**H5方向第二个月面试题库**

**1**、请描述一下**cookies**，**sessionStorage**和**localStorage**的区别？

sessionStorage用于本地存储一个会话（session）中的数据，这些数据只有在同一个会话中的页面才能访问并且当会话结束后数据也随之销毁。因此sessionStorage不是一种持久化的本地存储，仅仅是会话级别的存储。而localStorage用于持久化的本地存储，除非主动删除数据，数据是永远不会过期的。

web storage和cookie的区别

Web Storage的概念和cookie相似，区别是它是为了更大容量存储设计的。Cookie的大小是受限的，并且每次你请求一个新的页面的时候Cookie都会被发送过去，这样无形中浪费了带宽，另外cookie还需要指定作用域，不可以跨域调用。

除此之外，Web Storage拥有setItem,getItem,removeItem,clear等方法，不像cookie需要前端开发者自己封装setCookie，getCookie。但是Cookie也是不可以或缺的：Cookie的作用是与服务器进行交互，作为HTTP规范的一部分而存在 ，而Web Storage仅仅是为了在本地“存储”数据而生。

学员从失效时间和存储大小来回答：sessionStorage页面关闭及失效、cookies设置过期时间、localStorage永久不失效，除非手动删除

**2**、一个页面上有大量的图片（大型电商网站），加载很慢，你有哪些方法优化这些图片的加载，给用户更好的体验。

图片懒加载，在页面上的未可视区域可以添加一个滚动条事件，判断图片位置与浏览器顶端的距离与页面的距离，如果前者小于后者，优先加载。

如果为幻灯片、相册等，可以使用图片预加载技术，将当前展示图片的前一张和后一张优先下载。

如果图片为css图片，可以使用CSSsprite，SVGsprite，Iconfont、Base64等技术。

如果图片过大，可以使用特殊编码的图片，加载时会先加载一张压缩的特别厉害的缩略图，以提高用户体验。

如果图片展示区域小于图片的真实大小，则因在服务器端根据业务需要先行进行图片压缩，图片压缩后大小与展示一致。

此题比较灵活:图片懒加载或者预加载或者图片压缩等，例如学员回答图片预加载，此时要问一下图片预加载怎么实现，答出来即得满分

**3**、**split() join()** 的区别

**split()**是切割成数组的形式

**join()**是将数组转换成字符串

以上两个回答正确得满分

**4**、事件绑定和普通事件有什么区别

传统事件绑定和符合W3C标准的事件绑定有什么区别？

div1.onclick=function(){};

<button onmouseover=””></button>

1、如果说给同一个元素绑定了两次或者多次相同类型的事件，那么后面的绑定会覆盖前面的绑定

2、不支持DOM事件流事件捕获阶段目标元素阶段

addEventListener

1、 如果说给同一个元素绑定了两次或者多次相同类型的事件，所以的绑定将会依次触发

2、 支持DOM事件流的

3、 进行事件绑定传参不需要on前端

addEventListener(“click”,function(){},true);//此时的事件就是在事件冒泡阶段执行

ie9开始，ie11 edge：addEventListener

ie9以前：attachEvent/detachEvent

1、 进行事件类型传参需要带上on前缀

2、 这种方式只支持事件冒泡，不支持事件捕获

事件绑定是指把事件注册到具体的元素之上，普通事件指的是可以用来注册的事件

学员从两个角度去说：

1、不能绑定多次，例如说给同一个元素绑定了两次或者多次相同类型的事件，那么后面的绑定会覆盖前面的绑定

2、on的方式不支持事件捕获，不支持DOM事件流事件捕获阶段目标元素阶段

**5**、在**JS**中都有哪些兼容性的写法？

获取事件对象 ev = ev || window.event

滚轮事件(mousewheel/DOMMouseScroll)

事件挂接(addEventListener/attachEvent)

冒泡(stopPropagation/cancelBubble = true)

获取内联样式（currentStyle\getComputedStyle）

获取滚动条高度：document.documentElement.scrollTop || document.body.scrollTop

引入requestAnimationFrame.js

var target = ev.srcElement || ev.target

该题比较灵活，学员说出上面常见的三个及得满分（最少三个）

**6**、事件委托是什么

符合W3C标准的事件绑定addEventLisntener /attachEvent

利用事件冒泡的原理，自己所触发的事件让他的父元素代替执行！

学员根据自己的理解去解释，说出通过事件冒泡这个原理得满分

**7**、如何阻止事件冒泡和默认事件

Event.stopPropagation();//标准浏览器

event.canceBubble=true;//ie9之前

阻止默认事件：

为了不让a点击之后跳转，我们就要给他的点击事件进行阻止

return false

e.preventDefault();

事件冒泡两种方式必须答出，取消默认事件两种方式必须答出

**8**、添加删除替换插入到某个接点的方法

obj.appendChild()

obj.insertBefore() //原生的js中不提供insertAfter();

obj.replaceChild()//替换

obj.removeChild()//删除

上述说出三个方法得满分

**9**、**“==”**和**“===”**的不同

==：判断值是否相同，忽略数据类型

===：判断值和数据类型是否相同

**10**、看下列代码，将会输出什么**?(**变量声明提升**)**

var foo = 1;

(function( ){

    console.log(foo);

    var foo = 2;

    console.log(foo);

})( )

答案：输出undefined 和2。

函数声明与变量声明会被JavaScript引擎隐式地提升到当前作用域的顶部，但是只提升名称不会提升赋值部分。

**11**、用**js**实现随机选取**10–100**之间的**10**个数字，存入一个数组，并排序。

var iArray = [];

function getRandom(istart, iend){

        var iChoice = istart - iend +1;

        return Math.floor(**Math.random()** \* iChoice + istart;

}

for(var i=0; i<10; i++){

var result= getRandom(10,100);

        iArray.push(result);

}

**iArray.sort();**

改题比较灵活，让学员说出思路得满分，主要考察随机函数和数组中**sort**方法

**12**、把两个数组合并，并删除第二个元素。

var array1 = ['a','b','c'];

var bArray = ['d','e','f'];

var cArray = array1.**concat**(bArray);

cArray.splice(1,1);

改题比较灵活，让学员说出思路得满分，主要考察数组中**concat**方法和**splice**方法

**13**、怎样添加、移除、移动、复制、创建和查找节点

1）创建新节点

createDocumentFragment()    //创建一个文档碎片

createElement()   //创建一个具体的元素

createTextNode()   //创建一个文本节点

2）添加、移除、替换、插入

appendChild()      //添加

removeChild()      //移除

replaceChild()      //替换

insertBefore()      //插入

3）查找

getElementsByTagName()    //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

getElementsByClassName()

document.querySelector(“.div1”)  
document.querySelectorAll(“.div1”)

上述问题主要考察的是对标签节点的操作，增删改查回答出来得满分

**14**、匹配邮箱的正则表达式？

var regMail = /^([a-zA-Z0-9\_-])+@([a-zA-Z0-9\_-])+((.[a-zA-Z0-9\_-]{2,3}){1,2})$/;

该题答案比较灵活，匹配规则不同代码不同，主要考察正则的熟练程度

**15**、看下面代码，输出结果是什么并解释原因？

for(var i=1;i<=3;i++){

  setTimeout(function(){

      console.log(i);

  },0);

};

答案：4 4 4。

原因：Javascript事件处理函数是回调函数，回调函数在线程空闲之前不会运行。

**16**、列举浏览器对象模型**BOM**里常用的至少**4**个对象，并列举**window**对象的常用方法至少**3**个

对象：Window 、location、 history 、navigator

方法：Alert() confirm() prompt() open() close()

**17**、把**Script** 标签放在页面的最底部的**body**封闭之前和封闭之后有什么区别？浏览器会如何解析它们？(答案是什么？)

如果说放在body的封闭之前，将会阻塞其他资源的加载

如果放在body封闭之后，不会影响body内元素的加载

**18**、请你谈谈**Cookie**的弊端？

缺点：

1.`Cookie`数量和长度的限制。每个domain最多只能有20条cookie，每个cookie长度不能超过4KB，否则会被截掉。

2.安全性问题。如果cookie被人拦截了，就可以取得所有的session信息。即使加密也于事无补，因为拦截者并不需要知道cookie的意义，他只要原样转发cookie就可以达到目的了。

3.有些状态不可能保存在客户端。例如，为了防止重复提交表单，我们需要在服务器端保存一个计数器。如果我们把这个计数器保存在客户端，那么它起不到任何作用。

学员从cookie的大小安全方面考虑即可满分

**19**、生成**5**个不同的随机数；

//思路：5个不同的数，每生成一次就和前面的所有数字相比较，如果有相同的，则放弃当前生成的数字！

var num1 = [];

for(var i = 0; i < 5; i++){

    num1[i] = Math.floor(Math.random()\*10) + 1; //范围是 [1, 10]

    for(var j = 0; j < i; j++){

        if(num1[i] == num1[j]){

            i--;

        }

    }

}

该题要求学员手写，此题比较灵活考察学员逻辑。只要结果相同即可

**20、如何阻止冒泡函数，两种写法**

e.cancelBubble = true

e.stopPropagation();

**21**、阻止默认事件的两种方式

**return false**

**ev.preventDefault()**

**22、**使用**RegExp**找出字符串所有的数字，并以数组的形式返回

var reg = /\d/g;

console.log(str.match(reg));

**23**、从数组中找出最小的数，并且输出其下标

var arr = [34,25,324,345,14,34634,73,1,624,7,3423,7,456,31,0,41];

var min = arr[0];

var index = 0;

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

if(min > arr[i]){

//code...

min = arr[i];

index = i;

}

}

alert(index);

该题考察学员的逻辑，此题没有标准答案，只要是结果相同即可。要求学员说出思路或者手写

**24**、以下程序运行结果是什么？并解释原因。

var a = 3;

var b = 4;

var c = 5;

console.log(a>b&&c>b||a<c);

A.1 B.true C.false D.报错

答案是B

**25**、以下程序运行结果是什么？并解释原因。

var a = 3;

var b = 4;

var c = 5;

console.log(a>b?a:c);

A.3 B.4 C.5 D.报错

答案C，主要考察的三元运算符

**26**、以下代码**alert**的值分别是什么并解释原因。

var a = 100;

function test(){

alert(a);

var a = 10;

alert(a);

}

test();

A、100 ，10 B、undefined，10 C、10，10 D、报错

答案B、主要考察执行环境，作用域，作用域链，变量提升（hositing），说出变量提升则满分

**27**、以下代码**alert**的值分别是什么并解释原因。

var a = 100;

function test(){

alert(a);

a = 10;

alert(a);

}

test();

alert(a);

A.100,10,10 B.100,10,100 C.10,10,100 D.10,10,10

答案A、主要考察执行环境，作用域，作用域链，变量提升（hositing），说出变量提升则满分

**28**、下面程序的运行结果什么并解释原因。

for(i=0, j=0; i<10, j<6; i++, j++){

k = i + j;

}

console.log(k);

A.10 B.6 C.16 D.60

答案A、主要考察for循环里面如果有两个判断条件则按照最后一个条件执行