

模块二 ES新特性 与TypeScript、js性能优化

一、请说出下列最终的执行结果，并解释为什么

```
1 var a = [];  
2 for(var i = 0; i < 10; i++){  
3   a[i] = function(){  
4     console.log(i)  
5   }  
6 }  
7 a[6]()
```

答: 结果是10，原因是，在函数调用的时候，发现，在当前作用域中没有变量i，因此，需要沿着作用域链向上找，找到i的时候，i已经变为10了。

二、请说出下列最终的执行结果，并解释为什么

```
1 var tmp = 123;  
2 if(true){  
3   console.log(tmp)  
4   let tmp  
5 }  
6 }
```

答: 结果是 Uncaught ReferenceError: Cannot access 'tmp' before initialization，编译执行过程是，在编译阶段, 通过var 声明的 tmp 被压入执行上下文的变量环境，并初始化为undefined，执行过程中，变量环境中的tmp被赋值为123，执行到代码块时，let 声明的tmp 被压入到执行上下文词法环境的栈顶，但是没有初始化，接着执行console.log(tmp)这条语句的时候，从词法环境的中获取到了tmp，这是一个未被初始化的tmp，因此就会抛出这样的错误。

三、结合ES6新预防，用最简单的方式找出数组中的最小值

```
1 var arr = [12, 34, 32, 89, 4]
```

答:

```
1 var min = Math.min(...arr)
```

四、请详细说明var， let， const 三种声明变量的方式直接的具体差别

答:

- var 声明的变量，的作用域最小是函数作用域，let，const声明的变量作用域最小是块级作用域
- 在执行一段函数代码时，需要先编译，编译过程中，函数内的所有通过var 声明的变量都会被放到执行上下文的环境作用域中，并被初始化，也就是变量提升；通过let，const声明的变量，如果没有被块包裹，则会放到上下文的词法作用域中，但是不会初始化，let，const声明的变量，如果被块包裹，则不会被放到词法作用域中。
- 在编译后执行的过程中，进入一个块时，块中的let，const就被压入改执行上下文的词法作用域的栈顶，同样不会被初始化
- 由于变量不能在初始化前使用，所以var声明的变量，在声明前引用不会报错，因为var的变量提升。而let，const声明的变量，如果在声明前引用就会保持。
- const 声明的变量，不能改变， let 声明的变量可以改变。

五、出下列最终的执行结果，并解释为什么

```
1  var a = 10;
2  var obj = {
3    a: 20,
4    fn(){
5      setTimeout(()=>{
6        console.log(this.a)
7      })
8    }
9  }
10 obj.fn();
```

答: 结果是 20， 编译执行过程是，因为箭头函数中的this指向沿着作用域链向外查找的上级作用域的this，而他的上级作用域是fn，obj调用fn，因此this是obj，obj.a == 20

六、简述Symbol类型的用途

答: 可以解决对象中键重复的问题。可以作为类的私有属性。

七、说说什么是浅拷贝，什么是深拷贝

答: 浅拷贝是单层拷贝，深拷贝是多层拷贝。浅拷贝会复制子属性的地址，深拷贝不能复制子孙属性的地址，只复制对象结构。

八、请简述TypeScript 与 JavaScript 之间的关系

答: TypeScript是JavaScript的超级，他包括 JavaScript， ES6+（支持新特性）， 类型系统。

九、请简述TypeScript 优缺点

答:

优点:

- 开发者可以直接使用es6+新特性，最低可被编译成es3
- 利用它的类型系统，可以避免开发过程的类型异常，提高可靠性
- 代码提示更加智能
- 可减少不必要的类型判断

缺点:

- 多了学习成本
- 多了开发成本

十、描述引用计数的工作原理和优缺点

答: 工作原理: 通过引用计数器，在每次引用关系改变时，将修改引用计数器，引用数字为0的时候，立即回收。

优点:

- 即时性，发现垃圾立即回收
- 最大限度减少了程序暂停

缺点:

- 无法回收循环引用的对象

十一、描述标记整理算法的工作流程

答:

- 从一组根元素开始，递归遍历者组根元素，在这个遍历过程中，可达元素标记为活动对象。
- 回收非活动对象所占据的内存
- 内存整理

十二、描述V8中新生代存储区垃圾回收流程

答:

对半划分新生代区域，一半为对象区域，一半为空闲区域，新加入的对象会放到对象区域，当对象区域快被写满时，就需要执行一次垃圾清理操作。

- 垃圾回收过程中，首先对对象区域中的垃圾做标记，标记完成后就进入垃圾清理阶段。
- 垃圾回收器会将这些存活的对象复制起来到空闲区域中，同时它还会把这些对象有序的排列起来，所以这个复制过程，也就相当于完成了内存整理操作，复制后空闲区域就没有内存碎片了。
- 完成复制后，对象区域和空闲区域进行角色对调。这样就完成了垃圾对象的回收操作。

十三、描述增量标记算法在何时使用及算法原理

答:

为了降低老生代的垃圾回收而造成的卡顿，V8将标记过程分为一个个子标记过程，同时让垃圾回收标记和js应用逻辑交替进行，直到标记阶段完成，我们把这个算法称为增量标记。