Arrays Hold multiple values in comma-separated list

```
// Arrays can hold a mix of data types
let numArray = [13, 36, 45, 57, 14];
let mixedArray = ["a", "b", 1, 2, 3];
```

.length Property tells us the length of an array

```
let fruits = ["apple", "banana", "orange"];
// Assign array length to a variable
const numFruits = fruits.length;
console.log(numFruits); // Output: 3
```

Array Methods Used for editing array elements

```
// Adds element to end of array
fruits.push("pear");

// Adds element to beginning of array
fruits.unshift("strawberry");

// Removes and returns the first element
fruits.shift();

// Removes and returns the last element
fruits.pop();

// Returns the index of the value
fruits.indexOf("apple");

// Returns true or false if value in array
fruits.includes("dragonfruit");
```

Function Keyword used to define a function

```
// JS function names are in camelCase
function myFunction() {
  console.log("Hello, World!");
}
myFunction();
```

Return Keyword sends value back from a function

```
function myFunction() {
  return "Hello, World!";
}
// Will not appear on console
myFunction();
```

Objects Store multiple key : value pairs, or properties

```
const book = {
  title: "The Hobbit",
  author: "J.R.R. Tolkien",
  year: 1937
};
```

Dot Notation Used to access, add, & update object properties

```
// Access a property
console.log(book.title);
// Output: The Hobbit

// Assign a new property
book.edition = "1st Edition";

console.log(book);
/* Output: {
title: "The Hobbit",
year: 1937,
author: 'J.R.R. Tolkien',
edition: '1st Edition'} */

// Update a property
book.edition = "2nd Edition";

console.log(book);
/* Output: {
title: "The Hobbit",
year: 1937,
author: 'J.R.R. Tolkien',
edition: '2nd Edition'} */
```

Object Methods Function defined as properties

```
const dog = {
  name: "Benny",
  sound: "woof",
  // Method returns a string
  makeSound() {
   return dog.name + " says " + dog.sound + "!")
 }
};

console.log(dog.makeSound());
// Output: Benny says woof!
```