**Learning Objectives**

* To grasp mathematical operators in JavaScript, e.g. **+**，**-**，**\***，**/**，**+=**……
* To perform mathematical operations with mathematical operators in JavaScript.

**Learning Contents**

* In the preceding study, we know variables can be used to store information, including strings and numbers. In this section, we are going to learn how to implement basic mathematical operations with JavaScript.
* Let’s do a simple math： We have 15 apples and 20 pears. How many pieces of fruits do we have totally？ The answer is 35. Then how can we express this exercise with a program? For example:
* **var apple = 15;**
* **var pear = 20;**
* **var fruit = apple + pear;**
* **alert(fruit); *// 35***
* We should first declare the number of apples stored by a variable (**var apple = 15;**), then the number of pears stored by a variable (**var pear = 20;**) and finally the number of fruits stored by a variable. After that we initialize it to the sum of apples and pears (**var fruit = apple + pear;**), **fruit** stores the number of fruits we work out. Here （**+**） has the same meaning as in math, i.e. plus.
* First of all, let me introduce assignment operator, i.e. the equal sign **'='**. In JavaScript, an equal sign means assignment. That is to say, the value on the right side of the equal sign is assigned to the value on the left side of the equal sign. People tend to confuse assignment and judgment. One equal sign represents assignment. If we want to determine if **a** equals **b**, we need to use **===** or **==**. As for difference between **===** and **==**, to put it simply, three equal signs means strict equivalence, not only in value but also in data type. However, two equal signs represent equivalence in value only, not in data type. For example:
* **alert('0' === 0); *// false Strict equivalence. ’0‘ is a string, while 0 is a number, so it is not strictly equal.***
* **alert('0' == 0); *// true Not strict equivalence. ’0‘ will be converted into 0 through implicit conversion. By comparison, 0==0, so true will be returned.***
* Mathematical operators in JavaScript included:
* +: Addition.For example:
* **var x = 5;**
* **var result = x + 5;**
* **alert(result); *// 10***
* -：Subtraction. For example:
* **var x = 5;**
* **var result = x - 5;**
* **alert(result); *// 0***
* \*：Multiplication. For example:
* **var x = 5;**
* **var result = x \* 5;**
* **alert(result); *// 25***
* /：Division. For example:
* **var x = 5;**
* **var result = x / 2;**
* **alert(result); *// 2.5***
* %，means to get the remainder. For example: 10 divided by 3 is 3 and remainder 1, so % of 10:3 is 1. For example:
* **var x = 5;**
* **var result = x % 2;**
* **alert(result); *// 1***
* ++，means its value plus 1. For example: 3++ equals 4. For example:
* **var x = 5;**
* **x++;**
* **alert(x); *// 6***
* +=，for example:
* **var x = 5;**
* **x += 2; *// x = x + 2;***
* **alert(x); *// 7***
  + += means the value of current variable plus the value on the right side and then assigned to the current value. In the above case, **x += 2;** The value of **x** plus **2** and then assigned to **x**.
* --，means its value minus 1. For example, 4 – equals 3. For example:
* **var x = 5;**
* **x--;**
* **alert(x); *// 4***
* -=，for example:
* **var x = 5;**
* **x -= 2; *// x = x - 2;***
* **alert(x); *// 3***
  + -= means the value of current variable minus the value on the right side and then assigned to the current value. In the above case, **x -= 2;** The value of **x** minus **2** and assigned to **x**.
* **>** / **>=**, is used to made judgment in JavaScript. Their concepts are basically the same as greater than/greater than or equal to in math and true or false will be returned as a result. For example:
* **var x = 5;**
* **var y = 10;**
* **var z = 10;**
* **alert(y > x); *// true***
* **alert(y >= z); *// true***
* **<** / **<=** , is used to made judgment in JavaScript. Their concepts are basically the same as less than/less than or equal to in math and true or false will be returned as a result. For example:
* **var x = 5;**
* **var y = 10;**
* **var z = 10;**
* **alert(y < x); *// false***
* **alert(z <= x); *// false***
* Preference
  + There are various preference rules in math calculation. For example, multiplication and division come before addition and subtraction; the part in brackets comes before the part outside the brackets. These rules also apply to math operation in JavaScript. For example:
  + **alert(2 + 3 \* 3); *// 11***
  + **alert((1 + 2) \* 2 + 2); *// 8***
  + **alert((2 + 2) / 2); *// 2***

**Recommended Resources**

* JS Expression and Operators - MDN（[https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Expressions\_and\_Operators）](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Expressions_and_Operators%EF%BC%89)
* JS Operators - w3school（[https://www.w3schools.com/jsref/jsref\_operators.asp）](https://www.w3schools.com/jsref/jsref_operators.asp%EF%BC%89)
* Difference between == and === in JS（[https://appendto.com/2016/02/vs-javascript-abstract-vs-strict-equality/）](https://appendto.com/2016/02/vs-javascript-abstract-vs-strict-equality/%EF%BC%89)
* Precedence of Operators - MDN（[https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Reference/Operators/Operator\_Precedence）](https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Reference/Operators/Operator_Precedence%EF%BC%89)