## HTML篇

1. Doctype作用？标准模式与兼容模式各有什么区别?

答：

<!DOCTYPE>声明位于位于HTML文档中的第一行，处于 <html> 标签之前。告知浏览器的解析器用什么文档标准解析这个文档。DOCTYPE不存在或格式不正确会导致文档以兼容模式呈现。

标准模式的排版 和JS运作模式都是以该浏览器支持的最高标准运行。在兼容模式中，页面以宽松的向后兼容的方式显示,模拟老式浏览器的行为以防止站点无法工作。

注：

1. HTML5 为什么只需要写 <!DOCTYPE HTML>？

答：

HTML5 不基于 SGML，因此不需要对DTD进行引用，但是需要doctype来规范浏览器的行为（让浏览器按照它们应该的方式来运行）；

而HTML4.01基于SGML,所以需要对DTD进行引用，才能告知浏览器文档所使用的文档类型。

注：

1. 行内元素有哪些？块级元素有哪些？ 空(void)元素有那些？

答：

首先：CSS规范规定，每个元素都有display属性，确定该元素的类型，每个元素都有默认的display值，如div的display默认值为“block”，则为“块级”元素；span默认display属性值为“inline”，是“行内”元素。

（1）行内元素有：a b span img input select strong（强调的语气）

（2）块级元素有：div ul ol li dl dt dd h1 h2 h3 h4…p

（3）常见的空元素：

<br> <hr> <img> <input> <link> <meta>

鲜为人知的是：

<area> <base> <col> <command> <embed> <keygen> <param> <source> <track> <wbr>

注：

1. 页面导入样式时，使用link和@import有什么区别？

答：

（1）link属于XHTML标签，除了加载CSS外，还能用于定义RSS, 定义rel连接属性等作用；而@import是CSS提供的，只能用于加载CSS;

（2）页面被加载的时，link会同时被加载，而@import引用的CSS会等到页面被加载完再加载;

（3）import是CSS2.1 提出的，只在IE5以上才能被识别，而link是XHTML标签，无兼容问题;

注：

1. html5有哪些新特性、移除了那些元素？如何处理HTML5新标签的浏览器兼容问题？如何区分 HTML 和 HTML5？

答：

HTML5 现在已经不是 SGML 的子集，主要是关于图像，位置，存储，多任务等功能的增加。

拖拽释放(Drag and drop) API

语义化更好的内容标签（header,nav,footer,aside,article,section）

音频、视频API(audio,video)

画布(Canvas) API

地理(Geolocation) API

本地离线存储 localStorage 长期存储数据，浏览器关闭后数据不丢失；

sessionStorage 的数据在浏览器关闭后自动删除

表单控件，calendar、date、time、email、url、search

新的技术webworker, websocket, Geolocation

移除的元素：

纯表现的元素：basefont，big，center，font, s，strike，tt，u；

对可用性产生负面影响的元素：frame，frameset，noframes；

支持HTML5新标签：

IE8/IE7/IE6支持通过document.createElement方法产生的标签，可以利用这一特性让这些浏览器支持HTML5新标签，浏览器支持新标签后，还需要添加标签默认的样式：

当然最好的方式是直接使用成熟的框架、使用最多的是html5shim框架

<!--[if lt IE 9]>

<script> src="http://html5shim.googlecode.com/svn/trunk/html5.js"</script>

<![endif]-->

如何区分： DOCTYPE声明\新增的结构元素\功能元素

注：

1. 简述一下你对HTML语义化的理解？

答：

HTML语义化就是根据内容的结构化（内容语义化），选择合适的标签（代码语义化）便于开发者阅读和写出更优雅的代码的同时让浏览器的爬虫和机器很好地解析。

语义化的好处：

在没有CSS的情况下，页面也能呈现出很好地内容结构、代码结构；

用户体验；

有利于SEO；

方便其他设备解析（如屏幕阅读器、盲人阅读器、移动设备）以意义的方式来渲染网页；

便于团队开发和维护，语义化更具可读性。

注：

1. HTML5的离线储存怎么使用，工作原理能不能解释一下？

答：

使用 HTML5，通过创建 cache manifest 文件，可以轻松地创建 web 应用的离线版本。

HTML5引入了应用程序缓存。

manifest 文件是简单的文本文件，它告知浏览器被缓存的内容（以及不缓存的内容）。

manifest 文件可分为三个部分：

CACHE MANIFEST - 在此标题下列出的文件将在首次下载后进行缓存

NETWORK - 在此标题下列出的文件需要与服务器的连接，且不会被缓存

FALLBACK - 在此标题下列出的文件规定当页面无法访问时的回退页面（比如 404 页面）

在线的情况下,用户代理每次访问页面，都会去读一次manifest.如果发现其改变, 则重新加载全部清单中的资源

HTML5的离线存储是基于一个新建的.appcache文件的，通过这个文件上的解析清单离线存储资源，这些资源就会像cookie一样被存储了下来。之后当网络在处于离线状态下时，浏览器会通过被离线存储的数据进行页面展示。

1. iframe有那些缺点？

答：

iframe会阻塞主页面的Onload事件；

页面样式调试麻烦，出现多个滚动条；

浏览器的后退按钮失效；

过多会增加服务器的HTTP请求；

小型的移动设备无法完全显示框架；

产生多个页面，不易管理；

不容易打印；

代码复杂，无法被一些搜索引擎解读。

1. Label的作用是什么？是怎么用的？

答：

label标签来定义表单控件间的关系,当用户选择该标签时，浏览器会自动将焦点转到和标签相关的表单控件上。

1. HTML5的form如何关闭自动完成功能？

答：

网页中表单的自动完成功能，有时候很方便，但是有时候并不想让浏览器记忆表单，比如禁用密码域自动填充功能，网页禁用表单的自动完成功能是由input元素的autocomplete属性控制，关闭表单的自动完成就是设置autocomplete="off"，默认是就是开的值为on

关闭密码域的自动完成：

<input type="password" name="password" autocomplete="off">

关闭整个表单（form）自动提示功能：

在表单上设置autocomplete="off"

<form name="form1" autocomplete="off">

<input type="text" name="userName">

<input type="text" name="password">

</form>

1. title与h1的区别、b与strong的区别、i与em的区别？

答：

tilte与h1的区别

从搜索引擎角度来说，title标签是用来描述这个页面的主题的，是一个网页权重的最高点。但title标签并不出现在文章的正文中。而h1标签一般出现在文章的正文中，是展示给访问者的文章的标题。所以说这两个标签不仅不冲突的，而是合作的关系。一篇文章既要有title又要有h1标签，既突出了文章的主题，又突出了标题和关键字，达到双重优化网站的效果。

一般会把title和h1标签的内容写成一样，而且一般情况下一篇文章最好只用一个h1标签，过多的h1标签反而会让搜索引擎迷糊，认不清文章的主题。

b与strong的区别、i与em的区别

其实这两对标签最大区别就是一个给搜索引擎看的，一个是给用户看的。

b标签和strong标签给我们的主观感受都是加粗，但对搜索引擎来说b标签和普通的文字并没有什么区别，而strong标签却是起强调作用的。也就是说如果你想让搜索引擎认为你的某句话很重要时那就用strong标签。如果只是想让用户看到加粗的效果，那就用b标签。

同理如em标签也是针对搜索引擎来起作用的，i标签只是让用户看到展示的是斜体。

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？

答：

浏览器的默认行为是把inline元素间的空白字符（空格换行tab）渲染成一个空格，也就是我们上面的代码<li>换行后会产生换行字符，而它会变成一个空格，当然空格就占用一个字符的宽度

解决方法：

（1）浮动li中float：left

（2）在ul中用font-size：0（谷歌不支持）；可以使用letter-space：-3px

1. 简述一下src与href的区别

答：

src用于替换当前元素，href用于在当前文档和引用资源之间确立联系。

src是source的缩写，指向外部资源的位置，指向的内容将会嵌入到文档中当前标签所在位置；在请求src资源时会将其指向的资源下载并应用到文档内，例如js脚本，img图片和frame等元素。当浏览器解析到该元素时，会暂停其他资源的下载和处理，直到将该资源加载、编译、执行完毕。

href是Hypertext Reference的缩写，指向网络资源所在位置，建立和当前元素（锚点）或当前文档（链接）之间的链接。当浏览器解析到该元素时，会并行下载资源并且不会停止对当前文档的处理.

## CSS篇

1. 写出position不同值和区别

答：

static（静态定位）：默认值。没有定位，元素出现在正常的流中（忽略 top, bottom, left, right 或者 z-index 声明）；

relative（相对定位）：生成相对定位的元素，通过top,bottom,left,right的设置相对于其正常（原先本身）位置进行定位。可通过z-index进行层次分级；

absolute（绝对定位）：生成绝对定位的元素，相对于 static 定位以外的第一个父元素进行定位。元素的位置通过 "left", "top", "right" 以及 "bottom" 属性进行规定。可通过z-index进行层次分级；

fixed（固定定位）：生成绝对定位的元素，相对于浏览器窗口进行定位。元素的位置通过 "left", "top", "right" 以及 "bottom" 属性进行规定。可通过z-index进行层次分级

注：

1. 讲述你对reflow和repaint的理解

答：

reflow，浏览器根据各种样式来计算并根据计算结果将元素放到确定的地方。repaint： relfow完成之后，浏览器把这些元素按照各自的特性绘制一遍

引起repaint的操作

一个元素的外观改变，但是没有改变布局的情况

visibility

outline

background color

引起reflow的操作

改变窗口大小

改变字体

增加和删除样式表

内容的改变，比如用户在输入框输入文字

激活伪类

操作class属性

脚本操作DOM

计算offsetWidth和offsetHeight

设置style属性

display:none 触发reflow，而visibility:hidden 触发repaint，因为没有发生位置变化

避免和最小化影响

直接改变你想改变的元素（避免通过父元素作用于子元素，而是直接作用于子元素）

避免黄色至内联样式

对于动画的元素，其position设为fixed或absolute

权衡速度的平滑，速度慢，reflow比较频繁

避免table布局

避免css中含有js 表达式（只有IE）

在性能优先的前提下，性能消耗 reflow大于repaint。

repaint是某个DOM元素进行重绘；reflow是整个页面进行重排，也就是页面所有DOM元素渲染。

style变动造成repaint和reflow。

不涉及任何DOM元素的排版问题的变动为repaint，例如元素的color/text-align/text-decoration等等属性的变动。

除上面所提到的DOM元素style的修改基本为reflow。例如元素的任何涉及长、宽、行高、边框、display等style的修改。

注：

1. 介绍一下标准的CSS的盒子模型？低版本IE的盒子模型有什么不同的？

答：

CSS盒子模型：由四个属性组成的外边距(margin)、内边距(padding)、边界(border)、内容区(width和height);

标准的CSS盒子模型和低端IE CSS盒子模型不同：宽高不一样

标准的css盒子模型宽高就是内容区宽高；

低端IE css盒子模型宽高 内边距﹢边界﹢内容区；

1. CSS选择符有哪些？哪些属性可以继承？

答：

选择符有：

id选择器（ # myid）

类选择器（.myclassname）

标签选择器（div, h1, p）

相邻选择器（h1 + p）

子选择器（ul > li）

后代选择器（li a）

通配符选择器（ \* ）

属性选择器（a[rel = "external"]）

伪类选择器（a:hover, li:nth-child）

可继承的样式： font-size font-family color, UL LI DL DD DT;

不可继承的样式：border padding margin width height ;

1. CSS优先级算法如何计算？

答：

行内样式优先级specificity值为1,0,0,0 高于外部定义

　　如<div style="height:50px; width:50px;">Div</div> //行内样式

　　外部定义指经由<link>或<style>标签定义的规则

按CSS代码中出现的顺序决定，后者CSS样式居上；

!important声明specificity值优先级最高

由继承而得到的样式没有specificity的计算，它低于一切其他规则（比如全局选择符\*定义规则）

1. CSS3新增伪类有那些？

答：

p:first-of-type 选择属于其父元素的首个<p>元素的每个<p>元素。

p:last-of-type 选择属于其父元素的最后<p>元素的每个<p>元素。

p:only-of-type 选择属于其父元素唯一的<p>元素的每个<p>元素。

p:only-child 选择属于其父元素唯一的子元素的每个<p>元素。

p:nth-child(n) 选择属于其父元素的第n个子元素的每个<p>元素。

p:nth-last-child(n) 选择属于其父元素的倒数第n个子元素的每个<p>元素。

p:nth-of-type(n) 选择属于其父元素第n个<p>元素的每个<p>元素。

p:nth-last-of-type(n) 选择属于其父元素倒数第n个<p>元素的每个<p>元素。

p:last-child 选择属于其父元素最后一个子元素的每个<p>元素。

p:empty选择没有子元素的每个<p>元素（包括文本节点）。

p:target 选择当前活动的<p>元素。

:not(p) 选择非<p>元素的每个元素。

:enabled 控制表单控件的可用状态。

:disabled 控制表单控件的禁用状态。

:checked 单选框或复选框被选中。

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？

答：

1. 元素水平居中的方式
2. 行级元素水平居中对齐（父元素设置 text-align:center）
3. 块级元素水平居中对齐（margin: 0 auto）
4. 浮动元素水平居中

宽度不固定的浮动元素

<div class="outerbox">

<div class="innerbox">我是浮动的</div>

</div>

.outerbox{

float:left;

position:relative;

left:50%;

}

.innerbox{

float:left;

position:relative;

right:50%;

}

宽度固定的互动元素

<div class="outerbox">

<div>我是浮动的</div>

</div>

.outerbox{

background-color:pink; /\*方便看效果 \*/

width:500px ;

height:300px; /\*高度可以不设\*/

margin: -150px 0 0 -250px; /\*使用marin向左移动250px，保证元素居中\*/

position:relative; /\*相对定位\*/

left:50%;

top:50%;

}

1. 让绝对定位的元素水平居中对齐

.center{

position: absolute; /\*绝对定位\*/

width: 500px;

height:300px;

background: red;

margin: 0 auto; /\*水平居中\*/

left: 0; /\*此处不能省略，且为0\*/

right: 0; /\*此处不能省略，且为0\*/

}

1. 元素垂直居中对齐
2. 对**行级**元素垂直居中（heiht与line-height的值一样）
3. 对块级元素垂直居中对齐

父元素高度固定的情况

父元素的height与line-height值相同

需要垂直居中的元素

vertical-align:middle;// 垂直居中对齐

display:inline|inline-block 块级元素转行级元素

父元素高度不固定的情况

父元素的padding-top和padding-bottom一样

参考： <http://blog.csdn.net/lxcao/article/details/52670724>

1. display有哪些值？说明他们的作用。

答：



常用的有：

none：此元素不显示。

block：将元素显示为块级元素，前后会带换行符。

inline:默认值，元素会被显示为内联元素，前后没有换行符。

inline-block:行内块级元素。

1. CSS3有哪些新特性？

答：

1. 用纯CSS创建一个三角形的原理是什么？

答：

1. 一个满屏 品 字布局 如何设计?

答：

1. 为什么要初始化CSS样式。

答：

1. absolute的containing block(容器块)计算方式跟正常流有什么不同？

答：

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？

答：

1. css定义的权重

答：

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？清除浮动的方式

答：

1. 设置元素浮动后，该元素的display值是多少？（自动变成display:block）

答：

1. 让页面里的字体变清晰，变细用CSS怎么做？（-webkit-font-smoothing: antialiased;）

答：

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

答：

1. style标签写在body后与body前有什么区别？

答：

1. 什么是CSS 预处理器 / 后处理器？

答：

1. CSS实现垂直水平居中

答：

1. 什么是CSS Hack?

答：

1. px和em的区别

答：

## JavaScript篇

1. alert(1&&2),alert(1||0)

答：

* 2
* 1

注：

1. mouseenter和mouseover的区别

答：

注：

1. 用正则表达式匹配字符串，以字母开头，后面是数字、字符串或者下划线，长度为9-20

答：

/^[a-zA-Z]\w{8,19}$/

注：

1. js字符串两边截取空白的trim的原型方法的实现

答：

String.prototype.trim = function () {

return this.replace(/^\s+|\s+$/g, ‘’)

}

注：

1. 写出下列代码结果

var a = 4;

function b() {

a = 3;

console.log(a);

function a() {};

}

b();

答：

* 3

注：

变量提升

在ES6之前，JavaScript没有块级作用域(一对花括号{}即为一个块级作用域)，只有全局作用域和函数作用域。变量提升即将变量声明提升到它所在作用域的最开始的部分

console.log(global); // undefined

var global = 'global';

console.log(global); // global

function fn () {

　　 console.log(a); // undefined

　　var a = 'aaa';

　　console.log(a); // aaa

}

fn();

函数提升

js中创建函数有两种方式：函数声明式和函数字面量式。只有函数声明才存在函数提升！

console.log(f1); // function f1() {}

console.log(f2); // undefined

function f1() {}

var f2 = function() {}

1. 写出下列代码结果

var baz = 3;

var bazz = {

baz: 2,

getbaz: function () {

return this.baz

}

}

console.log(bazz.getbaz())

var g = bazz.getbaz;

console.log(g());

答：

* 2
* 3

注：

1. 写出下列代码的结果

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

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

arr[i] = function () {

alert(i)

}

}

arr[3]();

答：

* 5

注：

这就是 var 与 let 的区别

1. 写出下列代码的运行结果

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

setTimeout(function() {

console.log(new Date, i);

}, 1000);

}

答：

* Tue May 16 2017 15:05:27 GMT+0800 , 5
* Tue May 16 2017 15:05:27 GMT+0800 , 5
* Tue May 16 2017 15:05:27 GMT+0800 , 5
* Tue May 16 2017 15:05:27 GMT+0800 , 5
* Tue May 16 2017 15:05:27 GMT+0800 , 5

注：

时间：

星期（en） 月份（en） 日 年 时:分:秒 GMT+0800

同步 => 异步 => 回调

1. 请列举出JS中几种实用的跨域方法及原理

答：

注：

1. 请写出下列代码的运行结果

["1", "2", "3"].map(parseInt)

答：

* [1, NaN, NaN]

注：

map 函数会给参数 parseInt 函数传递两个参数 item, 和 index

parseInt 接收两个参数：数字字符串、基数（XX进制， 0表示10进制）

1. 请写出下列代码的运行结果

[typeof null, null instanceof Object]

答：

* [‘object’, false]

注：

1. 请写出下列代码的运行结果

[ [3,2,1].reduce(Math.pow), [].reduce(Math.pow) ]

答：

* error

注：

reduce()把结果继续和序列的下一个元素做累积计算，其效果就是：

[x1, x2, x3, x4].reduce(f) = f(f(f(x1, x2), x3), x4)

1. 请写出下列代码的运行结果

var val = 'smtg';

console.log('Value is ' + (val === 'smtg') ? 'Something' : 'Nothing');

答：

* 'Something'

注：

1. 请写出下列代码的运行结果

var name = 'World!';

(function () {

if (typeof name === 'undefined') {

var name = 'Jack';

console.log('Goodbye ' + name);

} else {

console.log('Hello ' + name);

}

})();

答：

* Goodbye Jack

注：变量在未赋值之前变量类型都是undefined

1. 请写出下列代码的运行结果

var END = Math.pow(2, 53);

var START = END - 100;

var count = 0;

for (var i = START; i <= END; i++) {

count++;

}

console.log(count);

答：

一直循环，没有输出

注：

2的53次方很大，相当于无穷大，无穷大加减1都是无穷大

1. 请写出下列代码的运行结果

var two = 0.2

var one = 0.1

var eight = 0.8

var six = 0.6

[two - one == one, eight - six == two]

答：

* [true, false]

注：

1. 请写出下列代码的运行结果

function showCase(value) {

switch(value) {

case 'A':

console.log('Case A');

break;

case 'B':

console.log('Case B');

break;

case undefined:

console.log('undefined');

break;

default:

console.log('Do not know!');

}

}

showCase(new String('A'));

答：

* Do not know!

注：switch 会对参数进行严格比较

1. 请写出下列代码的运行结果

function showCase2(value) {

switch(value) {

case 'A':

console.log('Case A');

break;

case 'B':

console.log('Case B');

break;

case undefined:

console.log('undefined');

break;

default:

console.log('Do not know!');

}

}

showCase2(String('A'));

答：

* Case A

注：

String() 函数把对象的值转换为字符串，对象不变。

1. 请写出下列代码的运行结果

function isOdd(num) {

return num % 2 == 1;

}

function isEven(num) {

return num % 2 == 0;

}

function isSane(num) {

return isEven(num) || isOdd(num);

}

var values = [7, 4, '13', -9, Infinity];

values.map(isSane);

答：

* [true, true, true, false, false]

注：

1. 请写出下列代码的运行结果

Array.isArray( Array.prototype )

答：

true

注：

1. 请写出下列代码的运行结果

if ([0]) {

console.log(a == true);

} else {

console.log("wut");

}

答：

* erro: a is not defined

注：

1. 请写出下列代码的运行结果

[]==[]

答：

* false

注：

1. 请写出下列代码的运行结果

'5' + 3

'5' – 3

答：

* ‘53’
* 2

注：

1. 请写出下列代码的运行结果

1 + - + + + - + 1

答：

2

注：

1 + - + + + - + 1

1 - + + + - 1

1 - + + + -1

1 - + + -1

1 - + -1

1 - -1 = 2

1. 请写出下列代码的运行结果

var ary = Array(3);

ary[0]=2

ary.map(function(elem) { return '1'; });

答：

["1", undefined × 2]

注：

1. 简述同步和异步的区别

答：

1. 怎样添加、移除、移动、复制、创建和查找节点？

答：

1. 实现一个函数clone，可以对JavaScript中的5种主要的数据类型（包括Number、String、Object、Array、Boolean）进行值复制。

答：

1. 如何消除一个数组里面重复的元素？

答：

1. 想实现一个对页面某个节点的拖曳？如何做？（使用原生JS）。

答：

1. 在Javascript中什么是伪数组？如何将伪数组转化为标准数组？

答：

伪数组：

具有length属性；

按索引方式存储数据；

不具有数组的push()、pop()等方法；

伪数组无法直接调用数组方法或期望length属性有什么特殊的行为，不具有数组的push()、pop()等方法，但仍可以对真正数组遍历方法来遍历它们。这种对象有很多，比较特别的是function内的arguments对象，还有像调用getElementsByTagName, document.childNodes之类的，它们都返回的NodeList对象都属于伪数组，也称为类数组，还有自定义的对象，也属于伪数组。我们可以通过Array.prototype.slice.call(fakeArray)将伪数组转变为真正的Array对象。

代码例子如下：

var fakeArray01 = {0:'a',1:'b',length:2};//这是一个标准的伪数组对象

var arr01 = Array.prototype.slice.call(fakeArray01);

alert(arr01[0]);//a

var arr02 = [].slice.call(fakeArray01);

alert(arr02[0]);//a

1. Javascript中callee和caller的作用？

答：

callee是arguments对象的一个属性，指向arguments对象的函数；

caller是函数对象的一个属性，该属性保存着调用当前函数的函数的引用（指向当前函数的直接父函数），对于函数来说，caller 属性只有在函数执行时才有定义。 如果函数是由 Javascript 程序的顶层调用的，那么 caller 包含的就是 null。

他们都会把函数内容打印出来。

1. 请描述一下cookies，sessionStorage和localStorage的区别

答：

1. 手写数组快速排序

答：

1. 统计字符串”aaaabbbccccddfgh”中字母个数或统计最多字母数。

答：

1. 写一个function，清除字符串前后的空格。（兼容所有浏览器）

答：

1. 正则表达式匹配，开头为11N, 12N或1NNN，后面是-7-8个数字的电话号码。

答：

1. 写出下面代码的输出值：

var obj = {

a: 1,

b: function () {console.log(this.a)}

};

var a = 2;

var objb = obj.b;

obj.b();

objb();

obj.b.call(window);

答：

* 1
* 2
* 2

1. 写出下列代码的输出值：

function A() {

}

function B(a) {

this.a = a;

}

function C(a) {

if (a) {

this.a = a;

}

}

A.prototype.a = 1;

B.prototype.a = 1;

C.prototype.a = 1;

console.log(new A());

console.log(new B());

console.log(new C(2));

答：

* {}
* {a: undefined}
* {a: 2}
* 它们的prototype.a都为1

1. 写出下列代码的输出值：

var a = 1;

function b() {

var a = 2;

function c() {

console.log(a);

}

return c;

}

b()();

答：

* 2

1. 添加些css让其水平垂直居中。

<div style="\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_">

颜海镜

</div>

答：

1. 如下代码，在空白处填写代码，是其点击时，前景色为白色，背景色为黑色。

<div onclick="\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_">颜海镜</div>

答：

1. 对比 event.preventDefault() 和 event.stopPropagation()

答：

event.stopPropagation(); // 事件停止冒泡到，即不让事件再向上传递到document，但是此事件的默认行为仍然被执行，如点击一个链接，调用了event.stopPropagation()，链接仍然会被打开；

event.preventDefault(); // 取消了事件的默认行为，如点击一个链接，链接不会被打开，但是此事件仍然会传递给更上一层的先辈元素。

## 算法篇

1. A、B两人分别在两座岛上。B生病了，A有B所需要的药。C有一艘小船和一个可以上锁的箱子。C愿意在A和B之间运东西，但东西只能放在箱子里。只要箱子没被上锁，C都会偷走箱子里的东西，不管箱子里有什么。如果A和B各自有一把锁和只能开自己那把锁的钥匙，A应该如何把东西安全递交给B？

答：

## AngularJs篇

1. ng-show/ng-hide 与 ng-if的区别？

答：

注：

1. 解释下什么是$rootScrope以及和$scope的区别？

答：

注：

1. 表达式 {{yourModel}}是如何工作的？

答：

注：

## jQuery篇

1. 解释下面这行代码会做什么：

$( "div#first, div.first, ol#items > [name$='first']" )

答：

1. 什么是 jQuery 的方法链式调用？写一个例子。说说这么写有什么好处？

答：

1. .clone() 方法

答：

1. 深拷贝有哪些例外情况？如何去控制这些情况？

答：

## 软件篇

1. 介绍一下你对浏览器内核的理解？

答：

注：

1. 常见的浏览器内核有哪些？

答：

注：

1. 浏览器是怎么对HTML5的离线储存资源进行管理和加载的呢？

答：

1. 如何实现浏览器内多个标签页之间的通信? (阿里)

答：

1. webSocket如何兼容低浏览器？(阿里)

答：

1. 页面可见性（Page Visibility API） 可以有哪些用途？

答：

1. 如何在页面上实现一个圆形的可点击区域？

答：

1. 网页验证码是干嘛的，是为了解决什么安全问题。

答：

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 ？

答：

1. 怎么让Chrome支持小于12px 的文字？

答：

1. 什么叫优雅降级和渐进增强？

答：

1. 一次完整的HTTP事务是怎样的一个过程？

答：

1. 写出下列代码在各个浏览器中的颜色值?

background: red;

\_background: green;

\*background: blue;

background: black\9;

答：

1. 简述在IE下mouseover和mouseenter的区别？

答：