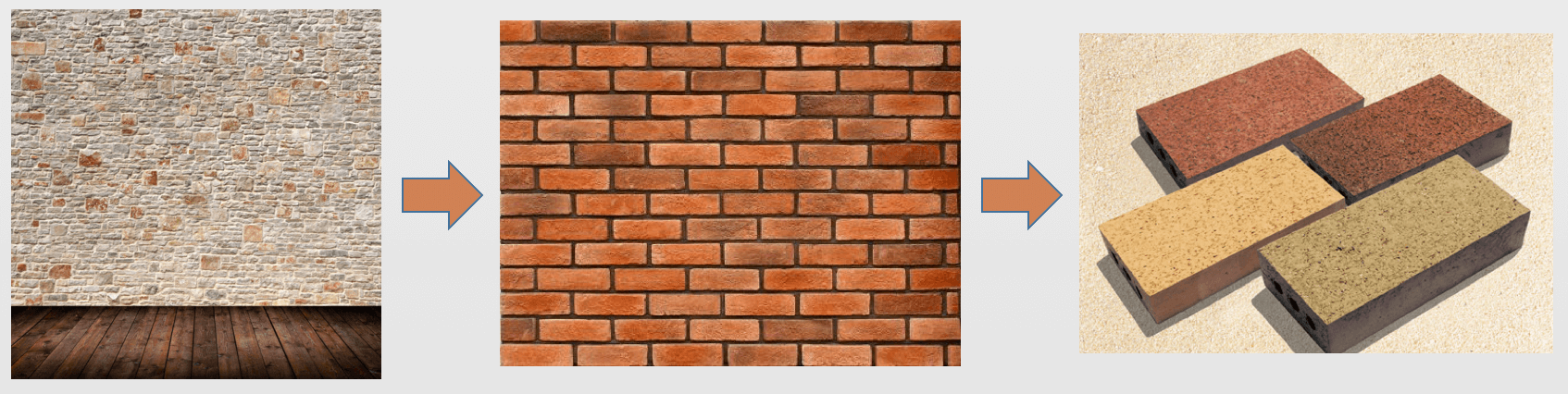
**Learning Objectives**

* To understand the concept and roles of function;
* To understand the roles of parameters in function;
* To grasp function definition and calling methods in JavaScript;

**Learning Contents**

* For any language, function is a core concept. Through functions, we can encapsulate multiple statements, which can be called at any time and any place. So what is a function?
* In a piece of computer program code, function is mainly used to solve the problem of program code sharing. The initial computer program is executed by sequence, but if multiple programs need to be executed with the same computation, codes for the same computation will appear for several times and repetition occurs. If a function is used, the situation will not be like this. If the computation rules are encapsulated in the same function, we can just call the function where necessary. For example:

-

* In the above figure, there is a beautiful wall. When we take the beautiful coat of the wall off, we find the wall is made of many bricks. If we compare a built-up wall to a program, the bricks may be regarded as functions. Function is a repeatable code block that can execute certain function. If we want to execute it in program, we just need to input and call parameters. Then the function will return us the execution result.
* In JavaScript, we use the keyword function to define a function which is followed by a set of parameters and function bodies. Basic syntax of the function is as below:
* **function functionName(arg0, arg1, arg2 ... argn){**
* **statements;**
* **}**
* Here is a simple function case:
* **function sayHi(name, message){**
* **alert('Hello ' + name + ', ' + message) + '!';**
* **}**
* This function can be called through other function name. Behind it is a pair of round brackets and parameters (if there are several parameters in the round brackets, they should be separated with comma.) Codes calling function sayHi() is shown as below:
* **sayHi('Mike', 'good morning'); // 'Hello Mike, good morning!'**
* Output of the function is 'Hello Mike, good morning!'. Naming parameters defined in the function, name and message, are used as two operands combined by string. The final result is displayed in the warning bar.
* The above case is a case without designated return value. When we define a function in JavaScript, we need not designate whether there will be a return value. As a matter of fact, any function can negate value to be returned through a return statement to realize the return value. Here is an example with return value:
* **function add(num1, num2){**
* **return num1 + num2;**
* **}**
* The role of the add function is sum up the two parameters and return the value. Except return statement, no other declaration indicates the function will return a value. Here is an example of calling this function:
* **var result = add(3, 5);**
* **alert(result); *// 8***
* The final result will be 8. We can notice that, we need a variable (result) to receive values returned by add(3,5), as we return the results in the add() function. If we do not receive them with a variable, we cannot operate on the results.

**Recommended Resources**

* JavaScript Function - MDN(<https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Reference/Global_Objects/Function>)
* JavaScript Function Summary(<http://www.qeefee.com/js/javascript-function-summary>)
* js Function Summary(<http://www.jianshu.com/p/421f158fbbb4?utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation>)