**Learning Objectives**

* To understand the concept of variables in a program;
* To grasp variable definition and initialization method in JavaScript.

**Learning Contents**

* To put it simply, variables are containers storing information. A variable is like a box, which is used to store information, e.g. “Bob”, “true” and 35. In order to distinguish information stored in different boxes, it is necessary to name every box, i.e. a “variable name”. Of course, information stored in the box can be divided into different data types. For example, if it stores a name “Bob”, it is a string; if it stores an age “35”, it is a number. Different languages have different data types. JavaScript also has its own data type, which will be discussed in the next section. There are normally two steps before use of a variable: declaration and initialization. Declaration is like that you got a box and said the box is used to store names. In other words, it designates a variable name. Initialization is like that you put information in the box. For example, you put “Bob” in the box.
* In JavaScript, variables are loosely typed, which means the variables can save any type of data. In other words, every variable is merely a placeholder of a value saved by a user. When a variable is defined in JavaScript, we use **var** to declare a variable, which is followed by a variable name. The variable name must be legitimate identifiers, including upper and lower case letters, arrays, underline ‘**\_**’ and dollar sign ‘**$**’ but it cannot start with a number. For example:
* **var message = 'Hello World'; *// Right!***
* **var 2age = 30; *// Wrong! The variable name cannot start with a number.***
* As mentioned in the preceding page, variables in JavaScript are loosely typed. However, it does not mean there is no data type in JavaScript. Instead, it means when we store information with a box, it is not necessary to stipulate the box can store numbers and rather, any type of information, including numbers, strings and dates, can be stored. However, data itself has data type.
* We can declare a variable like this:
* **var information;**
* This line of code defines a variable named **information**. The variable can store any value (such uninitialized variables will be saved as a special value - **undefined**）.
* In JavaScript, when a variable has been declared and initialized, value of the variable can be modified. For example:
* **var name = 'Bob';**
* **name = 'Mike';**
* We can see from the above codes, the name in the second line is changed into **’Mike‘**. As we mentioned before. JavaScript is loosely typed. Can we modify a variable like this?
* **var name = 'Bob';**
* **name = 10; *// Valid! But not recommended.***
* In this case, the variable **name** saved a string value **’Bob‘** first. Then the value is substituted with a number **10**. Although this method is absolutely valid in JavaScript, we do not recommend to modify type of value saved when we modify a variable.
* Sometimes, we may need to define multiple variables simultaneously, e.g. name, age and student number. We can define a variable like this:
* **var name = 'Bob';**
* **var age = 16;**
* **var stuNo = 0012323;**
* ***// Or, we can define like this:***
* **var name = 'Bob', age = 16, stuNo = 0012323;**
* With the first method, we define three variables respectively. By comparing the above two methods, we find that the semicolon is modified into a comma in the second method and only one **var** defines three variables at the same time. Similarly, since JavaScript is loosely typed, use of different types of initialized variables can be finished in one statement. Although linefeed and variable indentation in the codes are not mandatory, they can improve readability of codes.

**Recommended Resources**

* JS Variable - MDN（<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/var>）
* How to Declare Variables in JS?（<https://www.sitepoint.com/how-to-declare-variables-javascript/>）