**JavaScript objects**

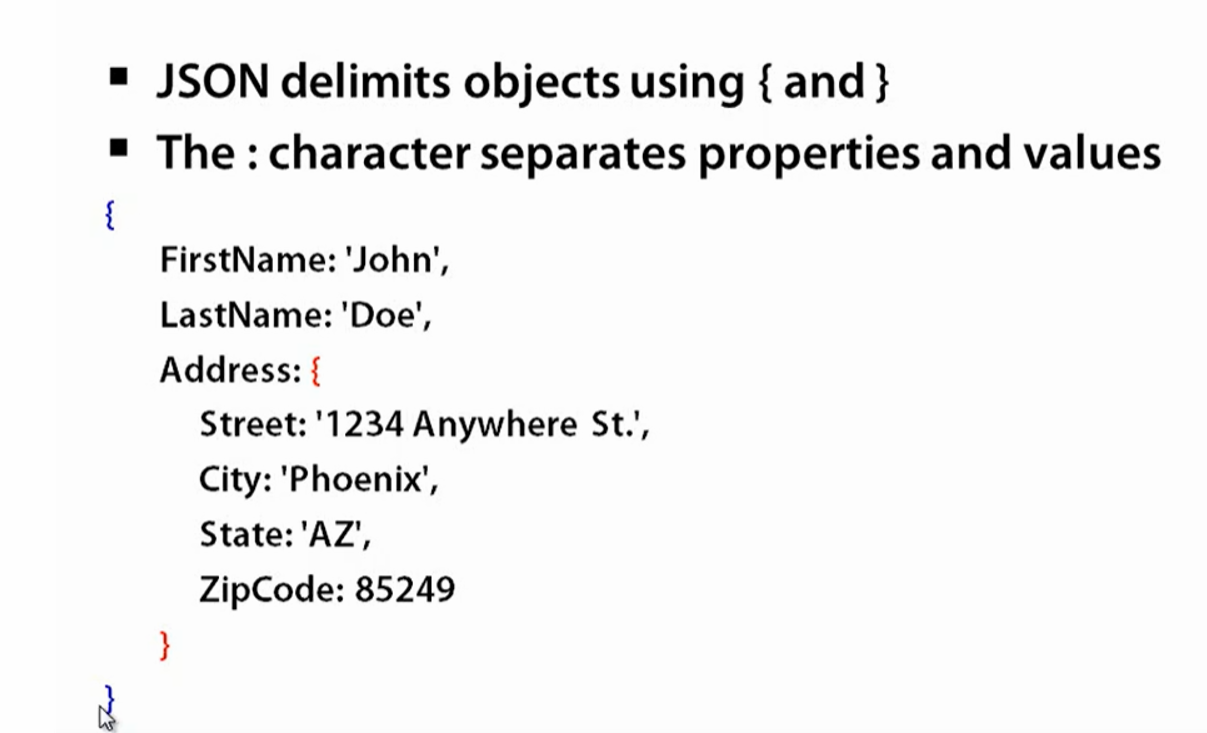
In programming there are multiple ways to store data in the code, we have been using variables for a while. In the last few classes we learned to use the array. But in this class we are going to learn about a very important way to store data in the code. This is so important that without this method your internet will crash in seconds, most of the data on the internet comes in the form of this data structure. This is called JavaScript objects.

This allows us to store the data at one place, which belongs to a certain object. For example: You are a student, you have certain information associated with you such as your name, in which school you study, your favorite subject, your marks [30,45,35] etc. It is not a good practice to keep this data in an array because it involves different data types, such as strings, characters, integers, even an array itself. Also if you want to store the data of multiple students then it will be very tedious to access the data of the correct student, because we have to remember the index of each student's data points. Here, JavaScript objects come to our help. This will allow us to store your data at one place and then we can access and change it very easily.

To define an object, first we declare a variable name and set it equal to empty curly brackets. var Student = { }; At the end of the parentheses there will be a semicolon. Once we have our empty object ready, we can start adding the data in this. Since this object is named as student, we are going to add data which is relevant to the student.

To access the data from an object, we use the name of the object .(dot) and the name of the key you want the value of.





**Game States:**

Let us introduce a variable which will hold the value of the game state. Game State could be PLAY or END. Let us set the initial state to be PLAY when we run the code. We use CAPITAL LETTERS in the name of those variables which hold constant values that do not change inside the program.

We will put an ‘if and else if’ condition inside the function draw().

“if and else if” block will do one thing if one condition is satisfied and it will do another thing if the other condition is satisfied.

There are some behaviours in the game which we want when the gameState is PLAY. We will put if (gameState === PLAY)code behaviors related to these statements inside . There are other behaviors in the game we want when the gameState is END. We will put code behaviors related to those statements inside the else if (gameState === END). There are some behaviors in the game we want irrespective of the gameState. We will put them outside the if else condition.