**Learning Objectives**

* To understand what a scope is;
* To understand why a scope is needed;
* To grasp scope principles in JavaScript.

**Learning Contents**

* Scope is used to descript all entities (or valid identifiers) visible in certain code block. More precisely, it can be called the context.
* Why a scope is needed?
  + Minimum accesses
  + What is the benefit of restricting visibility of a variable or forbidding availability of all things in the codes at any place? One advantage is that scope provides a security level for you codes. In computer security, there is a general principle: users can only access to things they need currently.
* Scope chain of JavaScript
  + First of all, please loot at the following codes:
  + <script type="text/javascript">
  + **var rain = 1;**
  + **function rainman(){**
  + **var man = 2;**
  + **function inner(){**
  + **var innerVar = 4;**
  + **alert(rain);**
  + **}**
  + **inner(); *//Call inner function***
  + **}**
  + **rainman(); *//Call rainman function***
  + </script>
  + Please observe the code alert (rain);. JavaScript first search in the inner function whether a variable rain is defined. If it is defined, the variable rain in the inner function is used; if not, JavaScript will continue to search in the rainman function whether a variable rain is defined. If the rainman function in the code does not define a variable rain, JavaScript engine will continue to search up (global objects) whether rain is defined. We defined rain = 1 in the global objects, so '1' will pop up finally.
  + Scope chain: When JavaScript needs to search a variable x, it will search the first object of the scope chain first. If the first object does not define the variable x, JavaScript will continue to search the second object and determine if the second object defines the variable x. If the second object does not define the variable x, it will continue to search.
  + The above codes involve three scope chain objects, inner, rainman, and window.
* Within the body of function, a local variable is superior to a global variable with the same name.
  + First of all, please look at the following codes:
  + <script type="text/javascript">
  + **var rain = 1; *//It defines a global variable rain.***
  + **function check(){**
  + **var rain = 100; *//It defines a local variable rain.***
  + **alert( rain ); *// 100 will pop up.***
  + **}**
  + **check();**
  + **alert( rain ); *//1 will pop up.***
  + </script>
* There is no block scope but function scope.
  + This is what makes JavaScript more flexible than other languages.
  + Please observe the following codes carefully. You will find scopes of the variable i, j, and k are the same and they are global in the body of the entire function rain.
  + <script type="text/javascript">
  + **function rainman(){**
  + ***// The body of function rainman contains three local variables, i, j and k.***
  + **var i = 0;**
  + **if ( 1 ) {**
  + **var j = 0;**
  + **for(var k = 0; k < 3; k++) {**
  + **alert( k ); *//0, 1 and 2 will pop up.***
  + **}**
  + **alert( k ); *//3 will pop up.***
  + **}**
  + **alert( j ); *//0 will pop up.***
  + **}**
  + </script>
* All variables declared in the function are defined in the entire function.
  + First of all, please observe the following codes:
  + <script type="text/javascript">
  + **function rain(){**
  + **var x = 1;**
  + **function man(){**
  + **x = 100;**
  + **}**
  + **man(); *//Call man***
  + **alert( x ); *//100 will pop up.***
  + **}**
  + **rain(); *//Call rain***
  + </script>
  + The above codes indicate, the variable x can be used in the body of the entire function rain and can be reassigned with a value. Due to this rule, an incredible result will occur. Please observe the following codes:
  + <script type="text/javascript">
  + **var x = 1;**
  + **function rain(){**
  + **alert( x ); *// 'undefined', instead of 1 will pop up.***
  + **var x = 'rain-man';**
  + **alert( x ); *// 'rain-man' will pop up.***
  + **}**
  + **rain();**
  + </script>
  + As the local variable x in the function rain is defined in the entire function (var x= 'rain-man', and a declaration is made), the global variable x with the same name is hidden in the entire function rain. The reason why 'undefined' pops up is that when alter(x) is executed first, the local variable x has not been initialized yet. Therefore, the above function rain is equivalent to the following function:
  + **function rain(){**
  + **var x;**
  + **alert( x );**
  + **x = 'rain-man';**
  + **alert( x );**
  + **}**
  + All variables not declared with key words var, let and const are global variables.
  + <script type="text/javascript">
  + **function rain(){**
  + **x = 100; *//It declares the global variable x and assigns a value.***
  + **}**
  + **rain();**
  + **alert( x ); *//100 will pop up.***
  + </script>
  + This is a common error of JavaScript novices, who leave many global variables unintentionally.
* All global variables are the attribute of the object window.
  + <script type="text/javascript">
  + **var x = 100 ;**
  + **alert( window.x );*//100 will pop up.***
  + **alert(x);**
  + </script>
    - It is equivalent to the following codes.
  + <script type="text/javascript">
  + **window.x = 100;**
  + **alert( window.x );**
  + **alert(x)**

</script>

**Recommended Resources**

* Understand JavaScript Scope（<http://web.jobbole.com/91134/>）
* Variable Scope（JavaScript）（<https://msdn.microsoft.com/zh-cn/library/bzt2dkta(v=vs.94).aspx>）
* JavaScript Scope（<http://www.runoob.com/js/js-scope.html>）