第一次写博，还蛮激动。。。

看到了三题经典题型，经老师讲解后，对此类题目有了更深刻的认识

就我目前的认识对此题进行总结。如有错误，敬请指正

首先，我们先明确一下JS引擎的工作步骤：

js引擎工作分为两步：

1.将这个js中的变量和函数声明保存到当前（注意，是当前）执行环境的变量对象中

2.再逐行解析执行js。当看到一个函数或一个变量时，js引擎就会去查询 这个函数或变量是在哪里定义的

查询的方式有两种：

a.按作用域链查找：先在当前执行环境查询这个变量或程序，如果没有，就到父执行环境 中查找，还没有就继续向上查找，一直到window

b.从对象中查找：如果一个函数是用对象来调用的，就可以从这个对象中查找，如果 在执行环境中没有找到，会报错，如果在对象中没有找到，会返回undefined

照我个人的理解：每一个函数都是一个独立的执行环境，在每一个执行环境中，都有两个工作步骤，也就是1和2，下面就题目分析

第一题：

[?](https://www.2cto.com/kf/201608/540480.html)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | 1 var a = 10;  2 function test(){  3 　　a = 100;  4 　　alert(a);  5 　　alert(this.a);  6 　　var a;  7 　　alert(a);  8 }  9 test(); |

答案：100 10 100

就代码解析：函数的工作域是window，首先进行第1步：将变量和函数声明保存到当前执行环境的变量对象中，个人对此句话的理解是：引擎工作的第一步是将变量和函数的声明提前，相当于：

[?](https://www.2cto.com/kf/201608/540480.html)

|  |  |
| --- | --- |
| 1  2 | var a;  var test; |

之后开始第2步：逐行解析执行js，首先将10赋值给a，然后跳过没有自执行的test函数，开始第9句，第9句是一句对test（）函数调用，执行环境要注意是window，因此，test函数中的this指向的必然是window。进入test函数内部，与开始时的步骤相同，第一步，将变量及函数声明提前至解析函数之前，即：

[?](https://www.2cto.com/kf/201608/540480.html)

|  |  |
| --- | --- |
| 1 | var a; |

之后开始对函数内代码逐条执行，首先将100赋值给a，则此时，a 结果为100；而this指的是调用此函数的对象，由第9句，由上述的分析知this指的是window，因此this.a 指的是window中的变量a，其值为10，因此第5句结果为10；第六句在test解析时就被提前了，所以可直接跳到第7句，此时的a指向的还是函数内部的变量a，因此结果为100。

第二题：

[?](https://www.2cto.com/kf/201608/540480.html)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | 1 var a = 100;  2 function test(){  3     alert(a);  4     var a = 10;  5     alert(a);  6 }  7 test(); |

答案：undefined 10

就代码解析：函数的工作域是window，依旧是进行第1步：

[?](https://www.2cto.com/kf/201608/540480.html)

|  |  |
| --- | --- |
| 1  2 | var a;  var test; |

之后进行逐行解析，第1行将100赋值给a，第7行调用此函数，在被调用的函数test（）内，首先有

[?](https://www.2cto.com/kf/201608/540480.html)

|  |  |
| --- | --- |
| 1 | var a; |

第2步，逐行解析，a在其执行环境中已经被定义，所以不会有是全局变量的可能，但因第一步只是把变量声明提前，而非将其赋值，因此在第3句中，a是一个声明但未初始化的值，因此其结果为undefined。第4句将10赋值给a，因此第5句结果为10。

第三题：

[?](https://www.2cto.com/kf/201608/540480.html)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | 1 var a = 100;  2 function test(){  3     alert(a);  4     a = 10;  5     alert(a);  6 }  7 test();  8 alert(a); |

答案：100 10 10

就代码解析：函数的工作域是window，依旧是进行第1步：

[?](https://www.2cto.com/kf/201608/540480.html)

|  |  |
| --- | --- |
| 1  2 | var a;  var test; |

逐行解析：第1行将100赋值给a，第7行调用此函数，在被调用的函数test（）内，第一遍并没有变量声明或函数声明，因此进入第2步，依旧是逐行解析，第3行，a在其执行环境中并无声明，由开头介绍的js引擎的第2步a中介绍，先在当前执行环境查询这个变量或程序，如果没有，就到父执行环境 中查找，还没有就继续向上查找，一直到window。因此a在test中并非局部变量，而是window的全局变量，即a的结果是100；第4行中，全局变量a的值修改为10，因此第5行结果为10；在函数结束后，全局变量a的值已经被修改为10，因此第8行结果为10。