[**深入理解JavaScript的变量作用域**](http://www.cnblogs.com/jenry/archive/2010/12/08/1900314.html)

在学习JavaScript的变量作用域之前，我们应当明确几点：

    a、JavaScript的变量作用域是基于其特有的作用域链的。

    b、JavaScript没有块级作用域。

    c、函数中声明的变量在整个函数中都有定义。

**1、JavaScript的作用域链**

首先看下下面这段代码：

[?](http://www.cnblogs.com/jenry/archive/2010/12/08/1900314.html)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | <script type="text/javascript" language="javascript">    var rain = 1;  function rainman(){      var man = 2;      function inner(){          var innerVar = 4;          alert(rain);      }      inner();    //调用inner函数  }  rainman();    //调用rainman函数    </script> |

观察alert(rain);这句代码。JavaScript首先在inner函数中查找是否定义了变量rain，如果定义了则使用inner函数中的 rain变量；如果inner函数中没有定义rain变量，JavaScript则会继续在rainman函数中查找是否定义了rain变量，在这段代码中rainman函数体内没有定义rain变量，则JavaScript引擎会继续向上（全局对象）查找是否定义了rain；在全局对象中我们定义了 rain = 1，因此最终结果会弹出'1'。

作用域链：JavaScript需要查询一个变量x时，首先会查找作用域链的第一个对象，如果以第一个对象没有定义x变量，JavaScript会继续查找有没有定义x变量，如果第二个对象没有定义则会继续查找，以此类推。

上面的代码涉及到了三个作用域链对象，依次是：inner、rainman、window。

**2、函数体内部，局部变量的优先级比同名的全局变量高**

[?](http://www.cnblogs.com/jenry/archive/2010/12/08/1900314.html)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | <script type="text/javascript" language="javascript">    var rain = 1;    //定义全局变量 rain  function check(){      var rain = 100;    //定义局部变量rain      alert( rain );    //这里会弹出 100  }  check();  alert( rain );    //这里会弹出1    </script> |

**3、JavaScript没有块级作用域。**

这一点也是JavaScript相比其它语言较灵活的部分。

仔细观察下面的代码，你会发现变量i、j、k作用域是相同的，他们在整个rain函数体内都是全局的。

[?](http://www.cnblogs.com/jenry/archive/2010/12/08/1900314.html)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18 | <script type="text/javascript" language="javascript">    function rainman(){  /\*\*   \* rainman函数体内存在三个局部变量 i j k   \*/      var i = 0;      if( 1 ){          var j = 0;          for( var k = 0 ; k < 3 ; k++ ){              alert( k );    //分别弹出 0 1 2          }          alert( k );    //弹出3      }      alert( j );    //弹出0  }    </script> |

**4、函数中声明的变量在整个函数中都有定义。**

首先观察这段代码。

[?](http://www.cnblogs.com/jenry/archive/2010/12/08/1900314.html)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | <script type="text/javascript" language="javascript">    function rain(){      var x = 1;      function man(){          x = 100;      }      man();    //调用man      alert( x );    //这里会弹出 100  }  rain();    //调用rain    </script> |

 上面得代码说明了，变量x在整个rain函数体内都可以使用，并可以重新赋值。由于这条规则，会产生“匪夷所思”的结果，观察下面的代码。

[?](http://www.cnblogs.com/jenry/archive/2010/12/08/1900314.html)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | <script type="text/javascript" language="javascript">    var x = 1;  function rain(){      alert( x );    //弹出 'undefined'，而不是1      var x = 'rain-man';      alert( x );    //弹出 'rain-man'  }  rain()    </script> |

这是由于在函数rain内局部变量x在整个函数体内都有定义（ var x= 'rain-man'，进行了声明），所以在整个rain函数体内隐藏了同名的全局变量x。这里之所以会弹出'undefined'是因为，第一个执行 alert(x)时，局部变量x仍未被初始化。

所以上面的rain函数等同于下面的函数。

[?](http://www.cnblogs.com/jenry/archive/2010/12/08/1900314.html)

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | function rain(){      var x;      alert( x );      x = 'rain-man';      alert( x );  } |

**5、未使用var关键字定义的变量都是全局变量。**

[?](http://www.cnblogs.com/jenry/archive/2010/12/08/1900314.html)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | <script type="text/javascript" language="javascript">  function rain(){      x = 100;    //声明了全局变量x并进行赋值  }  rain();  alert( x );    //会弹出100  </script> |

这也是JavaScript新手常见的错误，无意之中留下的许多全局变量。

**6、全局变量都是window对象的属性**

[?](http://www.cnblogs.com/jenry/archive/2010/12/08/1900314.html)

|  |  |
| --- | --- |
| 1  2  3  4  5 | <script type="text/javascript" language="javascript">  var x = 100 ;  alert( window.x );//弹出100  alert(x);  </script> |

等同于下面的代码

[?](http://www.cnblogs.com/jenry/archive/2010/12/08/1900314.html)

|  |  |
| --- | --- |
| 1  2  3  4  5 | <script type="text/javascript" language="javascript">  window.x = 100;  alert( window.x );  alert(x)  </script> |