**Learning Objectives**

* To learn about seven data types in JavaScript

**Learning Contents**

* We mentioned in the last section that every kind of programming language has different data types, or rather, whether a string or a number is stored in the box.
* JavaScript is a kind of **weakly typed** or **dynamic** language.It means there is no need to declare type of a variable in advance. The type will be determined automatically during running of the program. It also means we can use one variable to save different types of data:
* **var foo = 42; *// foo is a Number now***
* **var foo = "bar"; *// foo is a String now***
* **var foo = true; *// foo is a Boolean now***
* The most recent JavaScript standard defines 7 data types, including:
  + 6 primitive data types:
    - **Boolean**
    - **Null**
    - **Undefined**
    - **Number**
    - **String**
    - **Symbol (It is newly defined in ES6 and will not discussed in this section.)**
* Primitive: In JavaScript, all types other than Object are immutable (the value cannot be changed). We call these types of value as “primitive values”.
* Boolean: It represents a logical entity， which means true and false. It has two values: **true** and **false**.
* Null: It has one value only: **null**, which means null.
* Undefined: A variable which is not assigned with a value has a default value: **undefined**.
* Number: In JavaScript, numbers range from - (263-1) - (263-1). Except specific numbers, there are also some values with symbols in JavaScript: **+Infinity**, **-Infinity** and **NaN**(Not-a-Number), which represent positive infinity, negative infinity and not-a-number. For example: In JavaScript, numbers range from -(263-1) - (263-1)
* **19 / +0; *// Infinity***
* **19 / -0; *// -Infinity***
* String:
  + String in JavaScript is used to represent text data. Every element in string occupies a position of a string. Index of the first element is 0; the next one is index 1 and so forth. Length of a string indicates number of its elements.
  + String in JavaScript requires single quote **'\*\*'** or double quotes **"\*\*"** to indicate the value is a string.
  + String in JavaScript cannot be modified. It means once a string is created, it cannot be modified. However, a new string can be created based on operations on original string. For example:
    - We can select certain letter or use **String.substr()** to get a substring of a string.
    - We can use a joint operator (**+**) or **String.concat()** to connect two strings.
  + Symbol: Symbols are newly defined in ES6. Symbol is unique and cannot be modified.
  + Object:
    - Like other objects in programming language, objects in javascript can be comprehended by referring to objects in real life. Concept of objects in programming language can be comprehended by referring to physical objects in actual life.
    - In javascript, an object can be a separate entity with attribute and type. We can use a cup as a metaphor. A cup is an object, which has an attribute. The cup has color, pattern, weight and material and so forth. Similarly, a javascript object also has an attribute to define its features.
    - We can create an Object by affixing object type to be created after the operator **new**. And we can create a custom object by creating an object type case and add its attribute and (or) method. For example:
    - **var o = new Object();**
    - We can also create an object directly with the following method:
    - **var person = { name: 'Bob', age: 20, gender: 'male' };**
    - The above object defines a boy of 20 years old named ’Bob‘.
* **typeof** operator
  + Since JavaScript is loosely typed, it is necessary to detect and determine data type of variables with certain method - **typeof** is an operator responsible for providing information in this aspect. If we apply the operator **typeof** to a value, the following string might be returned:
    - **'undefined'** - undefined;
    - **'boolean'** - Boolean;
    - **'string'** - String;
    - **'number'** - Number;
    - **'object'** - Object or **null**
    - **function** - Function
  + An example of use of **typeof** is shown as below:
  + **var message = 'some string';**
  + **alert(typeof message); *// "string"***
  + **alert(typeof(message)); *// "string"***
  + **alert(typeof 95); *// number***
  + In the practical programming, we can use **typeof** to determine data type of any variable.

**Recommended Resources**

* JS Data Type - MDN（[https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Data\_structures）](https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Data_structures%EF%BC%89)
* JS Object - MDN（[https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Guide/Working\_with\_Objects）](https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Guide/Working_with_Objects%EF%BC%89)
* JS Data Type and Variables - Learning Javascript（[http://archive.oreilly.com/pub/a/javascript/excerpts/learning-javascript/javascript-datatypes-variables.html）](http://archive.oreilly.com/pub/a/javascript/excerpts/learning-javascript/javascript-datatypes-variables.html%EF%BC%89)