[Arrays](https://www.khanacademy.org/computing/computer-programming/programming/arrays/p/intro-to-arrays)

Store multiple values in your variables with arrays!

Video: Intro to Arrays

\*

Challenge: Favorite Fruits

Make the array

You're going to display your top 3 favorite fruits. To get started, make an array of them.

Whoaaa! Such fluffy animals

Display them!

Now, use the text() command to display each item on a different line, so that they're not overlapping.

Show the length!

Now display a header that says 'My top 3 favorite fruits', anywhere!

My

favorite fruits

Video: Modifying Objects

\*

Challenge: Favorite Animals

Make the array

You are going to display your top 3 favorite animals. To get started, make an array of them.

Hint: The animal names are strings, so don't forget to write them between "quotation marks".

Look over them!

In this step, you'll use a while loop to display each of the animals on the screen, using the text command. Make a counter variable to keep track of the name you are drawing, and use it to calculate a different y position each time, so that the names are spaced out in a list.

Tip: Be careful with condition of your while loop, and make sure you increment the variable properly, otherwise your program will get stuck in an infinite loop.

Add more animals!

Now that you have a loop to display your array of animals, test it out by adding at least 2 more animals (5 total).

Video: Modifying Arrays

\*

Challenge: Constellation Maker

Add a star

This program displays an array of stars based on two arrays, one of x positions and one of y positions. Add another star by adding a new number to each array.

xPositions

yPositions

Add stars on click

Now, make this program more dynamic, by making it add stars when the user clicks the mouse. Add a mouseClicked function that pushes a new x and y position onto the arrays based on the current mouse position (mouseX and mouseY), and then redraws the stars.

Now that youâ€™ve pushed the x and y positions of the mouse into the array, be sure to move the call to drawStars() into your mouseClicked() function, so that the stars will get re-drawn every time a new one is created.

Review: Arrays

This is a review of what we covered in this tutorial on arrays.

We often want to store lists of values when we're creating programs, and in JavaScript, we can do that using a type of value called an array.

To create an array, we declare a variable like we always do, but then we surround our list of values with square brackets and separate each value with a comma:

xPositions

We can store any sort of JavaScript value in an array - not just numbers. Here's an example where we store a list of strings:

myFriends

We often want to display the length of an array, or do something based on the length of the array. Thankfully, every array has a length property that will tell us the current length of the array:

Displays "4"

When we want to access a particular value in an array, we access it by referencing its "index" in the array, which represents its position. The first index in an array is "0", so if we want to access the first element in an array, we specify the name of the array variable, then square brackets, and 0:

Displays

The second element is at index "1", the third is at index "2", and the fourth is at index "3":

The zero-based indexing is one of the most confusing aspects of arrays for new programmers, so keep that in mind if you're just getting started with arrays. You'll get used to it eventually!

We often want to take some action for every element in an array, like how we used the text() command to display the names above. Instead of writing that code over and over, it's better to use a for loop to iterate through each of the elements in the array, and do something to each element inside the loop. We have to start from index 0, go until we reach the end of the array, and add 1 to the index each time. Here's how we'd do that:

Notice how we put i inside the square brackets, because it represents the current index each time the loop is run.

Arrays can be changed in many ways. To start off with, we can change a value in them:

TheErrorBuddy

We can also add entirely new values to them, using the push() method, passing in the new value:

After running that line of code, our array length property will change to reflect the new length, and the final index in the array will be 4 instead of 3.

If you want a full list of what you can do with arrays in JavaScript, check out this reference. But don't worry, everything in this review will get you very far!

\*

Spin-off of "Project: Make it Rain"

To make an animation of rain, it's best if we use arrays to keep track of the drops and their different properties. Start with this simple code and build on it to make a cool rain animation. Here are some ideas for what you could do:

Add more drops to the arrays.

Make it so that the drops start back at the top once they've reached the bottom, using a conditional.

Make an array of colors, so that every drop is a different color.

Make other things rain, like snowflakes (using more shape commands) or avatars (using the image commands).

Make it so that when the user clicks, a new drop is added to the array.

Initialize the arrays using a for loop and random() function, at the beginning of the program.