Variable Declaration

There are two types of variables, the Declared Variables and the Undeclared Variables. Undeclared Variables , on the other hand, are variables that have no variable declaration statements such as *var* and *let.* In contrast with the Declared Variables, Undeclared Variables can be accessed anywhere in the document, regardless of the execution context of the variable.

Declared Variables

Declared Variables are declared using the *var* statement or the *let* statement, and can only be accessed either within its execution context.

*var* statement

* Example : var variableName = *value*

A variable declared with the var statement can be accessed globally within the document. It is optionally initialized to a value.

*let* statement

* Example : let variableName = value

A variable declared with the let statement is a block scope local variable. Which means that it can only be accessed to the block, statement, or expression on where it is used. It is optionally be initialized to a value. Also, unlike the var statement, let statements declare variables that do not become properties of the window object.

*const* statement

* Example : const variableName = value

The const statement has similarities with the var and let statements. Like the let statement, variables declared with the const statement are block scoped, although can be still be accessed in anywhere when declared outside enclosing functions like the var statement, and do not become properties of the window object. Moreover, it declares read-only variable with a constant value, which cannot be changed.

Undeclared Variables

Undeclared Variables, on the other hand, are variables that have no variable declaration statements such as *var* and *let.* In contrast with the Declared Variables, Undeclared Variables can be accessed anywhere in the document, regardless of the execution context of the variable.

Data Types

Variables in Javascript are dynamically type or loosely typed which means they do not need to be explicitly declared to a specific data type. Even though data types of variables are implicitly declared, there is still a way of determining them using the *typeof* operator.

There are six primitive data types supported by Javascript:

1. Boolean
2. null
3. undefined
4. Number
5. String
6. Symbol

And an object

Literals

Boolean Literals

* true and false

Numeric Literals

Decimal Integers

* + are sequence of digits that are not preceded with zero (0)
  + (e.g. 1, 2, 100)

Floating-point Numbers

* + Can have the following parts:
    - A decimal that can either be preceded with a “+” or a “-” (signed), or not (unsigned).
    - A decimal point (“.”);
    - A fraction (decimal number after the decimal point)
    - An exponent that is delimited by the “e” or “E” character
  + [(+|-)][digits][.digits][(E|e)[(+|-)]digits]
  + (e.g. -5.1E+7, 1.134, .2e-34 )

Binary Integers

* + Binary digits preceded by 0b or 0B.
  + (e.g. 0b1010, 0B1100)

Octal Integers

* + Octal digits preceded by 0 or 0o or 0O.
  + (e.g. 01234567, 0o1234567, 0O1234567)

Hexadecimal Integers

* + Hexadecimal digits and characters preceded by 0x or 0X.
  + (e.g. 0xb, 0xB, 0x123, 0X123)

String literals

* Sequence of characters enclosed in double (“”) or single (“”) quotation marks.
* (e.g. ‘string’, “string”, ‘2431’, ‘first line \n second line’, “It’s a mouse”)
* Note: Read more here - <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Grammar_and_Types>

null Value

* Assigned programmatically to a variable to denote that it has no value.

undefined Value

* When a variable is undefined it means that is has not been declared or assigned a value yet.

Symbol