

09-第九章 作用域 js预解析

一、作用域

作用域：脚本的有效范围，作用范围。分两大类：全局(script) 和 局部(function)

1.全局(script)作用域

1) 直接定义在script标签下的 变量 及 函数 ，他们都作用在一个域，全局作用域，so..

```
1.    <script>
2.        var a = 123;
3.        alert( window.a );//123
4.        function abc(){}
5.        alert( window.abc );//function abc(){}
6.    </script>
```

2) 直接定义在script标签下的 变量 称之为 全局变量 ,script标签下的 函数 ，称之为全局函数

3)全局 变量 及 函数 都是 window的一个属性 ,都能通过 window.变量名 访问

```
1.    <script>
2.        var a = 123;
3.        alert( window.a );//123
4.        function abc(){}
5.        alert( window.abc );//function abc(){}
6.    </script>
```

2.局部 (function) 作用域

- 1) 任何一个function(){} ,都会开启一个局部作用域 ,
- 2) 定义在function(){} 内部的变量称之为 局部变量

```
1.     var a = 20;  
2.     function fn(){  
3.         alert(a);//20  
4.     }
```

- 3) 作用域链 : 局部作用域内部可以访问 父级作用域变量 及 全局作用域变量 , 也可以访问父级的函数 , 及全局函数 (往上爬)

```
1.     function fn(){  
2.         var a=10;  
3.     }
```

- 4) 局部变量会覆盖父级 (全局) 变量 , 函数亦如此

```
1.     var a = 10;  
2.     function fn(){  
3.         var a=20;  
4.         alert( a );//20  
5.     }
```

二、javascript解析

javascript解析 即 读取代码过程

- 1) javascript解析 是 致上而下

- 2) 预解析:正式解析前的工作 , 预解析过程会出现 变量提升 , 函数提升

```

1.  function (){
2.      alert( b );//undefined
3.      var b = 10 ;
4.  }

```

a) **变量提升**：在作用域内声明的变量会被提升到作用域的顶部，且对其赋值 `undefined`，这个过程称之为 **变量提升**，上面代码等同于下面

还可以访问父级的 **参数**

```

1.  function (){
2.      var b = undefined;
3.      alert( b );//undefined
4.      var b = 10 ;
5.  }

```

b) **函数提升**：在作用域内的函数定义 **函数会被提升到作用域的顶部**，其值为其函数本身，这个过程称之为 **变量提升**，上面代码等同于下面

```

1.  function (){
2.
3.      alert( fn );
4.      //function fn(){}
5.      //code....
6.      function fn(){}
7.  }

```

c) **不会提升的函数**：在作用域内的函数表达式 **函数不会被提升到作用域的顶部**，so ~

```

1.  function (){
2.
3.      alert( fn );//undefined
4.
5.      //code....
6.      var fn = function(){}
7.  }

```

d) 重名的变量只留一个，var和函数重名 **函数优先**

三、案例

```
1. var x = 5;
2. function a(){
3.     alert(x);
4.     var x = 10;
5. }
6. alert(x);
```

```
1.     alert(a);//function a(){var a = 20;alert(a);}
2.     var a = 10;
3.     alert(a);//10
4.     function a(){
5.         alert( 20 );
6.     }
7.     alert(a);//10
8.     var a = 30;
9.     alert(a);//30
10.    var a = function a(){
11.        var a = 20;
12.        alert(a);
13.    }
14.
15.    alert(a);//function a(){var a = 20;alert(a);}
16.    a();//20
17.    alert(a);//function a(){var a = 20;alert(a);}
18.
```

四、带return 的作用域

```
1. var a = 5;
2. function fn(){
3.     var a = 10;
4.     alert(a);
5.     function b(){
6.         a++;
7.         alert(a);
8.     }
9.     return b;
10. }
11.
12. var c = fn(); //10
13. c();//11
14. fn()();//10 11
15.
16. c();//12
17.
```