**[Objects](https://www.khanacademy.org/computing/computer-programming/programming/objects/p/intro-to-objects)**

Learn how to store complex data in objects.

Video: Intro to Objects

\*

Challenge: Recipe Card

Customize the recipe

What a boring recipe this is! Change the title and ingredients to one of your favorite recipes... though you may want to simplify it a bit!

recipeTitle

recipeServings

recipeIngredients

Bread

Flour

Water

Serves

Ingredients

Make sure you change the ingredients!

Make sure you have two ingredients in an array.

Make it an object

Now, instead of using three separate variables, make a single variable for the recipe that's an object, and store three properties on it.

recipe

title

servings

ingredients

Now change the text commands to output data from the object variable, using dot notation.

Video: Modifying Objects

\*

Challenge: Picture Painter

Change the initial paintbrush

This program uses an object to describe a paintbrush with x, y, and image properties. Change the initial x and y, and pick a different image that would be fun to paint with.

Tip: If you want to see all the images you can use, click on the string passed to getImage - the handy image picker will appear.

paintBrush

getImage

imageMode

image

Make sure you change the initial image so it's not the green leaf anymore.

Make sure you change the initial x position from its original value of 200.

Make it paint!

Now you'll actually make this program paint! Add a mouseMoved function that:

- changes the paintbrush x and y properties based on the current mouse position, using the

mouseX and mouseY variables;

- calls the painting function, paintCanvas;

Tip: Check the documentation if you're not sure how to use mouseMoved.

paintBrush

Now that you're setting the new position for your image, call the paintCanvas function inside mouseMoved to paint the image on the canvas.

Video: Arrays of Objects

\*

Challenge: Movie Reviews

Display the review

This program starts with an array of one movie, and displays the title. Now, make it so it also displays the review, in a smaller size, below the title.

movies

title

review

Puff the Magic Dragon

Best movie ever!!

textSize

Instead of printing out the title twice, print the title once and the review once.

Can you position the review so it's underneath the title, and has enough space between it and the title?

Add a second movie

Now, add another movie to the array.

Display them all

Now, to get both your movies displaying, use a for loop to iterate through the array and execute the text commands for each movie. If you don't remember how to iterate through arrays, re-watch the previous talk-through and/or Looping through Arrays.

Review: Objects

Hide tutorial navigation

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

We have many types of values that we can store in JavaScript variables, and sometimes we want to store a bunch of related values together: that's where objects come in!

An object is a data type that lets us store a collection of properties in a single variable. To create an object, we declare a variable like we normally would, and then we use curly braces to surround key-value property pairs:

var objectName = {

propertyName: propertyValue,

propertyName: propertyValue,

...

};

Here's an object that describes Winston - this object has two properties, hometown and hair, and each of the property values are strings:

var aboutWinston = {

hometown: "Mountain View, CA",

hair: "no"

};

Here's a more complex object that describes a cat,

with four properties- age, furColor, likes, and birthday.

var lizzieTheCat = {

age: 18,

furColor: "grey",

likes: ["catnip", "milk"],

birthday: {"month": 7, "day": 17, year: 1994}

};

Notice how each property stores a different data type- age stores a number, furColor stores a string, likes stores an array, and birthday stores another object. That's the cool thing about objects (well, one of the cool things) - they can store other objects inside them! So you can have deeply nested objects to describe complex data.

You might also see objects declared using quotes around the property names, like:

var aboutWinston = {

"hometown": "Mountain View, CA",

"hair": "no"

};

That's equivalent to what we saw before with no quote marks, but it takes longer to type. The only time that you absolutely need quote marks is if your property name has a whitespace in it, like:

var aboutWinston = {

"his hair": "none"

};

We have to use quotes there, because otherwise the JavaScript interpreter would get confused. Here's my tip for you: just don't use whitespace in your property names to begin with! Then you never have to use quote marks around property names.

Accessing object properties

An object is not useful unless we can look inside it and grab the values of the different properties. We can do that two ways - first, using what we call "dot notation", where we write the name of the variable, followed by a ".", and then the property name:

var aboutWinston = {

hometown: "Mountain View, CA",

hair: "no"

};

text("Winston is from " + aboutWinston.hometown,

100, 100);

text("Winston has " + aboutWinston.hair + " hair", 100, 150);

We can also use "bracket notation", which looks similar to how we access array elements, where we write the variable name, then square brackets, with the property name inside in quotes:

var hisHometown = aboutWinston["hometown"];

var hisHair = aboutWinston["hair"];

If we try to access a property that doesn't exist, we'd see "undefined":

text("Winston's life goal is " +

aboutWinston.lifeGoal, 100, 150);

Modifying object properties

Just like when we store other data types in variables, we can change the values of the object properties at any time during a program, using the dot or bracket notation:

aboutWinston.hair = "curly"; // Winston gets a wig!

We can also add entirely new properties!

aboutWinston.lifeGoal = "teach JavaScript";

If we're done with a property, we can delete it (but most of the time we'll probably just change its value):

delete aboutWinston.hair;

Arrays of objects

Now that you know both arrays and objects, you can combine them to make arrays of objects, which are actually really useful ways of storing data in programs. For example, an array of cats:

var myCats = [

{name: "Lizzie",

age: 18},

{name: "Daemon",

age: 1}

];

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

var myCat = myCats[i];

println(myCat.name + ' is ' + myCat.age + ' years old.');

}

Notice that we iterate through an array of objects the same way that we iterate through an array of numbers or strings, using a for loop. Inside each iteration, we access the current array element with bracket notation, and then access the properties of that array element (an object) with dot notation.

Here's another practical example that you might use in your programs, an array of coordinate positions:

var positions = [

{x: 200, y: 100},

{x: 200, y: 200},

{x: 200, y: 300}

];

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

var position = positions[i];

ellipse(position.x, position.y, 100, 100);

}

Pretty neat, aye? Objects can be confusing at first, but keep using them, and eventually you'll be addicted and turn everything into an object!

\*

Spin-off of "Project: Bookshelf"

What are your favorite books? Make a shelf of them!

This program only displays one book. Make an array of books (at least two of them), and use a loop to display a row of them.

Add an author property to each book, and display the author below the title.

Add a new property to each object to store a color, using the color command, and use that to fill each book differently. (Or maybe store an image?)

Add a boolean property to each book (like whether you would recommend it to friends), and use a conditional inside the loop to draw something different depending on whether the boolean is true or false.

Now add more books, and use the loop to draw more shelves down the canvas. Think about how you can use conditionals and/or the % operator to do that.

Don't want to make a bookshelf? You could also make a shelf of videos or video games, or re-create the home screen of your smartphone.

book

title

The Giver

draw shelf

draw one book

getImage

cute star