**Web前端面试题 100题**

1.行内元素有哪些？块级元素有哪些？行内元素和块级元素的区别？

行内元素：a span img input select；块级元素：div p ul h1 h2...ol dl dt；（空元素：br hr img input meta)

区别:1.块级元素会独占一行，默认的宽度占满父级元素，行内元素不会换行；

2.行内元素的width,height无效。

3.块级元素可以设置margin和padding属性，行内元素padding-left、padding-right、margin-left、margin-right

（水平方向）有边距效果，padding-top、padding-bottom、margin-top、margin-bottom（竖直方向）没有边距效果。

本质:浏览器将块级元素的display属性默认为block，行内元素属性默认为inline。因此行内元素与块级元素的切换可以通 过修改display属性实现。

(思考：块元素和行内元素分别如何实现水平居中和垂直居中？）

2、jQuery的事件委托方法on、live、delegate之间有什么区别？

（什么是事件委托？双11，三个人（两个老员工一个新员工想要取快递），两个办法：1.三个人都跑去公司门口等着快递（正常的事件绑定） 2.让前台美女帮忙签收（事件委托给前台美女）。前台美女认识新员工，快递可能不认识新员工。因此，后者新员工肯定能拿到快递）

live把事件委托交给了document；delegate可指定事件委托对象；on事件委托对象选填，如果不填，即给对象自身注册事件，填了作用和delegate一致。

例子：绑定的列表中，每一行都有删除按钮事件。

$().ready(function(){

$(“.btnDelete”).on(“click”,function(){

});//无效 （dom还未加载）

$(“.btnDelete”).live(“click”,function(){

});//有效

$.ajax(...绑定列表)；

});

3.AJAX如何现实跨域请求？

（原理：由于同源策略，正常的ajax无权访问其他域名的资源。后来发现src这个属性却不受同源策略的影响（img,iframed等），其中也包括<script>标签的src属性。于是，人们考虑在远程服务器上把数据放进js文件里返回。这样就可以像ajax一样进行跨域请求了。如：

remote.js：

remoteHandle({"result":"remote数据"});

<script type="text/javascript">

var localHandler = function(data){

alert('remote数据是：' + data.result);

};

</script>

<script type="text/javascript" src="http://xxxxxx.com/remote.js"></script>

这样虽然可以完成跨域请求，但是在大公司里有很多业务。如携程有这样一个场景：酒店业务可以获得中国的一二级城市。

后来机票网站也需要这些一二级城市的数据，怎么样从机票域下请求酒店的数据呢？？？

var queryCity= function(data){

......

};

var url = "[http://xxx.ashx?cityType=1&callback=queryCity](http://xxx.ashx/?cityType=1&callback=queryCity)";

// 创建script标签，设置其属性

var script = document.createElement('script');

script.setAttribute('src', url);

// 把script标签加入head，此时调用开始 document.getElementsByTagName('head')[0].appendChild(script);

)

dResponce.Write(callback+”(data)”);

）

在jQuery里，ajax方法里封装好对JsonP的使用：

$.ajax({

type : "get",

async:false,

url : "xxx.ashx",

dataType : "jsonp",

jsonp: "callbackparam",//传递给请求处理程序或页面的，用以获得jsonp回调函数名的参数名(默认为:callback)

jsonpCallback:"jsonpCallback",//自定义的jsonp回调函数名称，默认为jQuery自动生成的随机函数名

success : function(json){

},

error:function(){

}

});

4、页面导入样式时，使用link和@import有什么区别？

1：link属于XHTML标签，而@import完全是CSS提供的一种方式。

link标签除了可以加载CSS外，还可以做很多其它的事情，比如定义RSS，定义rel连接属性等，@import就只能加载CSS了。   
2：加载顺序的差别

当一个页面被加载的时候，link引用的CSS会同时被加载，而@import引用的CSS会等到页面全部被下载完再被加载。

3：兼容性的差别   
@import只有在IE5以上的才能识别，而link标签无此问题。   
4：使用DOM控制样式时的差别   
当使用JavaScript控制DOM去改变样式的时候，只能使用link标签，因为@import不是DOM可以控制的。   
5：@import可以在CSS中再次引入其他样式表   
可以在样式表中再引入其他的样式表。

4、如何消除一个数组里面重复的元素？

/ 方法一：

var arr1 =[1,2,2,2,3,3,3,4,5,6],

arr2 = [];

for(var i = 0,len = arr1.length; i< len; i++){

if(arr2.indexOf(arr1[i]) < 0){

arr2.push(arr1[i]);

}

}

document.write(arr2); // 1,2,3,4,5,6

* 如何修改函数中this的引用

call(this,arg1,arg2),apply(this,[arg1,arg2])

* 使用document创建以下表格

|  |  |
| --- | --- |
| 学号 | 姓名 |
| 1 | 张三 |
| 2 | 李四 |
| ... | ... |

var table = document.createElement(‘table’);

var tr = table.insertRow(0);

tr.insertCell(0).innerHTML = 1;

tr.insertCell(1).innerHTML = ‘张三’;

var tr1 = table.insertRow(1);

tr1.insertCell(0).innerHTML = 2;

tr1.insertCell(1).innerHTML = ‘李四’;

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

共同点：都是保存在浏览器端，且同源的，不安全，不可靠的。

|  |  |  |  |
| --- | --- | --- | --- |
|  | **cookie** | **localStorage** | **sessionStorage** |
| 数据的生命期 | 设置的过期时间后失效 | 除非被清除，否则永久保存 | 仅在当前会话下有效，关闭页面或浏览器后被清除 |
| 存放数据大小 | 4K左右 | 一般为5MB | 一般为5MB |
| 与服务器端通信 | 每次都会携带在HTTP头中，如果使用cookie保存过多数据会带来性能问题 | 仅在客户端（即浏览器）中保存，不参与和服务器的通信 | 仅在客户端（即浏览器）中保存，不参与和服务器的通信 |
| 场景 | 7天内自动登录等功能 | 移动端缓存数据 | 移动端缓存数据 |

9、统计字符串中字母个数或统计最多字母数。

var str = "aaaabbbccccddfgh";

var obj = {};

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

var v = str.charAt(i);

if(obj[v] && obj[v].value == v){

obj[v].count = ++ obj[v].count;

}else{

obj[v] = {};

obj[v].count = 1;

obj[v].value = v;

}

}

for(key in obj){

document.write(obj[key].value +'='+obj[key].count+'&nbsp;'); // a=4 b=3 c=4 d=2 f=1 g=1 h=1

}

10.请问以下代码输出的结果是什么,请写出代码执行流程？

var a = ‘bbb’; ...........1

function a(){ ............2

console.log(‘aaa’); ..........3

}

a(); .................4

1 2 4 3的顺序执行，函数先声明再调用，因此先执行2 4 再执行3

11.为数字添加一个add方法，实现两数相加

Number.prototype.add = function(a){

return this + a;

}

var a = 2;

var b = a.add(5);

b//7

12.开发中，你是如何解决程序中的错误。

首先使用浏览器调试工具判断错误是逻辑问题还是代码错误。如果是代码错误，找到出错之处，去网上查资料，判断原来的写法是否存在浏览器兼容性问题等。如果是逻辑问题，考虑是否兼容数据等于null时或数据重复时等特殊场景。

13.写一个function，清除字符串前后的空格。（兼容所有浏览器）

function trim(str) {

if (str && typeof str === "string") {

return str.replace(/(^\s\*)|(\s\*)$/g,""); //去除前后空白符

}

}

14.请说说你对闭包的理解，已经应用场景。

闭包的作用就是跨访问变量；修改变量的生存周期。

如： function f1(){

var a = 1; // 作用域

function f2(){

alert(a);

}

}

// 如何在这里使用变量a？只需要将f2写成f1的返回值：

function f1(){

var a =1;//作用域

Return function (){

alert(a);

}

}

Var outFun = f1();

outFun();//如果该变量不被销毁，变量a一直存在

15.JavaScript中原型是什么，有什么用？

每个函数对象都会在其内部初始化一个属性，就是prototype(原型)，当我们访问一个对象的属性时，如果这个对象内部不存在这个属性，那么他就会去prototype里找这个属性，这个prototype又会有自己的prototype，于是就这样一直找下去，也就是我们平时所说的原型链的概念。

关系：instance.constructor.prototype = instance.\_\_proto\_\_

特点：

JavaScript对象是通过引用来传递的，我们创建的每个新对象实体中并没有一份属于自己的原型副本。当我们修改原型时，与之相关的对象也会继承这一改变。

当我们需要一个属性的时，Javascript引擎会先看当前对象中是否有这个属性， 如果没有的话，就会查找他的Prototype对象是否有这个属性，如此递推下去，一直检索到 Object 内建对象。

function Func(){}

Func.prototype.name = "Sean";

Func.prototype.getInfo = function() {

return this.name;

}

var person = new Func();//现在可以参考var person = Object.create(oldObject);

console.log(person.getInfo());//它拥有了Func的属性和方法

//"Sean"

console.log(Func.prototype);

// Func { name="Sean", getInfo=function()}

1. 介绍一下标准的CSS的盒子模型？与低版本IE的盒子模型有什么不同的？

ie 盒子模型的范围包括 margin、border、padding、content，和标准 w3c 盒子模型不同的是：ie 盒子模型的 content 部分还包含了 border 和 pading。

18、CSS选择符有哪些？哪些属性可以继承？

通配选择符 \*，属性选择符 [title = “ydm”],包含选择符 table tr，子选择符 ul>li, ID选择符#btnSave, class选择符 .btnDel

可以继承的有:font-size font-family color等

不可继承有：border padding margin height等

1. CSS优先级算法如何计算？

1. #id > .class .> 标签

2. 层级越具体 优先级越高

3. 标签 #id > #id, 标签 .class > .class

20、CSS3新增伪类有那些？

:first-of-type p:first-of-type 选择属于其父元素的首个 <p> 元素的每个 <p> 元素。   
:last-of-type p:last-of-type 选择属于其父元素的最后 <p> 元素的每个 <p> 元素。  
:only-of-type p:only-of-type 选择属于其父元素唯一的 <p> 元素的每个 <p> 元素。  
:only-child p:only-child 选择属于其父元素的唯一子元素的每个 <p> 元素。  
:nth-child(n) p:nth-child(2) 选择属于其父元素的第二个子元素的每个 <p> 元素。   
:nth-last-child(n) p:nth-last-child(2) 同上，从最后一个子元素开始计数。  
:nth-of-type(n)p:nth-of-type(2) 选择属于其父元素第二个 <p> 元素的每个 <p> 元素。   
:nth-last-of-type(n) p:nth-last-of-type(2) 同上，但是从最后一个子元素开始计数。   
:last-child p:last-child 选择属于其父元素最后一个子元素每个 <p> 元素。   
:root :root 选择文档的根元素。   
:empty p:empty 选择没有子元素的每个 <p> 元素（包括文本节点）。   
:target #news:target 选择当前活动的 #news 元素。  
:enabled input:enabled 选择每个启用的 <input> 元素。   
:disabled input:disabled 选择每个禁用的 <input> 元素   
:checked input:checked 选择每个被选中的 <input> 元素。   
:not(selector) :not(p) 选择非 <p> 元素的每个元素。  
::selection ::selection 选择被用户选取的元素部分。

1. 如何居中div？如何居中一个浮动元素？如何让绝对定位的div居中？

水平居中：margin:0 auto;

居中浮动元素：position： relative;margin-left:-250px;left:50%;

position： absolut;margin-left:-250px;left:50%;

22、display有哪些值？说明他们的作用。

display:none; 隐藏元素

display:block; 设置为块元素

display:inline;设置为行内元素

display：inline-block; 既不会独占一行，又有宽高

1. position的值relative和absolute定位原点是？

absolute：定位的原点是浏览器的左上角

relative：相对于父元素的左上角

思考：如果父元素相对定位，子元素绝对定位，那么子元素的原点是哪里呢？

1. CSS3有哪些新特性？

新的伪类(:disabled,:checked),字体（word-wrap），多列布局（multi-column layout）

圆角（border-radius）,渐变(Gradient)，阴影（shadow），反射（reflect）

1. 请解释一下CSS3的Flexbox（弹性盒布局模型）,以及适用场景？

Flexbox提供了一种很方便的方式对容器中的条目进行排列，布局和分配空间。

属性flex-direction决定了排列顺序（水平或垂直），Order属性确定了条目排列顺序（由小到大排列），flex-grow确定了每一个条目所占空间比例。

26、用纯CSS创建一个三角形的原理是什么？

当div宽高为0时，设置其border的宽度，边框会组成4个三角形，只需要将不需要的三角形设置透明，颜色和背景颜色一致即留下了需要的三角形。

1. 一个满屏品字布局 如何设计?

Html:

<div class=”header”></div>

<div class=”main”>

<div class=”left”></div>

<div class=”right”></div>

<div class=”clear”></div>

</div>

Css:

\*{

margin:0;

Padding:0;

}

.main{

width:100%;

height: 60%;

}

.main .left, .main .right{

width:50%;

height: 100%;

float:left;

background: #a23;

}

.main .right{

background: #e11;

}

.clear{

clear:both;

}

.header{

height:40%;

background: #e33;

width:100%;

}

1. li与li之间有看不见的空白间隔是什么原因引起的？有什么解决办法？

行框的排列会受到中间空白（回车空格等等）的影响，这些空白也会被应用样式，占据空间，所以会有间隔

解决办法，可以设置浮动，不想浮动的话在ul用font-size:0去掉空白，像chrome不支持font-size:0的用letter-spacing:-3px这样的方式去掉li间的空白。

1. 经常遇到的浏览器的兼容性有哪些？原因，解决方法是什么，常用hack的技巧

\* png24位的图片在iE6浏览器上出现背景，解决方案是做成PNG8.

\* 浏览器默认的margin和padding不同。解决方案是加一个全局的\*{margin:0;padding:0;}来统一。

\* IE6双边距bug:块属性标签float后，又有横行的margin情况下，在ie6显示margin比设置的大。

浮动ie产生的双倍距离 #box{ float:left; width:10px; margin:0 0 0 100px;}

这种情况之下IE会产生20px的距离，解决方案是在float的标签样式控制中加入 ——\_display:inline;将其转化为行内属性。(\_这个符号只有ie6会识别)

渐进识别的方式，从总体中逐渐排除局部。

首先，巧妙的使用“\9”这一标记，将IE游览器从所有情况中分离出来。

接着，再次使用“+”将IE8和IE7、IE6分离开来，这样IE8已经独立识别。

css

.bb{

background-color:#f1ee18;/\*所有识别\*/

.background-color:#00deff\9; /\*IE6、7、8识别\*/

+background-color:#a200ff;/\*IE6、7识别\*/

\_background-color:#1e0bd1;/\*IE6识别\*/

}

1. 为什么要初始化CSS样式。

因为浏览器的兼容问题，不同浏览器对有些标签的默认值是不同的，如果没对CSS初始化往往会出现浏览器之间的页面显示差异。

1. positon有哪些值，分别代表什么意思？

Position属性有四个值：static、fixed、absolute和relative.当Position属性值为absolute时对象从文档流中抽取出来，原占有的位置被后面的对象顶替上来。

Top的值表示对象上边框与浏览器窗口顶部的距离,bottom的值表示对象下边框与浏览器窗口底部的距离，两者同时存在时，只有Top起作用；

left的值表示对象左边框与浏览器窗口左边的距离，right的值表示对象右边框与浏览器窗口右边的距离，两者同时存在时，只有left起作用；

当Position属性值为Relative时对象原来占有的位置保留，其后面的对象按原来文档流仍然保持原来的位置。

Top的值表示对象相对原位置向下偏移的距离，bottom的值表示对象相对原位置向上偏移的距离，两者同时存在时，只有Top起作用。

left的值表示对象相对原位置向右偏移的距离，right的值表示对象相对原位置向左偏移的距离，两者同时存在时，只有left起作用。

1. CSS里的visibility属性有个collapse属性值是干嘛用的？在不同浏览器下以后什么区别？

当一个元素的visibility属性被设置成collapse值后，对于一般的元素，它的表现跟hidden是一样的，会占用空间。但如果是table相关的元素，例如table行，table列，它的表现跟display:none一样，它们占用的空间会释放。

1. 对BFC规范(块级格式化上下文：block formatting context)的理解？

即"块级格式化范围"。可以把它理解为一个独立的容器或范围，这个容器决定了内容的位置以及与其他元素的关系和作用等。36、CSS权重优先级是如何计算的？

1. 请解释一下为什么会出现浮动和什么时候需要清除浮动？

设置了浮动后，元素会脱离文档流，因此父容器的背景，边框，margin值都不能正常显示。因此需要清除浮动，将父容器这些属性显示正常。

常用清除方式：clear：both； overflow: auto；

1. 移动端的布局用过媒体查询吗？

用过。它可以根据移动设备样式的不同来进行不同的布局。内联写法：@media screen and (min-width:960px //判断浏览器大小条件){body{background：red} //常规的样式}2.外联写法：当满足屏幕满足条件的时候连接href后的css文件<link='stylesheet' media='screen and (min-width:960)' href='xx.css'/>

1. 使用 CSS 预处理器吗？喜欢那个？

使用过。喜欢less，相较于sass，它上手更快，语法更简单灵活。

喜欢sass，因为sass 功能强大些，语法也稍微多一点点。而且，最新的bootstrap也是基于sass的。

1. CSS优化、提高性能的方法有哪些？

1.对CSS文件压缩

2.尽量减少CSS文件的数量

3.样式尽可能地重用，减少CSS代码

1. 在网页中的应该使用奇数还是偶数的字体？为什么呢？

偶数字体。1.多数设计师如UI都使用偶数字体。2.低版本的浏览器会讲奇数字体渲染成偶数。如IE6会讲13px渲染为14px。3.偶数字体更容易与其他部分构成比例关系。

1. margin和padding分别适合什么场景使用？

需要在border外侧添加空白时，使用margin。需要在boder内侧添加空白时，使用padding。

44、元素竖向的百分比设定是相对于容器的高度吗？

height属性取值百分比，是现对于容器高度的.

对于margin-top、margin-bottom、padding-top、padding-bottom这些竖直方向的内外边距属性的百分比取值，参考的其实是容器的宽度而不是高度。

45.全屏滚动的原理是什么？用到了CSS的那些属性？

（以水平滚动为例，3个div滚动）

1.宽度设为 300%，高度100%。

2.超出页面部分使用overflow:hidden 隐藏

1. 通过left属性控制滚动。
2. 什么是响应式设计？响应式设计的基本原理是什么？如何兼容低版本的IE？

响应式网站设计的理念是：集中创建页面的图片排版大小，可以智能地根据用户行为以及使用的设备环境（系统平台、屏幕尺寸、屏幕定向等）进行相对应的布局。基本原理: 媒体查询 @media兼容IE可以使用JS辅助一下来解决

1. 视差滚动效果，如何给每页做不同的动画？

视差滚动（Parallax Scrolling）就是这样的效果之一。这种技术通过在网页向下滚动的时候，控

制背景的移动速度比前景的移动速度慢来创建出令人惊叹的3D效果。

原理：（1）CSS3实现

优点：开发时间短、性能和开发效率比较好，缺点是不能兼容到低版本的浏览器

（2）jquery实现

通过控制不同层滚动速度，计算每一层的时间，控制滚动效果。

优点：能兼容到各个版本的，效果可控性好

缺点：开发起来对制作者要求高

（3）插件实现方式

例如：parallax-scrolling，兼容性十分好

1. ::before 和 :after中双冒号和单冒号 有什么区别？解释一下这2个伪元素的作用。

（1）单冒号(:)用于CSS3伪类，双冒号(::)用于CSS3伪元素

在css2之前用的是单冒号，之后css3使用时双冒号。目前除了IE外不兼容双冒号，其他的浏览器兼容双

冒号，建议还是使用单冒号。

（2）：：before就是以一个子元素的存在，定义在元素主体内容之前的一个伪元素。并不存在与dom之中，只存在在页面之中。同理，after是在主体内容之后显示的。

1. 如何修改chrome记住密码后自动填充表单的黄色背景 ？

chrome 表单自动填充后，对其赋予以下样式：

input : -webkit-autofill {

background-color : #FAFFBD ;

background-image : none ;

color : #000 ;

}

在有些情况下，这个黄色的背景会影响到我们界面的效果，尤其是在我们给input文本框使用图片背景的时候，原来的圆角和边框都被覆盖了，只需要再次将该背景颜色覆盖即可。

1. 你对line-height是如何理解的？

line-height只影响行内元素，并不能直接应用于块级元素。

line-height 具有可继承性，块级元素的子元素会继承该特性，并且在行内元素上生效。

“把line-height设置为您需要的box的大小可以实现单行文字的垂直居中”。

1. 设置元素浮动后，该元素的display值是多少？（自动变成display:block）
2. 怎么让Chrome支持小于12px 的文字？

CSS3缩放。之前的，-webkit-text-size-adjust:none;这一属性已被淘汰。可以使用如下方式: -webkit-transform-origin-x: 0;-webkit-transform: scale(0.90);

1. 让页面里的字体变清晰，变细用CSS怎么做？

-webkit-font-smoothing在window系统下没有起作用，但是在IOS设备上起作用

-webkit-font-smoothing：antialiased是最佳的，灰度平滑。

55、font-style属性可以让它赋值为“oblique” oblique是什么意思？

italic和oblique都是向右倾斜的文字, 但区别在于Italic是指斜体字，而Oblique是倾斜的文字，对于没有斜体的字体应该使用Oblique属性值来实现倾斜的文字效果.

1. position:fixed;在android下无效怎么处理？

在head头中加入<meta name="viewport" content="width=device-width, initial-scale=1.0,

maximum-scale=1.0, minimum-scale=1.0, user-scalable=no"/>57、如果需要手动写动画，你认为最小时间间隔是多久，为什么？（阿里）

多频显示器的默认频率是60HZ，即1秒刷新60次，所以理论上最小间隔为1/60\*1000ms = 16.7ms

1. display:inline-block 什么时候会显示间隙？(携程)

（1）有空格时候会有间隙 解决：移除空格

（2）margin正值的时候 解决：margin使用负值

（3）使用font-size时候 解决：font-size:0、letter-spacing、word-spacing

1. overflow: scroll时不能平滑滚动的问题怎么处理？

（1）高度尺寸不确定的时候，使用：overflow-y：scroll;

（2）高度尺寸确定的，要么没有滚动条，要么直接出现，不会出现跳动。

（3）css3计算calc和vw单位巧妙实现滚动条出现页面不跳动：

.wrap-outer {

margin-left: calc(100vw - 100%);

}

或.wrap-outer {

padding-left: calc(100vw - 100%);

}

首先，.wrap-outer指的是居中定宽主体的父级，如果没有，创建一个

然后，calc是css3的计算

100vw是浏览器的内部宽度，而100%是可用宽度，不含滚动条

calc（100vw-100%）是浏览器的滚动条的宽度

1. 有一个高度自适应的div，里面有两个div，一个高度100px，希望另一个填满剩下的高度。
2. height：calc（100%-100px）  
   （2）absolute positioning：外层position：relative；  
   百分百自适应元素 position: absolute; top: 100px; bottom: 0; left: 0
3. png、jpg、gif 这些图片格式解释一下，分别什么时候用。有没有了解过webp？

（1）png是便携式网络图片（Portable Network Graphics）是一种无损数据压缩位图文件格式，

优点是：压缩比高，色彩好。 大多数地方都可以用。

（2）jpg是一种针对相片使用的一种失真压缩方法，是一种破坏性的压缩，在色调及颜色平滑变化做的 不错。在www上，被用来储存和传输照片的格式。

（3）gif是一种位图文件格式，以8位色重现真色彩的图像。可以实现动画效果时候webp格式是谷歌在2010年推出的图片格式，压缩率只有jpg的2/3，大小比png小了45%，缺点是压缩的时间更久了 。兼容性不好，目前谷歌和opera支持。

1. 什么是Cookie 隔离？（或者说：请求资源的时候不要让它带cookie怎么做）

Cookie隔离问题，同一个网页,多个RemoteWebDriver会共享同一个Cookie。比如想要并行登陆并执 行操作，这样是不行的。

1. style标签写在body后与body前有什么区别？

页面加载自上而下 当然是先加载样式。

1. 介绍JavaScript的基本数据类型。

1. Number 数字类型

2. String 字符串类型

3. Boolean 布尔类型

4. Function 函数

5. Object 对象

6. Null

7. Undefined 没有定义类型

1. 说说写JavaScript的基本规范？

1、命名规范

1-1，javascript文件：js后缀为.js中，尽量减少html中的js代码。因为存在js代码会明显增加文件大

小，且不能对其进行缓存和压缩。

1-2，缩进单位是4个空格。避免使用tab键来缩进。因为tab始终没有统一tab长短标准，空格会增加文

件的大小，但是可以忽略。

1-3,每行长度不超过80个字符。

1-4。注释，让注释有意义

1-5，变量声明。所有变量必须在使用前通过var进行申明。

1-6，函数声明。所有函数在使用前进行声明。

1-7，命名。大小写字母，10个数字和下划线组成。

1. Javascript如何实现继承？

js中继承是通过他本身的原型机制实现的，js下的继承分为：原型继承和原型子类继承。

1. 说说this指向引用的场景。

（1）this总是指向函数的直接调用者（而非间接调用者）

（2）如果有new关键字，this指向new出来的那个对象

（3）在事件中，this指向触发这个事件的对象，特殊的是IE的attachEvent中的this总是指向全局对象

window。

1. null，undefined的区别？

ull表示一个对象被定义了，但存放了空指针。undefined表示这个值不存在。typeof(null)-- object;typeof(undefined)--undefined

1. [‘22’,true,null].map(parseInt) 答案是多少？

[22, NaN, NaN]原因：parseInt接收的是两个参数，map传递的是3个参 数。

(1)parseInt函数：parseInt(string, radix);string: 需要转化的字符，如果不是字符串会被转换， 忽视空格符。radix：数字2-36之前的整型。默认使用10，表示十进制。需要注意的是，如果radix在2 -36之外会返回NaN。（2）map函数：arr.map(callback[,thisArg]);

1. 什么是闭包（closure），为什么要用它？

闭包指的是一个函数可以访问另一个函数作用域中变量的函数。

常见的构造方法，是在一个函数内部定义另外一个函数。内部函数可以引用外层的参数和变量；参数和变量不会被垃圾回收机制回收。

注意，闭包的原理是作用域链，所以闭包访问的上级作用域中的变量是个对象，其值为其运算结束后的最后一个值。

除非用立即执行函数来解决。

1. javascript 代码中的”use strict”;是什么意思 ? 使用它区别是什么？

除了正常模式运行外，ECMAscript添加了第二种运行模式：“严格模式”。

区别：

（1）消除js不合理，不严谨地方，减少怪异行为

（2）消除代码运行的不安全之处，

（3）提高编译器的效率，增加运行速度

（4）为未来的js新版本做铺垫。

1. 如何判断一个对象是否属于某个类？

使用instanceof 即if(a instanceof Person){alert('yes');}

1. js延迟加载的方式有哪些？

（1）使用setTimeout延迟方法的加载时间 $(function (){

setTimeout('A()', 1000); //延迟1秒})

1. 让js最后加载引入外部js脚本文件时，如果放入html的head中,则页面加载前该js脚本就会被加载入页面，而放入 body中，则会按照页面从上倒下的加载顺序来运行javascript的代码~~~ 所以我们可以把js外部引入的 文件放到页面底部，来让js最后引入，从而加快页面加载速度。
2. 同步和异步的区别，异步编程的实现方式有哪几种?

同步的概念是os中：不同进程协同完成某项工作而先后次序调整（通过阻塞、唤醒等方式），同步 强调的是顺序性，谁先谁后。。异步不存在顺序性。同步：浏览器访问服务器，用户看到页面刷新，重 新发请求，等请求完，页面刷新，新内容出现，用户看到新内容之后进行下一步操作。异步：浏览器访 问服务器请求，用户正常操作，浏览器在后端进行请求。等请求完，也买你不刷新，新内容也会出现。

1. 页面编码和被请求的资源编码如果不一致如何处理？

若请求的资源编码，如外引js文件编码与页面编码不同。可根据外引资源编码方式定义为

charset="utf-8"或"gbk"。

1. JavaScript如何实现模块化开发？

模块化开发指的是在解决某一个复杂问题或者一系列问题时，依照一种分类的思维吧问题进行系统 性的分解以之解决。模块化是一种处理复杂系统分解为代码结构更合理，可维护性更高的可管理的模块 方式。对于软件行业：系统被分解为一组高内聚，低耦合的模块。

（1）定义封装的模块

（2）定义新模块对其他模块的依赖

（3）可对其他模块的引入支持

1. 谈一谈你对ECMAScript6的了解？
2. 默认参数2.模板对象3.多行字符串4.解构赋值5.增强的对象字面量6.箭头函数7. Promises8.块作用域和构造let和const 9. 类10.模块
3. ECMAScript6 怎么写class么，为什么会出现class这种东西?

class baseModel {

constructor(options, data) { // class constructor，node.js 5.6暂时不支持options = {}, data = []这样传参

this.name = 'Base';

this.url = 'http://azat.co/api';

this.data = data;

this.options = options;

}

getName() { // class method

console.log(`Class name: ${this.name}`);

}

类的创建和使用真是一件令人头疼的事情在过去的ES5中，因为没有一个关键字class （它被保留，但是什么也不能做）。在此之上，大量的继承模型像pseudo classical, classical, functional 更加增加了混乱。

1. 异步加载的方式有哪些？

方案一：<script>标签的async="async"属性（详细参见：script标签的async属性）

点评：HTML5中新增的属性，Chrome、FF、IE9&IE9+均支持（IE6~8不支持）。此外，这种方法不能保证脚本按顺序执行。

方案二：<script>标签的defer="defer"属性

点评：兼容所有浏览器。此外，这种方法可以确保所有设置defer属性的脚本按顺序执行。

方案三：动态创建<script>标签

方案四：AJAX eval（使用AJAX得到脚本内容，然后通过eval\_r(xmlhttp.responseText)来运行脚本）

点评：兼容所有浏览器。

方案五：iframe方式（这里可以参照：iframe异步加载技术及性能 中关于Meboo的部分）

点评：兼容所有浏览器。

1. call() 和 .apply() 的含义和区别？

借给另一个对象的调用去完成任务,原理上是方法执行时上下文对象改变了即：切换执行函数上下文。

二者参数不同。Call后面传入数组，apply传入变量。

83、数组和对象有哪些原生方法，列举一下？

数组：

1 shift()t:删除数组的第一个元素,返回删除的值。这里是0

2 unshift(3,4):把参数加载数组的前面，返回数组的长度。现在list:中是3,4,0,1,2

3pop():删除数组的最后一个元素，返回删除的值。这里是2.

4push(3):将参数加载到数组的最后，返回数组的长度，现在List中时：0,1,2,3

5concat(3,4):把两个数组拼接起来。

6splice(start,deleteCount,val1,val2,...)：从start位置开始删除deleteCount项，并从该位置起插入val1,val2,...

reverse：将数组反序

var a = [1,2,3,4,5];

var b = a.reverse(); //a：[5,4,3,2,1] b：[5,4,3,2,1]

sort(orderfunction)：按指定的参数对数组进行排序

var a = [1,2,3,4,5];

var b = a.sort(); //a：[1,2,3,4,5] b：[1,2,3,4,5]

slice(start,end)：返回从原数组中指定开始下标到结束下标之间的项组成的新数组

var a = [1,2,3,4,5];

var b = a.slice(2,5); //a：[1,2,3,4,5] b：[3,4,5]

join(separator)：将数组的元素组起一个字符串，以separator为分隔符，省略的话则用默认用逗号为分隔符

var a = [1,2,3,4,5];

对象：

hasOwnProperty(property):判断对象是否有某个特定的属性。必须用字符串指定该属性（例如，o.hasOwnProperty(”name”)）。

isPrototypeOf(object):判断该对象是否为另一个对象的原型。

propertyIsEnumerable(property):判断给定的属性是否可以用for…in语句进行枚举。

toString():返回对象的原始字符串表示。对于Object类，ECMA－262没有定义这个值，所以不同的ECMAScriipt实现具有不同的值。

valueOf():返回最适合该对象的原值。对于许多类，该方法返回的值都与toString()的返回值相同。

84、JavaScript中的作用域与变量声明提升？

如果存在函数声明和变量声明（注意：仅仅是声明，还没有被赋值），而且变量名跟函数名是相同的，那么，它们都会被提示到外部作用域的开头，但是，函数的优先级更高，所以变量的值会被函数覆盖掉。

85、什么是“前端路由”?什么时候适合使用“前端路由”? “前端路由”有哪些优点和缺点?

根据不同的url展示不同的数据这一步交给前端来完成。SPA应用适合使用“前端路由”。优点是：路由处理速度较快，不用每次都通过服务返回。缺点是使用浏览器的前进，后退键的时候会重新发送请求，没有合理地利用缓存。单页面无法记住之前滚动的位置，无法在前进，后退的时候记住滚动的位置

86、如何实现前后端分离开发？

即：后端负责业务/数据接口，前端负责展现/交互逻辑。展现数据都是通过异步接口(AJAX/JSONP)的方式提供的，前端只管展现。前后端分离的好处有：前后端分离的好处有前端模块化，和后端分离，部署方便；后端使用webAPI的方式部署；良好的可扩展性

92、NaN===NaN输出什么？

输出false。原因就和无穷大不等于无穷大一样。

93、下面的代码会输出什么？为啥？

for (var i = 0; i < 5; i++) {

setTimeout(function() { console.log(i); }, i \* 1000 );

}

依次输出5个5。因为setTimeout是异步的，先执行了5次i++此时i已经等于5了。

可以通过迭代等手段输出0 1 2 3 4

94、解释下面代码的输出

console.log(false == '0') //true

console.log(false === '0') //false

95.如果编写一个jQuery插件？

$.xxx => $.extend(object);

$(dom).XXX => $.fn.extend(object);

（function ($) {

$.fn.exetend({  
 haha: function () {

}

})

}）（jQuery）