below.

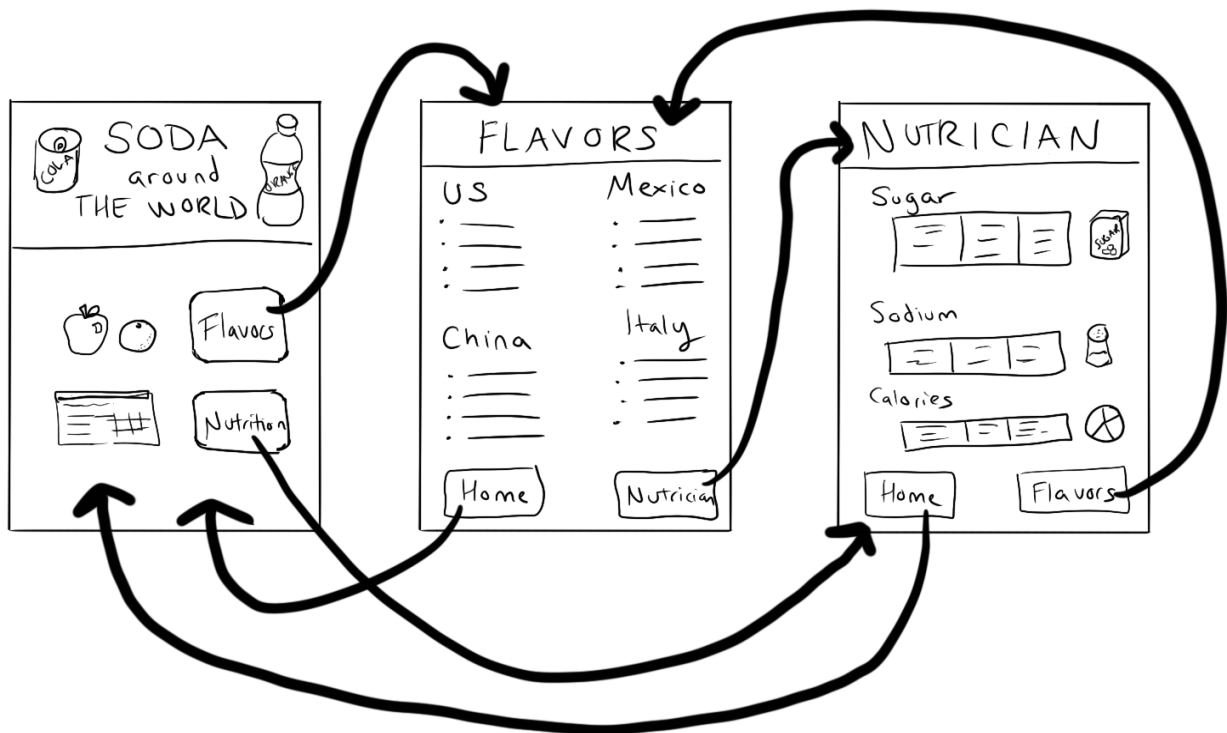
- **Filter** (most common option): use the list from one column to determine information that will be filtered from a list created by another column
 - Example: dogHeight filters dogNames, so only the names of small dogs are added to the filtered list
- **Map:** Add or change each item in a list
 - Example: map a list of numbers pulled from a column using Math.round - now each number is rounded
- **Reduce:** Reduce the data in a list to a single number
 - Example: find the smallest number in a list

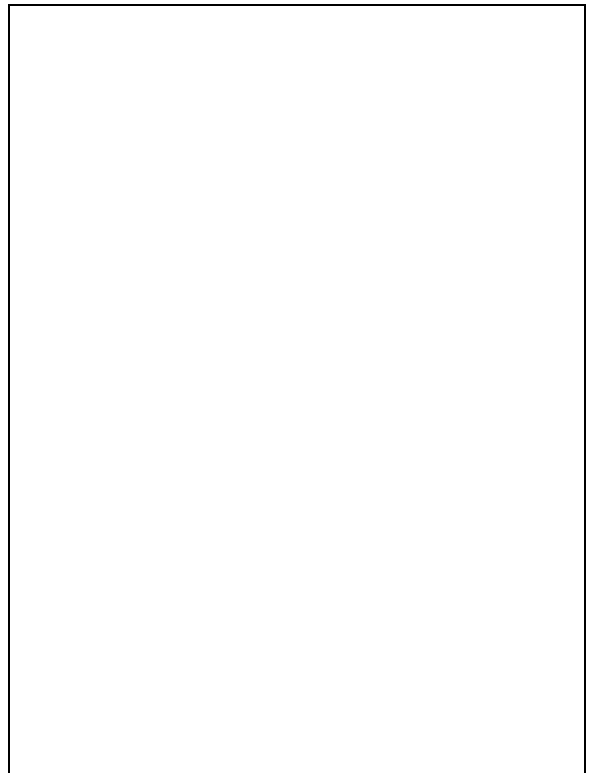
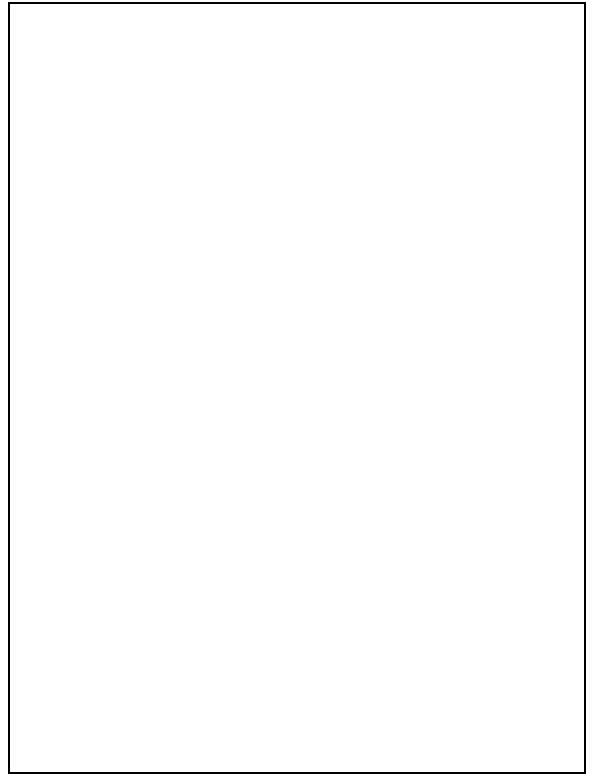
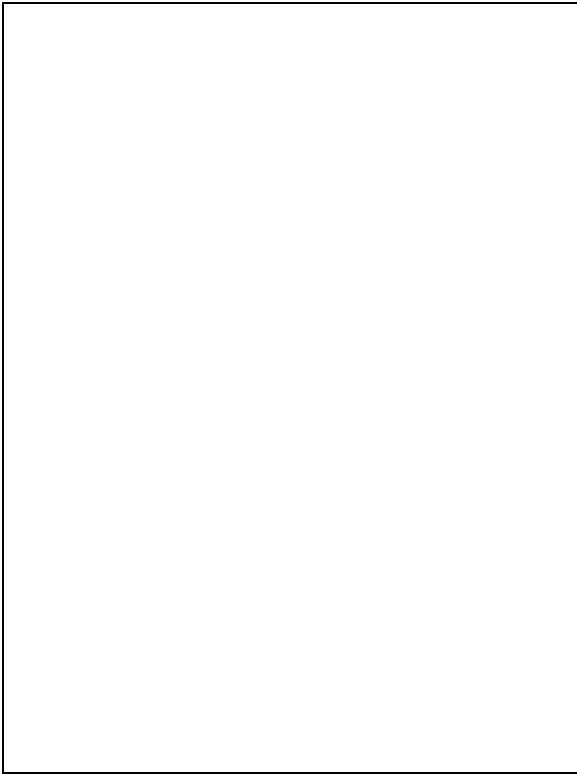
Your app will use (circle): MAP REDUCE FILTER

Explain in more detail:

Design Phase

Step 3. Create a Paper Prototype: On the next page you should draw a **prototype** that shows how your app will actually run. This means you should include all the buttons, text, and images that the user will be able to use and see. Write notes or draw arrows showing how different user interface elements should work. For example, if clicking a button takes me to another screen, I should draw an arrow from that button to the drawing of the screen.





Step 4. Prepare to Build Your App:

Before you begin to code, fill out the chart below. Add more rows (or use a scrap piece of paper) as needed.

Element IDs		
Name	Type	Notes
<i>nameOutput</i>	<i>label</i>	<i>Displays the name of the dog on the screen</i>

Variable(s)		
Name	Type	What it's storing
<i>dogSize</i>	<i>list</i>	<i>A list of dog sizes pulled from the "Max Height" column</i>
Conditional(s)		
Boolean Expression (pseudocode is ok)		Notes
<i>if dog size is greater than 15, append to a list</i>		<i>Inside a for loop that traverses the dogSize list</i>
Loop(s)		
For Loop (pseudocode is ok)		Notes
<i>for(var i=0; i<dogSize.length; i++)</i>		<i>Traverses dogSize list</i>

onEvent(s)		
Element ID	Action	What happens?
"dogButton"	"click"	A picture of a dog appears The background of the screen changes to green
function(s)		
Name	Purpose	How it works
updateScreen	updates what appears on the screen after the user selects a dog	Filters the lists and displays the images and names of dogs on the screen.

Build Phase

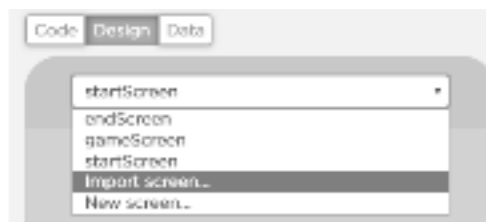
Step 5: Test Screens: Build a quick version of the screens which includes all elements with their proper element IDs. Do not worry about design at all. This is purely to allow the programmer the ability to test their code as they go along.

Step 6: Build

Programmers: Use the chart to guide you in adding programming statements to your program.

Designers: Use the chart to guide you in adding screen elements to your program. You can work on a separate computer from your partner.

- When you have finished screens, your partner should delete the test screens and then import the finished screens.
- Note: If screens are not deleted before importing the project you will get an error message because element IDs will be the same. To avoid this, add a blank screen, delete the old project screens, import the finished screens, and delete the blank screen. Make sure to set your home screen to be the default.



Scoring Guidelines

Written Response (individual) - In the Written Response the student...	
Task	Point
Describes the purpose of the app	/1
Includes a program segment that shows how the initial data has been stored in a list	/1
Includes a program segment that shows how the data in this list is being processed	/1
Identifies the name of the list being processed in this response	/1
Identifies what the data contained in this list is representing in the program	/1
Explains how the list manages complexity by explaining how this program code would be written differently without using this list.	/1
Overall Project (both partners)	
Program uses a dataset in a meaningful way towards the program's purpose	/1
At least one list is traversed using: map, reduce, or filter (indicate which in comment) in a meaningful way towards the program's purpose	/1
Program contains the following programming constructs that are used in meaningful ways towards the program's purpose	
• variable(s)	/1
• function(s)	/1
• conditional(s)	/1
• list(s)	/1
• loop(s)	/1
The app has at least three different screens	
• screen #1	/1
• screen #2	/1
• screen #3	/1
All screens can be easily navigated to through the user interface	/1
All functions include comments that explain purpose and how they work.	/1
No errors in the code	/1
All element IDs have meaningful names	/1
Total	/20

Unit 5 Lesson 16

Project - Hackathon Part 4

Resources

Hackathon Project Planning Guide

Project Description

For this project you will work with a partner. Together you will create an app that uses a dataset. There are two roles in this project: designer and programmer. You are both responsible for the overall project, but you will focus on different parts of the project at any given time. On the last day of the project, you will individually complete a written response that will help you practice for the Create Performance Task.

You will submit

- Your final app
- This completed project-planning guide
- A written response

App Requirements

- At least three screens
 - All screens can be easily navigated to through the user interface
- A dataset used in a meaningful way towards the program's purpose
- At least one list is traversed using: map, reduce, or filter (indicate which in a comment) in a meaningful way towards the program's purpose
- Programming constructs: variable(s), function(s), conditional(s), list(s), loop(s)
- All functions include comments that explain their purpose and how they work
- All element IDs have meaningful names
- No errors in the code

Steps

- Collaborate with your partner to pick a dataset
- Decide on an app to build which traverses a list from the dataset
- Create a paper prototype
- **Designer:**
 - Create the screens
 - Support the programmer with pair programming as needed
- **Programmer:**
 - Program the app
 - Support the designer as needed.

Investigate Phase

Step 1. Choose a Dataset: Open a project on Code Studio and look through the different datasets available. Choose one that looks interesting to both you and your partner.

Dataset: _____.

What column(s) of data will you use in your app?

-
-
-

Step 2. Brainstorm an App: Consider the columns of data that you are using. How will your app traverse this data? Circle one and explain below.

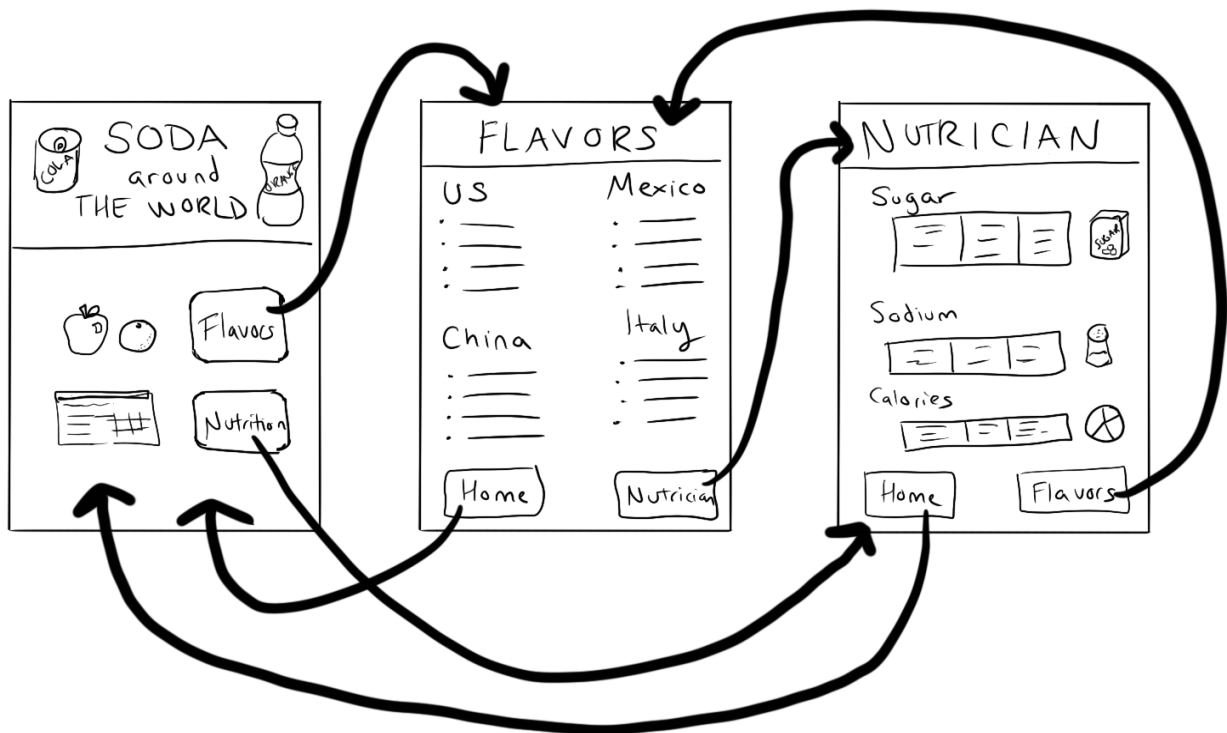
- **Filter** (most common option): use the list from one column to determine information that will be filtered from a list created by another column
 - Example: dogHeight filters dogNames, so only the names of small dogs are added to the filtered list
- **Map:** Add or change each item in a list
 - Example: map a list of numbers pulled from a column using Math.round - now each number is rounded
- **Reduce:** Reduce the data in a list to a single number
 - Example: find the smallest number in a list

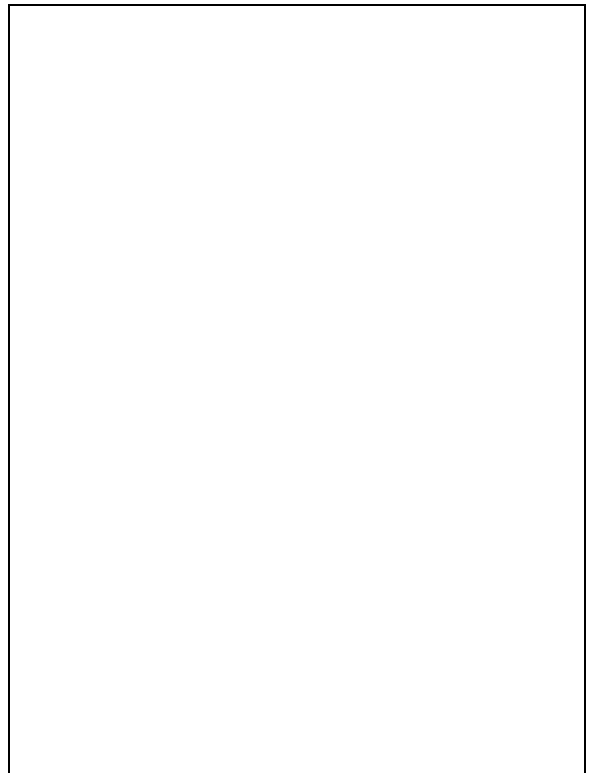
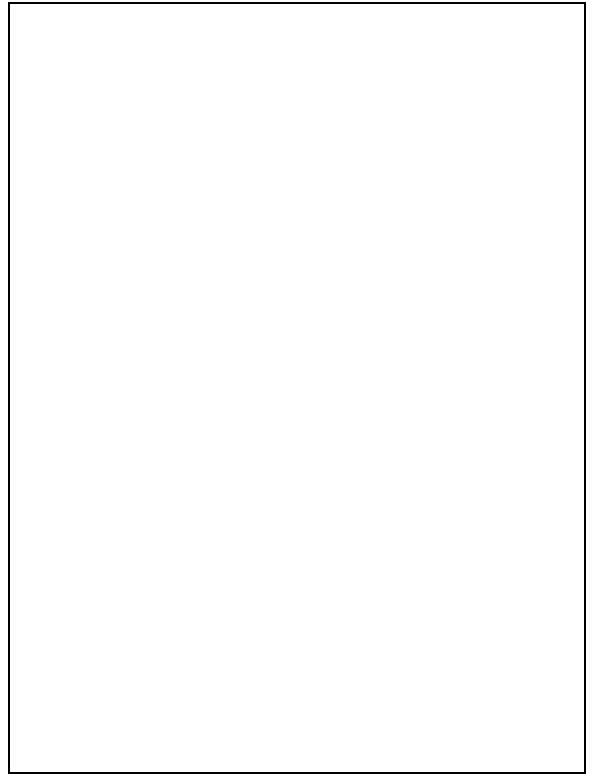
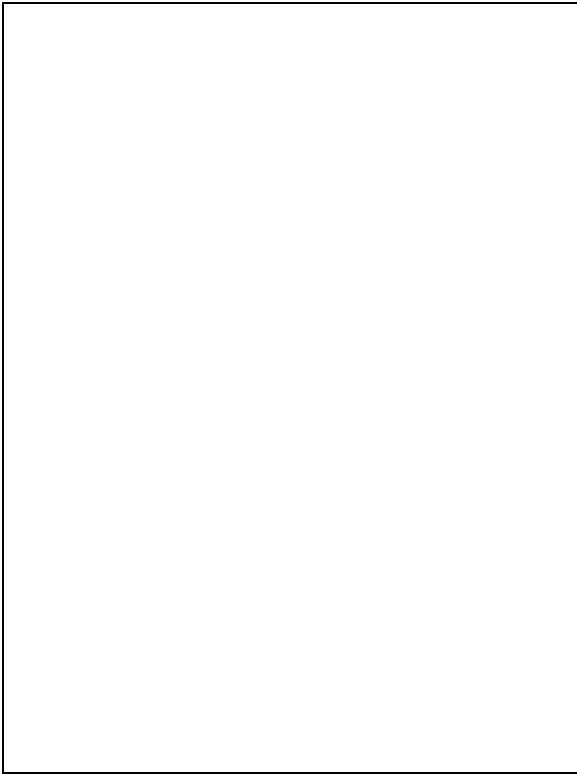
Your app will use (circle): MAP REDUCE FILTER

Explain in more detail:

Design Phase

Step 3. Create a Paper Prototype: On the next page you should draw a **prototype** that shows how your app will actually run. This means you should include all the buttons, text, and images that the user will be able to use and see. Write notes or draw arrows showing how different user interface elements should work. For example, if clicking a button takes me to another screen, I should draw an arrow from that button to the drawing of the screen.





Step 4. Prepare to Build Your App:

Before you begin to code, fill out the chart below. Add more rows (or use a scrap piece of paper) as needed.

Element IDs		
Name	Type	Notes
<i>nameOutput</i>	<i>label</i>	<i>Displays the name of the dog on the screen</i>

Variable(s)		
Name	Type	What it's storing
<i>dogSize</i>	<i>list</i>	<i>A list of dog sizes pulled from the "Max Height" column</i>
Conditional(s)		
Boolean Expression (pseudocode is ok)		Notes
<i>if dog size is greater than 15, append to a list</i>		<i>Inside a for loop that traverses the dogSize list</i>
Loop(s)		
For Loop (pseudocode is ok)		Notes
<i>for(var i=0; i<dogSize.length; i++)</i>		<i>Traverses dogSize list</i>

onEvent(s)		
Element ID	Action	What happens?
"dogButton"	"click"	A picture of a dog appears The background of the screen changes to green
function(s)		
Name	Purpose	How it works
updateScreen	updates what appears on the screen after the user selects a dog	Filters the lists and displays the images and names of dogs on the screen.

Build Phase

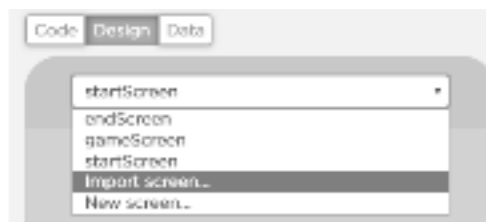
Step 5: Test Screens: Build a quick version of the screens which includes all elements with their proper element IDs. Do not worry about design at all. This is purely to allow the programmer the ability to test their code as they go along.

Step 6: Build

Programmers: Use the chart to guide you in adding programming statements to your program.

Designers: Use the chart to guide you in adding screen elements to your program. You can work on a separate computer from your partner.

- When you have finished screens, your partner should delete the test screens and then import the finished screens.
- Note: If screens are not deleted before importing the project you will get an error message because element IDs will be the same. To avoid this, add a blank screen, delete the old project screens, import the finished screens, and delete the blank screen. Make sure to set your home screen to be the default.



Scoring Guidelines

Written Response (individual) - In the Written Response the student...	
Task	Point
Describes the purpose of the app	/1
Includes a program segment that shows how the initial data has been stored in a list	/1
Includes a program segment that shows how the data in this list is being processed	/1
Identifies the name of the list being processed in this response	/1
Identifies what the data contained in this list is representing in the program	/1
Explains how the list manages complexity by explaining how this program code would be written differently without using this list.	/1
Overall Project (both partners)	
Program uses a dataset in a meaningful way towards the program's purpose	/1
At least one list is traversed using: map, reduce, or filter (indicate which in comment) in a meaningful way towards the program's purpose	/1
Program contains the following programming constructs that are used in meaningful ways towards the program's purpose	
• variable(s)	/1
• function(s)	/1
• conditional(s)	/1
• list(s)	/1
• loop(s)	/1
The app has at least three different screens	
• screen #1	/1
• screen #2	/1
• screen #3	/1
All screens can be easily navigated to through the user interface	/1
All functions include comments that explain purpose and how they work.	/1
No errors in the code	/1
All element IDs have meaningful names	/1
Total	/20

Unit 5 Lesson 17

Project - Hackathon Part 5

Resources

Hackathon Submission on AP Classroom



Description: This document is intended for teachers who use AP Classroom. These steps show how to set up a quiz for your students to submit their U5 Hackathon Project Written Responses on AP Classroom using a similar interface they will use to submit the Create PT.

1. Log on to AP Classroom at myap.collegeboard.org



2. Open the Question Bank



3. Filter by Question Type for only "PT: Formative"



4. Make a quiz with 2 questions.
 - a. The question named "3a - Row 1 - 4 Program Purpose" (**Note:** This question also asks students to upload a video. Choose if you want your students to practice that now or submit this question without the video.)
 - b. The question named "3b - Rows 2 and 3 - 6 Manage Complexity with Lists"

3a - Row 1 - 3 Program Functionality	PT	CRD, AAP	1, 2	SW1 4.A	Preview	Add
3a - Row 1 - 4 Program Purpose	PT	CRD, AAP	1, 2	SW1 1.A	Preview	Add
3a - Row 1 - 5 Program Function and Purpose	PT	CRD, AAP	1, 2	SW1 4.A	Preview	Add
3b - Row 2 - 1 Use of Var	PT	AAP	3, 1	SW1 3.A	Preview	Add
3b - Row 2 - 4 Use of Lists	PT	AAP	3, 3, 3, 10	SW1 3.B	Preview	Add
3b - Rows 2 and 3 - 2 Value of Var	PT	AAP	3, 1	SW1 3.A	Preview	Add
3b - Rows 2 and 3 - 3 Manage Complexity with Vars	PT	AAP	3, 1, 3, 2	SW1 3.A	Preview	Add
3b - Rows 2 and 3 - 5 List Values	PT	AAP	3, 3, 3, 10	SW1 3.C	Preview	Add
3b - Rows 2 and 3 - 6 Manage Complexity with Lists	PT	AAP	3, 3, 3, 10	SW1 3.C	Preview	Add
3c - Row 4 - 1 Procedures without Params	PT	AAP	3, 1, 2, 3, 10	SW1 3.B	Preview	Add

5. Assign the quiz before having your students attempt to submit their work.



Hackathon Project Planning Guide

Project Description

For this project you will work with a partner. Together you will create an app that uses a dataset. There are two roles in this project: designer and programmer. You are both responsible for the overall project, but you will focus on different parts of the project at any given time. On the last day of the project, you will individually complete a written response that will help you practice for the Create Performance Task.

You will submit

- Your final app
- This completed project-planning guide
- A written response

App Requirements

- At least three screens
 - All screens can be easily navigated to through the user interface
- A dataset used in a meaningful way towards the program's purpose
- At least one list is traversed using: map, reduce, or filter (indicate which in a comment) in a meaningful way towards the program's purpose
- Programming constructs: variable(s), function(s), conditional(s), list(s), loop(s)
- All functions include comments that explain their purpose and how they work
- All element IDs have meaningful names
- No errors in the code

Steps

- Collaborate with your partner to pick a dataset
- Decide on an app to build which traverses a list from the dataset
- Create a paper prototype
- **Designer:**
 - Create the screens
 - Support the programmer with pair programming as needed
- **Programmer:**
 - Program the app
 - Support the designer as needed.

Investigate Phase

Step 1. Choose a Dataset: Open a project on Code Studio and look through the different datasets available. Choose one that looks interesting to both you and your partner.

Dataset: _____.

What column(s) of data will you use in your app?

-
-
-

Step 2. Brainstorm an App: Consider the columns of data that you are using. How will your app traverse this data? Circle one and explain below.

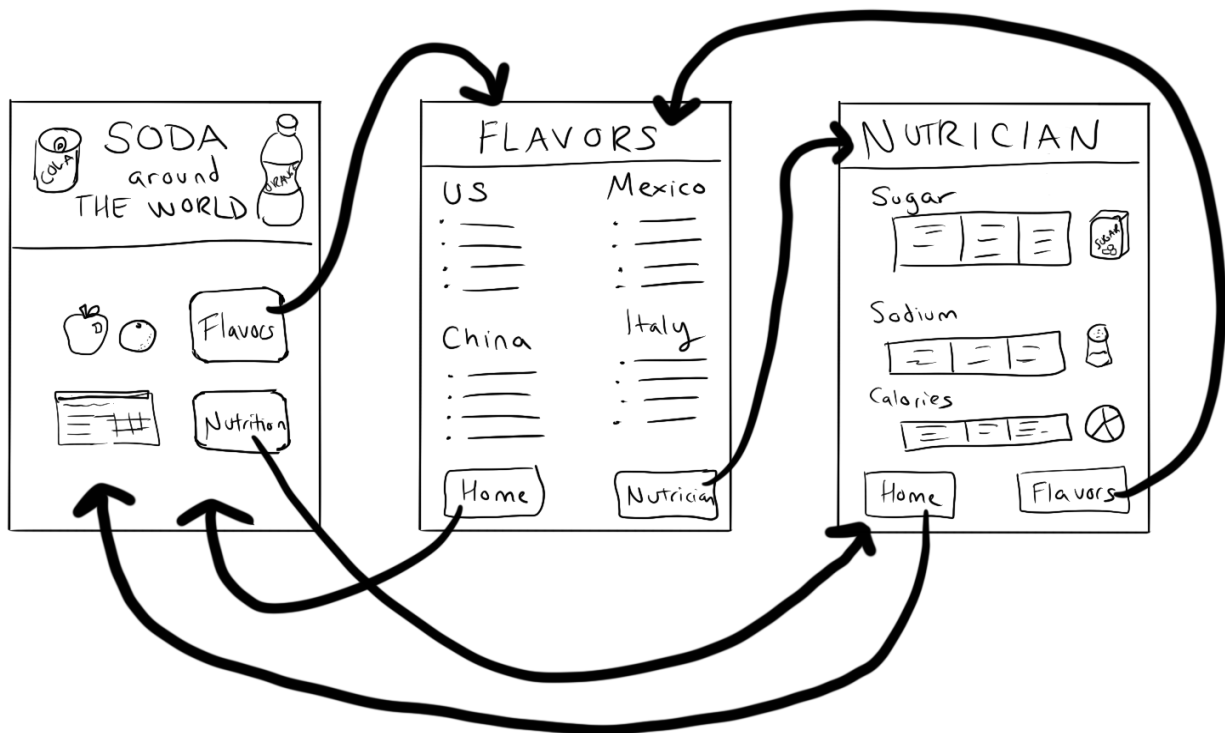
- **Filter** (most common option): use the list from one column to determine information that will be filtered from a list created by another column
 - Example: dogHeight filters dogNames, so only the names of small dogs are added to the filtered list
- **Map:** Add or change each item in a list
 - Example: map a list of numbers pulled from a column using Math.round - now each number is rounded
- **Reduce:** Reduce the data in a list to a single number
 - Example: find the smallest number in a list

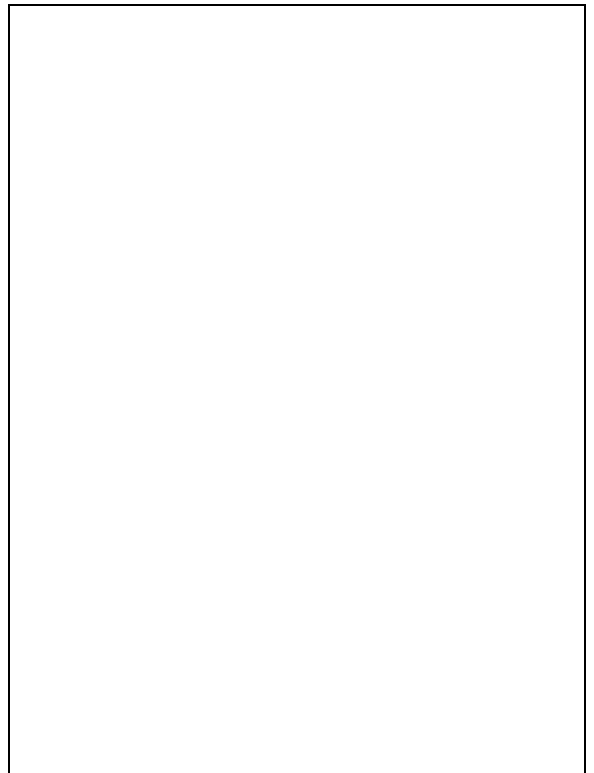
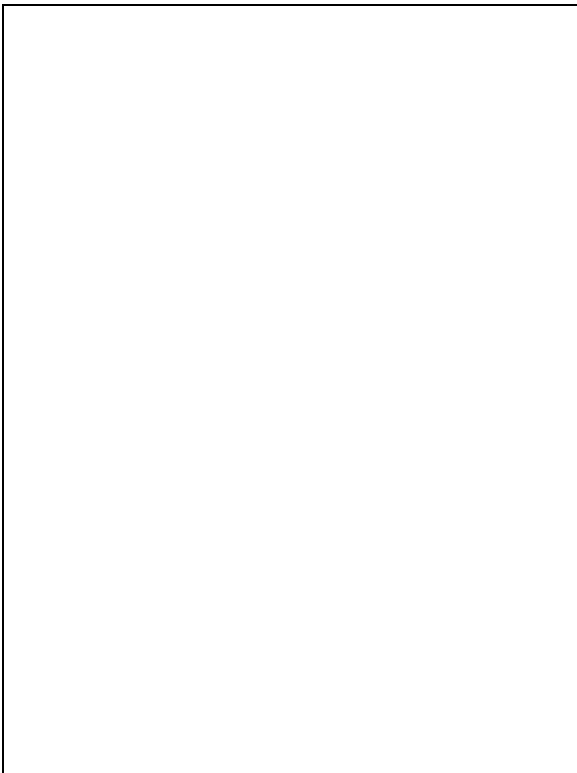
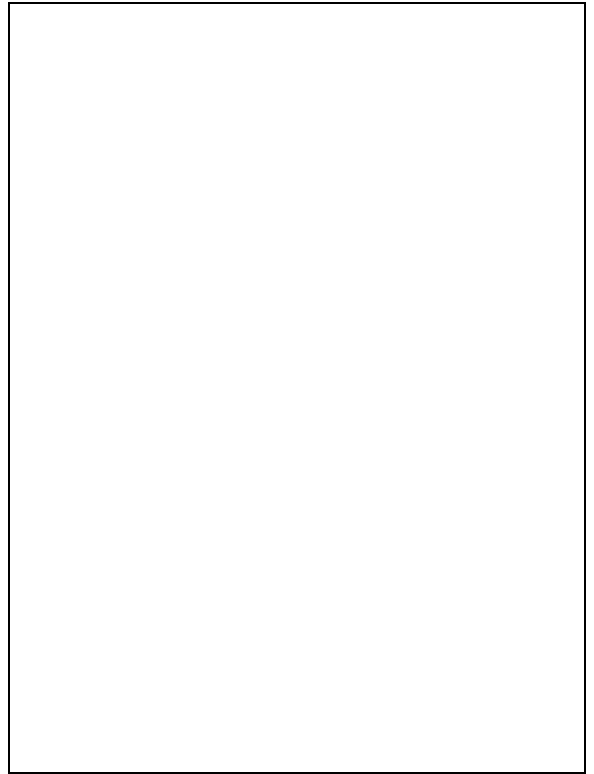
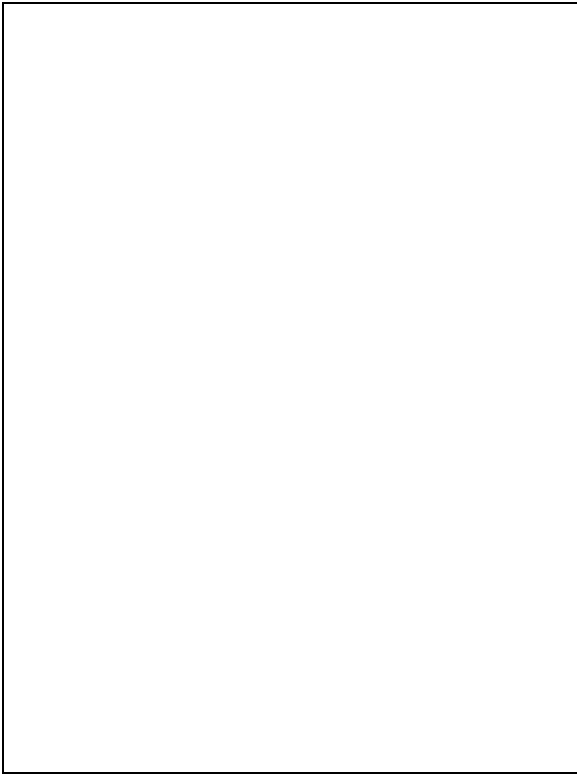
Your app will use (circle): MAP REDUCE FILTER

Explain in more detail:

Design Phase

Step 3. Create a Paper Prototype: On the next page you should draw a **prototype** that shows how your app will actually run. This means you should include all the buttons, text, and images that the user will be able to use and see. Write notes or draw arrows showing how different user interface elements should work. For example, if clicking a button takes me to another screen, I should draw an arrow from that button to the drawing of the screen.





Step 4. Prepare to Build Your App:

Before you begin to code, fill out the chart below. Add more rows (or use a scrap piece of paper) as needed.

Element IDs		
Name	Type	Notes
<i>nameOutput</i>	<i>label</i>	<i>Displays the name of the dog on the screen</i>

Variable(s)		
Name	Type	What it's storing
<i>dogSize</i>	<i>list</i>	<i>A list of dog sizes pulled from the "Max Height" column</i>
Conditional(s)		
Boolean Expression (pseudocode is ok)		Notes
<i>if dog size is greater than 15, append to a list</i>		<i>Inside a for loop that traverses the dogSize list</i>
Loop(s)		
For Loop (pseudocode is ok)		Notes
<i>for(var i=0; i<dogSize.length; i++)</i>		<i>Traverses dogSize list</i>

onEvent(s)		
Element ID	Action	What happens?
"dogButton"	"click"	A picture of a dog appears The background of the screen changes to green
function(s)		
Name	Purpose	How it works
updateScreen	updates what appears on the screen after the user selects a dog	Filters the lists and displays the images and names of dogs on the screen.

Build Phase

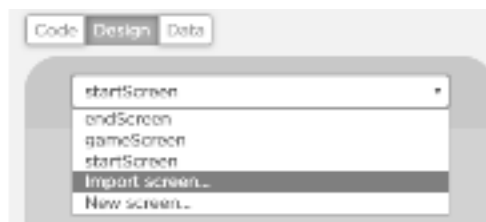
Step 5: Test Screens: Build a quick version of the screens which includes all elements with their proper element IDs. Do not worry about design at all. This is purely to allow the programmer the ability to test their code as they go along.

Step 6: Build

Programmers: Use the chart to guide you in adding programming statements to your program.

Designers: Use the chart to guide you in adding screen elements to your program. You can work on a separate computer from your partner.

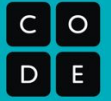
- When you have finished screens, your partner should delete the test screens and then import the finished screens.
- Note: If screens are not deleted before importing the project you will get an error message because element IDs will be the same. To avoid this, add a blank screen, delete the old project screens, import the finished screens, and delete the blank screen. Make sure to set your home screen to be the default.



Scoring Guidelines

Written Response (individual) - In the Written Response the student...	
Task	Point
Describes the purpose of the app	/1
Includes a program segment that shows how the initial data has been stored in a list	/1
Includes a program segment that shows how the data in this list is being processed	/1
Identifies the name of the list being processed in this response	/1
Identifies what the data contained in this list is representing in the program	/1
Explains how the list manages complexity by explaining how this program code would be written differently without using this list.	/1
Overall Project (both partners)	
Program uses a dataset in a meaningful way towards the program's purpose	/1
At least one list is traversed using: map, reduce, or filter (indicate which in comment) in a meaningful way towards the program's purpose	/1
Program contains the following programming constructs that are used in meaningful ways towards the program's purpose	
• variable(s)	/1
• function(s)	/1
• conditional(s)	/1
• list(s)	/1
• loop(s)	/1
The app has at least three different screens	
• screen #1	/1
• screen #2	/1
• screen #3	/1
All screens can be easily navigated to through the user interface	/1
All functions include comments that explain purpose and how they work.	/1
No errors in the code	/1
All element IDs have meaningful names	/1
Total	/20

Hackathon Project Written Response



Question 1: Provide a written response that describes the overall purpose of the program

Write your response here

Question 2: Insert screenshots of two program code segments which contain a list being used in your program.

The first program code segment should show how the data has been stored in the list.

Insert the screenshot of the first program code here

The second program code segment should show the data in the same list being used, such as creating new data from the existing data or accessing multiple items in the list.

Insert the screenshot of the second program code here

Write a written response that:

- Identifies the name of the list being processed
- Identifies what the data contained in the list is representing in your program
- Explains how the list manages complexity in your program by explaining how your program code would be written differently if you did not use the list

Name of the list	
Data contained in the list	
The list manages complexity by...	

Unit 5 Lesson 18

Assessment Day

Resources