1, 判断字符串是否是这样组成的，第一个必须是字母，后面可以是字母、数字、下划线，总长度为 5-20

```
var reg = /^[a-zA-Z][a-zA-Z_0-9]{4,19}$/;//定义 RegExp 对象，大括号表示重复次数 4-19 次
reg.test("a1a__a1a__a1a__a1a__");//检查一个字符串中是否存在创建 RegExp 对象实例时所指定的表达式模式，true/false
```

2，截取字符串 abcdefg 的 efg

```
var str = "abcdefg";
if (/efg/.test(str)) {
 var efg = str.substr(str.indexOf("efg"), 3);//注意与 substring 的区别，substr(index,length)参数为子串开始位置，长度 alert(efg);
}
```

3，判断一个字符串中出现次数最多的字符，统计这个次数

方法 1：

```
//将字符串的字符保存在一个 hash table 中，key 是字符，value 是这个字符出现的次数
var str = "abcdefgaddda";
var obj = {};
for (var i = 0, l = str.length; i < l; i++) {
   var key = str[i];
   if (!obj[key]) {
      obj[key] = 1;
   } else {
      obj[key]++;
   }
}
/*遍历这个 hash table，获取 value 最大的 key 和 value*/
var max = -1;
var max_key = "";
var key;
for (key in obj) {
   if (max < obj[key]) {
      max = obj[key];
      max_key = key;
   }
}
alert("max:"+max+" max_key:"+max_key);
```

方法 2：

```
var aa='121321';
var str =aa.split('');//将字符串转换为字符数组
var m={},k=0;//k 记录重复次数
for(var i=0;i<str.length;i++){
     for(var j=0;j<str.length;j++){
         if(str[i]==str[j])
             k++;
     }
     var x=str[i];
     m[x]=k;
     k=0
};
console.log(m)//Object {1: 3, 2: 2, 3: 1}
```

4，IE 与 FF 脚本兼容性问题

(1) window.event：

表示当前的事件对象，IE 有这个对象，FF 没有，FF 通过给事件处理函数传递事件对象，比如 check(e){}//e 为事件对象

(2) 获取事件源

IE 用 srcElement 获取事件源，而 FF 用 target 获取事件源

在 IE 下,event 对象有 **srcElement** 属性,但是没有 target 属性;
在 Firefox 下,event 对象有 **target** 属性,但是没有 srcElement 属性.

```
obj = event.srcElement ? event.srcElement : event.target;
```

(3) 添加，去除事件

IE：

```
element.attachEvent(“onclick”, function)
element.detachEvent(“onclick”, function)
```

FF：

```
element.addEventListener(“click”, function, true)
element.removeEventListener(“click”, function, true)
```

(4) 获取标签的自定义属性

IE： **div1.value 或 div1[“value”]**

FF：可用 **div1.getAttribute(“value”)**

(5) document.getElementByName()和 document.all[name]//页面所有元素的集合

IE：document.getElementByName()和 document.all[name]均不能获取 div 元素

FF：可以

(6) input.type 的属性

IE：input.type 只读

FF：input.type 可读写

(7) innerText textContent outerHTML

IE：支持 innerText, outerHTML

FF：支持 textContent

(8) 是否可用 id 代替 HTML 元素

IE：可以用 id 来代替 HTML 元素

FF：不可以

这里只列出了常见的，还有不少，更多的介绍可以参看 JavaScript 在 IE 浏览器和 Firefox 浏览器中的差异总结

5，规避 javascript 多人开发函数重名问题

(1) 可以开发前规定命名规范， 根据不同开发人员开发的功能在函数前加前缀

(2) 将每个开发人员的 函数封装到类中，调用的时候就调用类的函数，即使函数重名只要类名不重复就 ok

6，javascript 面向对象中继承实现

javascript 面向对象中的继承实现一般都使用到了构造函数和 Prototype 原型链，简单的代码如下：

```
function Animal(name) {
    this.name = name;
}
Animal.prototype.getName = function() {alert(this.name)}
function Dog() {};
Dog.prototype = new Animal("Buddy");
Dog.prototype.constructor = Dog;
var dog = new Dog();
```

7，FF 下面实现 outerHTML

FF 不支持 outerHTML，要实现 outerHTML 还需要特殊处理

思路如下：

在页面中添加一个新的元素 A，克隆一份需要获取 outerHTML 的元素，将这个元素 append 到新的 A 中，然后获取 A 的 innerHTML 就可以了。

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312"
/>
<title>获取 outerHMTL</title>
<style>
  div{ background:#0000FF;width:100px;height:100px;}
  span{ background:#00FF00;width:100px;height:100px;}
  p{ background:#FF0000;width:100px;height:100px;}
</style>
</head>
<body>
  <div id="a"><span>SPAN</span>DIV</div>
  <span>SPAN</span>
  <p>P</p>
<script type="text/javascript">
  function getOuterHTML(id){
    var el = document.getElementById(id);
    var newNode = document.createElement("div");
    document.appendChild(newNode);
    var clone = el.cloneNode(true);
    newNode.appendChild(clone);
    alert(newNode.innerHTML);
    document.removeChild(newNode);
}
getOuterHTML("a");
</script>
</body>
</html>
```

8，编写一个方法 求一个字符串的字节长度

假设：

一个英文字符占用一个字节，一个中文字符占用两个字节

```
function GetBytes(str){
    var len = str.length;
    var bytes = len;
    for(var i=0; i<len; i++){
        if (str.charCodeAt(i) > 255) bytes++;
    }
    return bytes;
}
alert(GetBytes("你好,as"));
```

9，编写一个方法 去掉一个数组的重复元素

```
var arr = [1 ,1 ,2, 3, 3, 2, 1];
Array.prototype.unique = function(){
    var ret = [];
    var o = {};
    var len = this.length;
    for (var i=0; i<len; i++){
        var v = this[i];
        if (!o[v]){
            o[v] = 1;
            ret.push(v);
        }
    }
    return ret;
};
alert(arr.unique());
```

10，写出 3 个使用 this 的典型应用

（1）在 html 元素事件属性中使用，如

```
<input type=”button” onclick=”showInfo(this);” value=”点击一下”/>
```

（2）构造函数

```
function Animal(name, color) {
    this.name = name;
    this.color = color;
}
```

（3）

```
<input type="button" id="text" value="点击一下" />
<script type="text/javascript">
var btn = document.getElementById("text");
btn.onclick = function() {
alert(this.value); //此处的 this 是按钮元素
}
</script>
```

（4）CSS expression 表达式中使用 this 关键字

```
<table width="100px" height="100px">
  <tr>
    <td>
    <div style="width:expression(this.parentNode.width);">div element</div>
    </td>
  </tr>
</table>
```

12，如何显示/隐藏一个 DOM 元素？

```
el.style.display = "";
el.style.display = "none";
```

el 是要操作的 DOM 元素

13，JavaScript 中如何检测一个变量是一个 String 类型？请写出函数实现

String 类型有两种生成方式：

```
(1)Var str = “hello world”;
```

(2)Var str2 = new String(“hello world”);

```
function IsString(str){
    return (typeof str == "string" || str.constructor == String);
}
var str = "";
alert(IsString(1));
alert(IsString(str));
alert(IsString(new String(str)));
```

14，网页中实现一个计算当年还剩多少时间的倒数计时程序，要求网页上实时动态显示“××年还剩××天××时××分××秒”

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
   <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
   <title>倒计时</title>
</head>
<body>
<input type="text" value="" id="input" size="1000"/>
<script type="text/javascript">
   function counter() {
      var date = new Date();
      var year = date.getFullYear();
      var date2 = new Date(year, 12, 31, 23, 59, 59);
      var time = (date2 - date)/1000;
      var day =Math.floor(time/(24*60*60))
      var hour = Math.floor(time%(24*60*60)/(60*60))
      var minute = Math.floor(time%(24*60*60)%(60*60)/60);
      var second = Math.floor(time%(24*60*60)%(60*60)%60);
      var str = year + "年还剩"+day+"天"+hour+"时"+minute+"分
"+second+"秒";
      document.getElementById("input").value = str;
   }
   window.setInterval("counter()", 1000);
</script>
</body>
</html>
```

15，补充代码，鼠标单击 Button1 后将 Button1 移动到 Button2 的后面

```
<div> <input type=” button” id =” button1″ value=” 1″
onclick=” ???” > <input type=” button” id =” button2″ value=” 2″
/” > </div>
```

```
<div>
   <input type="button" id ="button1" value="1"
onclick="moveBtn(this);">
   <input type="button" id ="button2" value="2" />
</div>
<script type="text/javascript">
function moveBtn(obj) {
   var clone = obj.cloneNode(true);
   var parent = obj.parentNode;
   parent.appendChild(clone);
   parent.removeChild(obj);
}
</script>
```

16，JavaScript 有哪几种数据类型

简单：Number，Boolean，String，Null，Undefined

复合：Object，Array，Function

17，下面 css 标签在 JavaScript 中调用应如何拼写，border-left-color，-moz-viewport

borderLeftColor

mozViewport

18，JavaScript 中如何对一个对象进行深度 clone

```
function cloneObject(o) {
```

```
    if(!o || 'object' !== typeof o) {
        return o;
    }
    var c = 'function' === typeof o.pop ? [] : {};
    var p, v;
    for(p in o) {
        if(o.hasOwnProperty(p)) {
            v = o[p];
            if(v && 'object' === typeof v) {
                c[p] = Ext.ux.clone(v);
            }
            else {
                c[p] = v;
            }
        }
    }
    return c;
};
```

19，如何控制 alert 中的换行

alert(“p\np”);

20，请实现，鼠标点击页面中的任意标签，alert 该标签的名称.（注意兼容性）

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312"
/>
<title>鼠标点击页面中的任意标签，alert 该标签的名称</title>
<style>
div{ background:#0000FF;width:100px;height:100px;}
span{ background:#00FF00;width:100px;height:100px;}
p{ background:#FF0000;width:100px;height:100px;}
</style>
<script type="text/javascript">
document.onclick = function(evt){
 var e = window.event || evt;
 var tag = e["target"] || e["srcElement"];
 alert(tag.tagName);
};
</script>
</head>
<body>
<div id="div"><span>SPAN</span>DIV</div>
<span>SPAN</span>
<p>P</p>
</body>
</html>
```

21，请编写一个 JavaScript 函数 parseQueryString，它的用途是把 URL 参数解析为一个对象，如：

var url = “http://witmax.cn/index.php?key0=0&key1=1&key2=2″;

```
function parseQueryString(url){
   var params = {};
   var arr = url.split("?");
   if (arr.length <= 1)
      return params;
   arr = arr[1].split("&");
   for(var i=0, l=arr.length; i<l; i++){
      var a = arr[i].split("=");
      params[a[0]] = a[1];
   }
   return params;
}
var url = "http://witmax.cn/index.php?key0=0&key1=1&key2=2";
var ps = parseQueryString(url);
alert(ps["key1"]);
```

22，ajax 是什么? ajax 的交互模型？同步和异步的区别? 如何解决跨域问题?

Ajax 是多种技术组合起来的一种浏览器和服务器交互技术，基本思想是允许一个互联网浏览器向一个远程页面/服务做异步的 http 调用，并且用收到的数据来更新一个当前 web 页面而不必刷新整个页面。该技术能够改进客户端的体验。包含的技术：

XHTML：对应 W3C 的 XHTML 规范，目前是 XHTML1.0。

CSS：对应 W3C 的 CSS 规范，目前是 CSS2.0

DOM：这里的 DOM 主要是指 HTML DOM，XML DOM 包括在下面的 XML 中

JavaScript：对应于 ECMA 的 ECMAScript 规范

XML：对应 W3C 的 XML DOM、XSLT、XPath 等等规范

XMLHttpRequest：对应 WhatWG 的 Web Applications1.0 规范（http://whatwg.org/specs/web-apps/current-work/）

AJAX 交互模型

同步：脚本会停留并等待服务器发送回复然后再继续

异步：脚本允许页面继续其进程并处理可能的回复

跨域问题简单的理解就是因为 JS 同源策略的限制，a.com 域名下的 JS 无法操作 b.com 或 c.a.com 下的对象，具体场景如下：

PS：（1）如果是端口或者协议造成的跨域问题前端是无能为力的

(2) 在跨域问题上，域仅仅通过 URL 的首部来识别而不会尝试判断相同的 IP 地址对应的域或者两个域是否对应一个 IP

前端对于跨域的解决办法：

(1) document.domain+iframe

(2) 动态创建 script 标签

23，什么是闭包？下面这个 ul，如何点击每一列的时候 alert 其 index?

```
<ul id=”test”>
<li>这是第一条</li>
<li>这是第二条</li>
<li>这是第三条</li>
</ul>
```

内部函数被定义它的函数的外部区域调用的时候就产生了闭包。

```
(function A() {
   var index = 0;
   var ul = document.getElementById("test");
   var obj = {};
   for (var i = 0, l = ul.childNodes.length; i < l; i++) {
      if (ul.childNodes[i].nodeName.toLowerCase() == "li") {
         var li = ul.childNodes[i];
         li.onclick = function() {
            index++;
            alert(index);
         }
      }
   }
})();
```

24，请给出异步加载 js 方案，不少于两种

默认情况 javascript 是同步加载的，也就是 javascript 的加载时阻塞的，后面的元素要等待 javascript 加载完毕后才能进行再加载，对于一些意义不是很大的 javascript，如果放在页头会导致加载很慢的话，是会严重影响用户体验的。

异步加载方式：

(1) defer，只支持 IE

(2) async：

(3) 创建 script，插入到 DOM 中，加载完毕后 callBack，见代码：

```
function loadScript(url, callback){
   var script = document.createElement("script")
   script.type = "text/javascript";
   if (script.readyState){ //IE
      script.onreadystatechange = function(){
         if (script.readyState == "loaded" ||
            script.readyState == "complete"){
            script.onreadystatechange = null;
            callback();
         }
```

```
        };
    } else { //Others: Firefox, Safari, Chrome, and Opera
        script.onload = function(){
            callback();
        };
    }
    script.src = url;
    document.body.appendChild(script);
}
```

25，请设计一套方案，用于确保页面中 JS 加载完全。

```
var n = document.createElement("script");
n.type = "text/javascript";
//以上省略部分代码
//ie 支持 script 的 readystatechange 属性(IE support the
readystatechange event for script and css nodes)
if(ua.ie){
    n.onreadystatechange = function(){
        var rs = this.readyState;
        if('loaded' === rs || 'complete'===rs){
            n.onreadystatechange = null;
            f(id,url); //回调函数
        }
};
//省略部分代码
//safari 3.x supports the load event for script nodes(DOM2)
    n.addEventListener('load',function(){
        f(id,url);
    });
//firefox and opera support onload(but not dom2 in ff) handlers for
//script nodes. opera, but no ff, support the onload event for link
//nodes.
}else{
    n.onload = function(){
        f(id,url);
    };
}
```

26，js 中如何定义 class,如何扩展 prototype?

Ele.className = “***”; //***在 css 中定义，形式如下：.*** {…}

A.prototype.B = C;

A 是某个构造函数的名字

B 是这个构造函数的属性

C 是想要定义的属性的值

27，如何添加 html 元素的事件,有几种方法.

（1） 为 HTML 元素的事件属性赋值

（2） 在 JS 中使用 ele.on*** = function() {…}

（3） 使用 DOM2 的添加事件的方法 addEventListener 或 attachEvent

28，documen.write 和 innerHTML 的区别

document.write 只能重绘整个页面

innerHTML 可以重绘页面的一部分

29，多浏览器检测通过什么？

（1） navigator.userAgent

（2） 不同浏览器的特性，如 addEventListener

30，js 的基础对象有那些, window 和 document 的常用的方法和属性列出来

String,Number,Boolean

Window:

方法：setInterval,setTimeout,clearInterval,clearTimeout,alert,confirm,open

属性：name,parent,screenLeft,screenTop,self,top,status

Document

方法：
createElement,execCommand,getElementById,getElementsByName,getElementByTagName,write,writeln

属性：cookie,doctype,domain,documentElement,readyState,URL,

31，前端开发的优化问题

（1） 减少 http 请求次数：css spirit,data uri

（2） JS，CSS 源码压缩

（3） 前端模板 JS+数据，减少由于 HTML 标签导致的带宽浪费，前端用变量保存 AJAX 请求结果，每次操作本地变量，不用请求，减少请求次数

（4）用 innerHTML 代替 DOM 操作，减少 DOM 操作次数，优化 javascript 性能

（5） 用 setTimeout 来避免页面失去响应

（6） 用 hash-table 来优化查找

（7） 当需要设置的样式很多时设置 className 而不是直接操作 style

（8） 少用全局变量

（9） 缓存 DOM 节点查找的结果

（10） 避免使用 CSS Expression

（11） 图片预载

（12） 避免在页面的主体布局中使用 table，table 要等其中的内容完全下载之后才会显示出来，显示比 div+css 布局慢

32，如何控制网页在网络传输过程中的数据量

启用 GZIP 压缩

保持良好的编程习惯，避免重复的 CSS，JavaScript 代码，多余的 HTML 标签和属性

33，Flash、Ajax 各自的优缺点，在使用中如何取舍？

Ajax 的优势

（1） 可搜索型

（2） 开放性

（3） 费用

（4） 易用性

（5） 易于开发

Flash 的优势

（1） 多媒体处理

（2） 兼容性

（3） 矢量图形 比 SVG，Canvas 优势大很多

（4） 客户端资源调度，比如麦克风，摄像头