**Variables**

Variables are a named reference to a value. They let you access values through a predetermined name.

Variables are declared using:

1. Const
2. Let
3. Var

const name = 'Steve';

var name = 'Lisa';

let name = 'Charlotte';

**Arrays**

An array is an ordered collection of data (either primitive or object depending upon the language). Compared with variable that can add only one value, arrays are used to store multiple values in a single variable.

let names = ['Steve', 'Lisa', 'Charlotte'];

**Objects**

An object is a collection of properties, and a property is an association between a name (or key) and a value.

let student = {

name: 'Steve',

age: 20,

classes: {name: 'Intro to Programming', name: 'Advanced Algebra'}

}

**Functions**

Functions are one of the fundamental building blocks in JavaScript. A function in JavaScript is similar to a procedure—a set of statements that performs a task or calculates a value, but for a procedure to qualify as a function, it should take some input and return an output where there's some obvious relationship between the input and the output.

A function definition (also called a function declaration, or function statement) consists of the function keyword, followed by:

* The name of the function
* A list of parameters to the function, enclosed in parentheses and separated by commas
* The JavaScript statements that define the function, enclosed in curly brackets, {...}

Here is a function, called sayName.

function sayName(name) {

console.log(name);

}

The function sayName takes one parameter, called name. The function consists of one statement that says to print the parameter of the function (i.e. name) to the console.

Primitive parameters (such as a number) are passed into functions by value. The value is also passed to a function, but if the function changes the value of the parameter, this change is not reflected globally or in the calling function.

If you pass an object (i.e., a non-primitive value, such as an array or a user-defined object) as a parameter and the function changes the object's properties, that change is visible outside the function as shown in the following example:

function myFunc(theObject) {

theObject.make = 'Toyota';

}

var mycar = {make: 'Honda', model: 'Accord', year: 1998};

var x, y;

x = mycar.make; // x gets the value "Honda"

myFunc(mycar);

y = mycar.make; // y gets the value "Toyota"

// (the make property was changed by the function)

**For loops**

The for statement creates a loop that consists of three optional expressions, enclosed in parentheses and separated by semicolons, followed by a statement (usually a block statement) to be executed in the loop.

let str = '';

for (let i = 0; i < 9; i++) {

str = str + i;

}

console.log(str);

// expected output: "012345678"

Source:

MITxPro Bootcamp

<https://executive-ed.xpro.mit.edu/professional-certificate-coding?utm_source=Google&utm_medium=c&utm_term=%2Bmit%20%2Bcoding&utm_location=9007574&utm_campaign=B-365D_US_GG_SE_PCC_Brand&utm_content=MIT-Coding___School_Duration&gclid=Cj0KCQjw3duCBhCAARIsAJeFyPVYZJwvrGBn_93CMgrokibm87uQ_OnDXxiBDYwzsRiQcPqa2_vVawoaAiLCEALw_wcB>

[MDN](https://developer.mozilla.org/en-US/)