# 临毕业摸底测验（第一部分）

学习笔记

## JavaScript （前端玩家必备技能）

1. **ele.getAttribute('propName') 和 ele.propName区别**

    - ele.getAttribute('propName') 是获取ele元素的Dom节点上的propName属性的属性值，当前Dom节点上存在这个属性和属性值，是看得见的；
    - ele.propName是获得ele对象上的propName属性的属性值，这个属性属性值是ele对象上的一个键值对，春- 在于ele的堆内存中，是抽象的看不见的

2. **mouseover和mouseenter的区别**

    - mouseover存在事件冒泡机制（进入父元素里面的子元素会频繁的触发其子孙元素的mouseover事件，耗性能）
    - mouseenter不存在事件冒泡机制（进入到父元素中触发一次mouseenter事件，再进到子元素里的时候不再触发，全部处理为父元素内部）

3. **什么是事件代理**

    - 由于事件冒泡机制，在浏览器上我们点击任何一个div，他都会有一个传播路径：div->body->html->document->window,所以当我们触发div的任意一个方法的时候都会由于事件冒泡机制传递给他的父级元素一直到window为止，有时我们利用这种机制，直接给document绑定一个事件，当有很多个div的时候我们点击任意一个都可以执行相对应的方法，而不必给每一个div都绑定一个方法，这样即避免代码冗余，也可以为浏览器节省一丢丢性能，我们称这种做法为事件代理；

4. **localStorage和cookie的区别，cookie和session的关系！**

    - localStorage是在本地浏览器的缓存中一直存在的，除非用户自己手动删除，而cookie是在url的请求中，发送请求和返回相应的时候都要带着cookie，他一直在浏览器和服务器之间传来传去，所以我们对cookie的大小有限制，一般最大4KB，一个浏览器最多只能有20个cookie；
    - session是服务端保留的一个数据结构，用来跟踪用户的状态，这个数据可以保留数据库中，我们一般根据在session里面设置的一些id值什么的来辨别用户；其中session的这个标识是通过cooKies传递过去的，所以cookie是实现session的一种手段；

5. **什么是闭包，你在项目中哪一块用到了闭包！**

    - 函数运行的时候形成一个独立的作用域，保护里面的私有属性和方法不受外界的干扰，我们称这种机制为闭包；
    - 闭包具有保存和保护作用，保存是保护方法里面的私有属性和方法不受外界的干扰，保护是保护被外界引用的函数不被销毁，也就是形成一个不销毁的作用域，供外界使用，因为这样所以理想情况下我们是避免使用闭包的

- 但是真实项目中我们往往为了避免自己的变量造成全局污染而选择用一个闭包将自己的js代码包起来，或者使用高级单例的时候也是利用了闭包的机制来实现复杂的业务逻辑，还有封装插件的时候为了防止我们使用的变量和用户自己程序上的冲突，往往会使用闭包将自己的插件保护起来，如JQ和Zepto这些插件都是这么用的，另外就是一些高级语法，比如柯力化函数思想等也都是利用闭包思想实现的；

6. **js中定义函数的方式有哪些，区别是什么！**
   定义函数的方式：

   - function fn() {...}这种定义的方法是可以申明的，也就意味这我们可以在任何地方直接执行fn，不具有严谨性
   - let fn = function(){...} 这种要使用fn必须在let fn之后才能使用，稍微严禁一些
   - ;(function(){...})()自执行函数

7. **说出你掌握的继承方式及优缺点，并加以改进！**

   - **原型继承**（比如B想继承A的原型上的方法，就让B指向A的一个实例，这样B就可以通过A的实例获取到A原型上的共有属性和方法，但是缺点是，这样A的实例上的私有属性就也暴露了）
   - **寄生组合式继承**（这个是让A的原型指向B的原型，这样A的原型里面就B的原型所有的方法和属性了，然后B私有的属性方法，我们可以通过A.call(this)来把B传到A里面，从而实现使用A的方法）
   - **class继承**（这个是ES6新出的比较完善的继承方式，继承方式如下：）

```
class A {
 constructor () {//A私有属性和方法}
fn(){//公有属性和方法}
}
class B  extends A{
    constructor(){
        super();
        //B私有的属性和方法;
    }
}
```

8. 说出ES6和ES5的区别！

   - **let和var**的区别： let会形成一个块级作用域，而且不存在变量提升，不允许重复声明，并且切断了与window的映射关系；
   - typeof一个未被声明的变量会报错，在之前ES5中只是得到的结果是undefined；
   - **ES6的解构赋值，模板字符串**
   - **箭头函数(this&&arguments)**ES6中的箭头函数中不存在this，他里面使用的this都是上下文中的this，而且在ES6中arguments跟形参之间的映射关系也不存在了；
   - **class继承**
   - ****for…of(interator接口)
   - 还有给Array和String，Math等添加了一些方便的属性方法供我们使用；
   - **Promise，这个用来管控异步特别方便，在项目中用的非常多；**
   - set和Map，用set来做数组去重特别方便；

9. **阐述JS中的同步编程和异步编程，以及你在项目中是如何来使用异步操作的！**

因为js是单线程的，所以在同一时间内只能执行一件事情，为了同时做一件事情不阻塞其他事情的操作，js产生了同步和异步两种方式来管理我们的操作，js中的异步操作有以下几种：

- 定时器的操作
- 事件绑定
- Ajax
- 回调函数

在我们的的项目中，我们一般使用Promise来管理异步，我们通常让一个异步操作返回一个Promise实例，然后把后面想要做的事情放到Promise的then方法中，当返回的实例的状态达到条件是运行相对应的方法，这样管控既方便逻辑也很清晰，使得异步操作变得非常简单！

4. 实现一个Promise

```
let promise = new Promise((resolve,reject)=>{
    resolve();
});
promise.then(res=>{
    //成功之后传参res
}).catch(err=>{
    //失败之后捕捉到失败的信息
})
```

# HTTP && AJAX && 跨域 （18+玩家必备技能，初级玩家需要了解一些的）

1. **写出项目中经常用到的性能优化方案**

    - CSS，js，图片压缩合并，雪碧图
    - 懒加载，首屏加载
    - 304缓存
    - 使用iconfont字体图标
    - 使用locaLStorage做本地缓存
    - 减少回流和重绘，分离读写
    - 实现js代码的高耦合低内聚

2. **从浏览器地址栏输入URL到显示页面，中间都经历了什么（尽可能写详细，最好回答出TCP的三次握手和四次挥手，以及浏览器加载页面的细节）**

    - 1. 输入网址，浏览器向DNS发送请求；
      2. DNS收到请求网址，进行反解析，找到这个网址对应的外网服务器的IP地址
      3. 根据IP地址以及端口找到对应服务器的对应的项目
      4. 服务器将找到的项目的源代码返回给客户端浏览器；
      5. 浏览器收到源代码，开始自上而下执行，遇到css外链，js，图片地址请求等都会重新执行上面的请求步骤，直到所有的请求都发送完毕，所有的代码都返回到

浏览器

6. 浏览器开始根据自己的内核渲染拿到的代码，首先生成render Tree，然后加载css，生成render TRee，最后生成成完成页面；
    - 在此期间，浏览器跟服务之间的发送请求和接送请求也涉及到很多的步骤，其中就包括经典的三次握手和四次挥手：
    - 浏览器和服务器建立连接的时候：1.浏览器向服务器发送一个建立连接的请求；2.服务器收到请求，同意建立连接并且反馈给浏览器响应；3.浏览器发送给服务器确认收到响应的请求
    - 服务器和浏览器断开链接是一般有四次对话：1. 浏览器发送给服务器想断开连接的请求；2. 服务器收到浏览器的请求；3. 发送给浏览器同意断开请求的响应；4.浏览器断开链接，并且发送给服务器已经断开连接的请求；

3. **说出你所熟知的HTTP状态码！GET和POST有啥区别！**

    - http状态码：
    - 200 （成功）
    - 301（永久转移）
    - 302（临时转移）
    - 307（临时重定向）
    - 304（缓存）
    - 400（参数不对）
    - 401（没有权限）
    - 404（找不到）
    - 413（客户端发送请求过大，服务器拒绝处理）
    - 500（服务器错误）
    - 502（服务器繁忙，当前请求需要排队）
    - get和POST传参的方式不同get是url传参而POST是直接在请求主体中发送参数
    - POST相对安全，get请求的地址很容易被黑客劫持，不安全；
    - get有缓存，POST没有，所以一般我们频繁发送get请求而且想得到每一次都不一样的值就在get后面穿一个时间戳的参数；

4. **什么是HTTP报文，你熟知的报文都有哪些！**

    - http报文就是浏览器发送请求的时候携带的一些请求头，响应头，通用头，请求主体，响应主体等
    - 熟知的报文有：通用头:request Url/request Method/status code/requestHeaders/responseHeaders

5. **能说下304具体怎样实现吗？**
6. **跨域是什么？http协议中如何判断跨域？如何解决跨域问题？**

    - 浏览器之间的通信需要同源，也就是协议，域名和端口都要相同才能通信，

7. **HTTP2具体内容？SDPY了解么？**
8. **HTTPS如何实现？tsl/ssl是什么？对称加密、非对称加密在什么时候、对什么数据加密？**
9. **DNS劫持是什么？**

    - DNS劫持就是浏览器输入域名向服务器发送请求的时候，第三方劫持DNS服务器，将请求转换到另一台伪DNS上或者直接抓到域名返回一个错误的IP，导致域名的请求错

误或者请求到另一个服务器的IP上，这就是DNS劫持吧。。。

10. **封装一个AJAX库！**