Variables

When a program has a value that needs to be used often or saved it is assigned to a variable. When you give a name to or define a value (to make it easier to reference) the value becomes a variable. The value of a variable can change over time. A variable whose value does not change is called a constant. There are two types of variables- primitive and object. Primitive variables are made up of numbers, texts, or Boolean truth values. Any variable that doesn’t fit into those categories are object types of variables or a collection of properties which are defined by a name and value.

There are three ways to declare your variables (let everyone know that x value is to x variable), let, const, and var. Let and const are in JavaScript version ES6 and on and var is in versions prior to ES6. Let can declare multiple variables in one statement, however, you need to define the initial value for the variable when using let. If this is forgotten an error will pop up.

Var is similar to let. In previous versions of JavaScript there was no way to declare constants and var was the only way to declare all variables. With var you can declare the same variable many times and the definitions are hoisted to the top of the function. However, if you don’t initialize the variable right away and use it, you will not get an error- which can cause bugs.

Const is used when you want to declare a constant (a variable in which the value never changes), for example, program version numbers or byte sequences. Like let, you need to initialize the variable when you declare it. If you try to change the value, it becomes a TypeError. Many users declare consts’ in all caps to make it easy to differentiate between them and lets’. The main benefit of using const all the time is to rule out bugs that may occur because of value switches. When changes need to be made you can switch variables to let.

Null is a value that is used to indicate the absence of a value. Null is a type of its own, it shows when there is no value for numbers, strings, or objects. It is also known as NULL, nil, and None. Undefined is a deeper type of null. It means that the value does not exist, not just that it is missing. It is a “predefined global constant that is initialized to the undefined value” (Flanagan, 40). In simple terms, null is more missing and undefined is more error. They can be used interchangeably and behave like false when Boolean is required.

Resources:

FLANAGAN, DAVID. (2020). JavaScript the definitive guide (7th ed.). O'REILLY MEDIA, INC, USA.