

百度 面试题

1、如何提升页面加载速度，并简述原理

答案：

合并压缩js、css文件

延迟加载不需要的资源

使用sprites合并细碎的小图片

使用内嵌的base64图片代替url

对静态资源使用CDN

合理配置缓存策略

服务端启用gzip

支持http2

减少阻塞脚本，使用async

ssr后端渲染

减少重定向

2、下面程序输出的结果为:

```
1 function add(m){
2     return function b(n){
3         return n+m++
4     }
5 }
6 var add1 = add(070);
7 var add2 = add(050);
8 console.log(add1(010));
9 console.log(add2(010));
```

答案：

js将0开头的数默认为8进制的数

(70).toString(8) = 56

(50).toString(8) = 40

(010).toString(8) = 8

56+40 40 + 8

3、下面的代码输出的结果为：

```
1  var baidu = {  
2      count: 2,  
3      getBaiduCount: function() {  
4          return this.count;  
5      }  
6  };  
7  console.log(baidu.getBaiduCount());  
8  var func = baidu.getBaiduCount;  
9  console.log(func());
```

答案：

this指向的是最后调用他的对象，第一次this指向的是baidu，所以count为2，但是当var func的时候this指向了window，而又不存在全局的count，所以是undefined

4、下面代码输出的结果为：

```
1  function Foo() {  
2      var i = 0;  
3      return function() {  
4          console.log(i++);  
5      }  
6  }  
7  var f1 = Foo(),  
8  f2 = Foo();  
9  f1();  
10 f1();  
11 f2();
```

答案：

闭包中的局部变量的值会一直保存在同一个引用下。所以第一次f1调用输出i为0，后i++。第二次调用f1时i=1所以输出1。

5、dom 的操作，常用的有哪些，如何创建、添加、移除、移动、复制、查找节点？

答案：

创建：

```
createDocumentFragment() //创建一个DOM片段
createElement() //创建一个具体的元素
createTextNode() //创建一个文本节点
```

添加：

```
appendChild()
```

移出：

```
removeChild()
```

替换：

```
replaceChild()
```

插入：

```
insertBefore()
```

复制：

```
cloneNode(true)
```

查找：

```
getElementsByName() //通过标签名称
getElementsByClassName() //通过标签名称
getElementsByName() //通过元素的Name属性的值
getElementById() //通过元素Id，唯一性
```

6、javascript语言的执行环境是”单线程模式”，这种模式的好处是实现起来比较简单，执行环境相对单纯；坏处是只要有一个任务耗时很长，后面的任务都必须排队等着，会拖延整个程序的执行，因此很多时候需要进行“异步模式”，请列举js异步编程的方法。

答案：

JavaScript异步编程的四种方法

1.回调函数

```
f1(f2);
```

回调函数是异步编程的基本方法。其优点是易编写、易理解和易部署；缺点是不利于代码的阅读和维护，各个部分之间高度耦合（Coupling），流程比较混乱，而且每个任务只能指定一个回调函数。

2.事件监听

```
f1.on('done',f2);
```

事件监听即采用事件驱动模式，任务的执行不取决于代码的顺序，而取决于某个事件是否发生。其优点是易理解，可以绑定多个事件，每个事件可以指定多个回调函数，可以去耦合，有利于实现模块化；缺点是整个程序都要变成事件驱动型，运行流程会变得不清晰。

3. 发布/订阅

```
f1: jQuery.publish("done");
```

```
f2: jQuery.subscribe("done", f2);
```

假定存在一个“信号中心”，某个任务执行完成，就向信号中心“发布”（publish）一个信号，其他任务可以向信号中心“订阅”（subscribe）这个信号，从而知道什么时候自己可以开始执行，这就叫做“发布/订阅模式”（publish-subscribe pattern），又称“观察者模式”（observer pattern）。该方法的性质与“事件监听”类似，但其优势在于可以通过查看“消息中心”，了解存在多少信号、每个信号有多少订阅者，从而监控程序的运行。

4.promise对象

```
f1().then(f2);
```

Promises对象是CommonJS工作组提出的一种规范，目的是为异步编程提供统一接口；思想是，每一个异步任务返回一个Promise对象，该对象有一个then方法，允许指定回调函数。其优点是回调函数是链式写法，程序的流程非常清晰，而且有一整套的配套方法，可以实现许多强大的功能，如指定多个回调函数、指定发生错误时的回调函数，如果一个任务已经完成，再添加回调函数，该回调函数会立即执行，所以不用担心是否错过了某个事件或信号；缺点就是编写和理解相对比较难。