**Js中的作用域**

1. **什么是作用域？**

我们把它一分为二。作用和域

域：就是一个空间，一个范围 ，一片区域。哪么什么是区域呢？用<script></script>作演示，只有在这个区域里的，js语句才有用

作用：我们定义了一个var a = 1;哪么我们在它的下面就可以读到它和写它了。

哪么作用域就是在一个范围或者空间里，你可以对某个值进行读和写的操作就叫作用域。

**例：**

alert(a);

var a = 1;

为什么？

**例：**

alert(a);

a = 1;

为什么？

这个问题咱们先留着，我们想要了解作用域，首先得了解**浏览器**

**二、浏览器的js解析器**

我们知道浏览器是一个综合体，你在里面拖入图片，它显示图片，拖入文字，它显示文字，这里我们只关心读取js的这部分。我们把它叫做js解析器。（这个名字有可能不太准确，我们自己理解用的）

浏览器从上到下执行时，当读到<script></script>标签时，js解析器就起来干活了。（我们想要了解作用域，其实就是了解js解析器它的工作方式）

**三、js解析器它是如何工作的？**

1、根据“参数、function、var ”来找一些东西，放入它的仓库中，我们称为“预解析”

2、逐行解读代码。

**例：**

<script>

alert(a);

var a = 1;

function fn1(){alert(2);}

</script>

预解析过程：当读到<script>标签时，就找“参数、function、var ”，把它们放入仓库中。

找参数、function 、var 的规则说明：

1、所有的变量，在正式运行代码之前，都提前赋了一个值：undefined

2、所有的函数，在正式运行代码之前，都是整个函数块

3、参数，同变量

哪么上面代码的预解析的结果：

参数：没有

function : function fn1(){alert(2);}

var：a = undefined

遂行解读代码：

实际上每读取一行，它都会跑到它的这个仓库里来看一看。看看里面有没有东西。

读取代码是从上向下读取。

当它读到alert(a);。要跑到仓库里去看一看，看这个仓库里有不有。

接着读第二行，它见着一个表达式

var a叫做声明，function fn1(){alert(2)}叫函数声明。

a = 1叫表达式。

我们发现能改变值的就叫表达式。表达式就是一个动做。

见着表达式，它就去改仓库里的值。

**总结：**

先预解析，再遂行解读代码，遂行解读时，就是改仓库里的值。

哪我们再来分析我们最开始的哪个例子，读解这两个例子。

**例：**

alert(a);

var a = 1;

alert(a);

function a (){ alert(2); }

alert(a);

var a = 3;

alert(a);

function a (){ alert(4); }

alert(a);

alert(typeof a)

// a();

发散：我们一上来，心情要放平和，你现在就是js解析器，模拟它的工作。打开记事本，当做一个仓库，来一步一步做。

1. 预解析：参数、function 、var

注意：预解析的过程中，遇到变量和函数重名的，只会留一个。

函数覆盖变量。后面的函数覆盖前面的函数。

2、执行代码

**四、域，我们再来分析这个域**

**全局：**<script>标签里的 全局变量、全局函数（自上而下执行）

如果页面有两个<script>标签，它会遵循一个<script>标签预解析再遂行读取完成之后，再进到下一个<script>标签。即一块处理完了，再处理下一块。

**如：**

<script>

alert(a)

</script>

<script>

var a = 1;

</script>

把它们交换一下位置再看看

说明：我们为什么要把base.js文件放在前面。

**局部：**由里到外（就近原则，即作用域链）

域：除了script这个全局的，还有函数。只要是域，就会开辟一个新的作用域（仓库），发生预解析，遂行解读。（函数是个局部的域）。当在局部作用域中没有找到值时，就顺着作用域链，返到父级的域里面来找。（由里到外）

**例：**

var a = 1;

function fn1(){

alert(a);

var a = 2;

}

fn1();

alert(a);

**例再变个形：**

var a = 1;

function fn1(){

alert(a);

a = 2;

}

fn1();

alert(a);

**例：**

var a = 1;

function fn1(a){

alert(a);

a = 2;

}

fn1();

alert(a);

说明：参数本质就是一个局部变量

**例：**

var a = 1;

function fn1(a){

alert(a);

a = 2;

}

fn1(a);

alert(a);

说明：遂行解读代码时，实际上是从参数处开始读。

读参数时，其实相当于function fn1(var a = 1;){}

特别说明：两个a是不相干的，一个是局部的，一个是全局的。

function fn(a){

console.log(a);

var a=2;

function a(){}

console.log(a)

}

fn(1);

**例：**

var num = 0;

function fn1(){

num++;

}

function fn2(){

num--;

}

fn2();

fn1();

fn2();

alert(num);

if (!("a" in window)) {

var a = 1;

}

alert(a);

**例：**

function fn1(){

var a = '钻石项链';

}

fn1();

alert(a);

说明：我想拿到这个项链，怎么办呢。

**例子变形：**

var str = '';

function fn1(){

var a = '钻石项链';

str = a;

}

fn1();

alert(str);

说明：利用一个全局变量，来获取局部变量里的值。

**例：**

function fn2(){

var a = '钻石项链';

fn3(a);

}

fn2();

function fn3(a){

alert(a);

}

说明：fn3是一个全局函数，可以在局部里调用，调用时，可以传参。这样就把参数传出来了。

**哪么if、for等是不是一个作用域呢。**

if(){}中的大括号不是个作用域。

for(){}中的大括号也不是个作用域。

**例：**

alert(a);

function fn1(){

var a = 1;

}

执行报错，证明函数是个作用域。因为在外面找不到函数里面的变量。

**再例：**

alert(a);

if(true){

var a = 1;

}

执行弹出undefined，证明if不是个作用域。

再例：

alert(fn1);

if(true){

function fn1(){

alert(123);

}

}

我们说if不是个作用域，这是不是应该弹出整个函数块。但是火狐和谷歌弹出undefined，只有IE正常弹出。

说明：火狐和谷歌不对能if、for等等大括号包着的里面的函数进行预解析。而IE能正确解析。

结论：我们在if或者for里面定义的函数，其实你是想定义的是一个全局的变量或者函数，所以， 请把它挪出来，别放在if或for的里面。这样就没有问题了。

如：

alert(fn1);

function fn1(){

alert(123);

}

if(true){}

这样就没有问题了。

**特别说明：** if或for的里面，最好函数和变量都不要放。放在全局的里面。

**现实工作中的例子，点击一个按钮，让几个按钮都变黄。**

<input type="button" value="按钮" />

<input type="button" value="按钮" />

<input type="button" value="按钮" />

var aBtn = document.getElementsByTagName('input');

for( var i=0; i<aBtn.length; i++ ){

aBtn[i].onclick = function (){

//alert(i);

aBtn[i].style.background = 'yellow';

}

}

为什么报错？i为什么成了3，我们就要用我们今天学的来分析一下。

当for执行完了，才可以点击按钮。当点击时，for已经完了。此时i++已经到了3，找不到第四个按钮，所以报错。

**再改改这个例子：**

var aBtn = document.getElementsByTagName('input');

for( var i=0; i<aBtn.length; i++ ){

aBtn[i].onclick = function (){

// alert(i); // i的值？

for( var i=0; i<aBtn.length; i++ ){

aBtn[i].style.background = 'yellow';

}

}

}

1、现在这个i弹出几？

2、如果这时把循环中的var删除，这时i是几？