

第一部分 HTML+CSS

一、Doctype 作用？严格模式与混杂模式如何区分？它们有何意义？

- (1)、<!DOCTYPE> 声明位于文档中的最前面，处于 <html> 标签之前。告知浏览器的解析器，用什么文档类型 规范来解析这个文档。
- (2)、严格模式的排版和 JS 运作模式是 以该浏览器支持的最高标准运行。
- (3)、在混杂模式中，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。
- (4)、DOCTYPE 不存在或格式不正确会导致文档以混杂模式呈现。

二、行内元素有哪些？块级元素有哪些？空(void)元素有哪些？

- (1) CSS 规范要求，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，
比如 div 默认 display 属性值为 “block”，成为 “块级” 元素；
span 默认 display 属性值为 “inline”，是 “行内” 元素。
- (2) 行内元素有：a b span img input select strong (强调的语气)
- (1) 块级元素有：div ul ol li dl dt dd h1 h2 h3 h4...p
- (3) 知名的空元素：
 <hr> <input> <link> <meta>

三、link 和@import 的区别是？

- (1) link 属于 XHTML 标签，而@import 是 CSS 提供的;
- (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 只在 IE5 以上才能识别，而 link 是 XHTML 标签，无兼容问题;
- (4) link 方式的样式的权重 高于@import 的权重。

四、浏览器的内核分别是什么？

IE 浏览器的内核 Trident、Mozilla 的 Gecko、Chrome 的 Blink (WebKit 的分支)、Opera 内核原为 Presto，现为 Blink；

五、常见兼容问题

- (1)、png24 位的图片在 ie6 浏览器上出现背景，解决方案是做成 PNG8。
 - (2)、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。
 - (3)、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。
 - (4)、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 浏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*/
}
```

IE 下,可以使用获取常规属性的方法来获取自定义属性, 也可以使用 getAttribute()获取自定义属性; Firefox 下,只能使用 getAttribute()获取自定义属性. 解决方法:统一通过 getAttribute()获取自定义属性. * IE 下,event 对象有 x,y 属性,但是没有 pageX,pageY 属性; Firefox 下,event 对象有 pageX,pageY 属性,但是没有 x,y 属性. * 解决方法: (条件注释) 缺点是在 IE 浏览器下可能会增加额外的 HTTP 请求数. * Chrome 中文界面下默认会将小于 12px 的文本强制按照 12px 显示, 可通过加入 CSS 属性 -webkit-text-size-adjust: none; 解决. 超链接访问过后 hover 样式就不出现了 被点击访问过的超链接样式不在具有 hover 和 active 了解决方法是改变 CSS 属性的排列顺序: L-V-H-A : a:link {} a:visited {} a:hover {} a:active {}

六、前端页面由哪三层构成，分别是什么？作用是什么？

网页分成三个层次，即：结构层、表示层、行为层。

网页的结构层 (structural layer) 由 HTML 或 XHTML 之类的标记语言负责创建。标签，也就是那些出现在尖括号里的单词，对网页内容的语义含义做出了描述，但有些标签不包含任何关于如何显示有关内容的信息。例如，P 标签表达了这样一种语义：“这是一个文本段。”

网页的表示层 (presentation layer) 由 CSS 负责创建。CSS 对 “如何显示有关内容” 的问题做出了回答。

网页的行为层(behavior layer)负责回答“内容应该如何对事件做出反应”这一问题。
这是 Javascript 语言和 DOM 主宰的领域。

七、html 的意义

- (1)、HTML 指的是超文本标记语言(Hyper Text Markup Language)
- (2)、HTML 不是一种编程语言,而是一种标记语言(markup language)
- (3)、标记语言是一套标记标签(markup tag)
- (4)、HTML 使用标记标签来描述网页
- (5)、HTML 标记标签通常被称为 HTML 标签(HTML tag)。
- (6)、HTML 标签是由尖括号包围的关键词,比如 <html>
- (7)、HTML 标签通常是成对出现的,比如 和
- (8)、标签对中的第一个标签是开始标签,第二个标签是结束标签
- (9)、开始和结束标签也被称为开放标签和闭合标签

八、介绍一下 CSS 盒子模型

- (1)有两种,IE 盒子模型、标准 W3C 盒子模型;IE 的 content 部分包含了 border 和 padding;
- (2)盒模型:内容(content)、填充(padding)、边界(margin)、边框(border)。

九、CSS 选择符有哪些?哪些属性可以继承?优先级算法如何计算?CSS3 新增伪类有那些?

- 、id 选择器(# myid)
 - (2)、类选择器(.myclassname)
 - (3)、标签选择器(div, h1, p)
 - (4)、相邻选择器(h1 + p)
 - (5)、子选择器(ul < li)
 - (6)、后代选择器(li a)
 - (7)、通配符选择器(*)
 - (8)、属性选择器(a[rel = "external"])
 - (9)、伪类选择器(a: hover, li: nth - child)
 - (10)、可继承的样式: font-size font-family color, UL LI DL DD DT;
 - (11)、不可继承的样式: border padding margin width height ;
 - (12)、优先级就近原则,同权重情况下样式定义最近者为准;
 - (13)、载入样式以最后载入的定位为准;
 - 、优先级为:!important > id > class > tag //important 比 内联优先级高
- CSS3 新增伪类举例:
- 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 选择属于其父元素的唯一子元素的每个 <p> 元素。
 - 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。
 - :enabled :disabled 控制表单控件的禁用状态。
 - :checked 单选框或复选框被选中。

十、如何居中一个 DIV? 如何居中一个浮动元素

- (1)、给 div 设置一个宽度,然后添加 margin:0 auto 属性

```
div{  
  
width:200px;  
margin:0 auto;  
  
}
```

- (2)、居中一个浮动元素

确定容器的宽高 宽 500 高 300 的层
设置层的外边距

```
div {  
  
Width:500px ; height:300px;//高度可以不设  
Margin: -150px 0 0 -250px;  
position:relative;相对定位  
background-color:pink;//方便看效果  
left:50%;  
op:50%;  
  
}
```

列出 display 的值,说明他们的作用 position 的值 relative 和 absolute 定位的原点是?

- (1)、display 的值

- block 象块类型元素一样显示。
- none 缺省值。象行内元素类型一样显示。
- inline-block 象行内元素一样显示,但其内容象块类型元素一样显示。
- list-item 象块类型元素一样显示,并添加样式列表标记。

- (2)、作用域 position 的值

- *absolute 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。
- *fixed (老 IE 不支持) 生成绝对定位的元素,相对于浏览器窗口进行定位。
- *relative 生成相对定位的元素,相对于其正常位置进行定位。
- *static 默认值。没有定位,元素出现在正常的流中
- *(忽略 top, bottom, left, right z-index 声明)。
- *inherit 规定从父元素继承 position 属性的值。

十一、为什么要初始化 CSS 样式

- (1)、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。
- (2)、当然,初始化样式会对 SEO 有一定的影响,但鱼和熊掌不可兼得,但力求影响最小的情况下初始化。
- (3)、*最简单的初始化方法就是: * {padding: 0; margin: 0;} (不建议)

十二、对 BFC 规范的理解?

- (W3C CSS 2.1 规范中的一个概念,它决定了元素如何对其内容进行定位,以及与其他元素的关系和相互作用。)

十三、使用 CSS 预处理器吗？喜欢那个？

- (1)、Less Scss
- (2)、Less

十四、css 的基本语句构成是？

- (1)、选择器{属性 1:值 1;属性 2:值 2;.....}

十五、CSS 引入的方式有哪些? link 和@import 的区别是？

- (1)、使用 LINK 标签

将样式规则写在.css 的样式文件中，再以<link>标签引入。

```
<link rel=stylesheet type="text/css" href="example.css">
```

- (2)、使用@import 引入

跟 link 方法很像，但必须放在<STYLE>...</STYLE> 中

```
<STYLE TYPE="text/css">
<!--
    @import url(css/example.css);
-->
</STYLE>
```

- (3)、使用 STYLE 标签

将样式规则写在<STYLE>...</STYLE> 标签之中。

```
<STYLE TYPE="text/css">
<!--
body {color: #666;background: #f0f0f0;font-size: 12px;}
td,p {color:#c00;font-size: 12px;}
-->
</STYLE>
```

- (4)、使用 STYLE 属性

将 STYLE 属性直接加在个别的元件标签里，<元件(标签) STYLE="性质(属性)1: 设定值 1; 性质(属性)2: 设定值 2; ...}

使用标记引入样式

```
<span style="font:12px/20px #000000;">cnwebshow.com</span>
```

两者区别：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。@import 可以在 css 中再次引入其他样式表，比如可以创建一个主样式表，在主样式表中再引入其他的样式表，如：

main.css

```
@import "sub1.css";
@import "sub2.css";
```

这样做有一个缺点，会对网站服务器产生过多的 HTTP 请求，以前是一个文件，而现在却是两个或更多文件了，服务器的压力增大，浏览量大的网站还是谨慎使用。

十六、CSS 层叠是什么?介绍一下？

(1)、CSS 翻译过来叫做层叠样式表。运用到层叠的时候，主要就是靠 CSS 的组合与子选择器。去编辑样式。。它的作用是定义网页的外观（例如字体，颜色等等），它也可以和 javascript 等浏览器端脚本语言合作做出许多动态的效果。

(2)、CSS 指层叠样式表 (Cascading Style Sheets)

- 样式定义如何显示 HTML 元素
- 样式通常存储在样式表中
- 把样式添加到 HTML 4.0 中，是为了解决内容与表现分离的问题
- 外部样式表可以极大提高工作效率
- 外部样式表通常存储在 CSS 文件中
- 多个样式定义可层叠为一

十七、在 Css 中那个属性会影响 dom 读取文档流的顺序？

Float

十八、列举几个你工作中遇到的 CSS 和 JS 兼容问题，并写出解决方案

CSS：

(1)、每个浮动的 dl 上定义了 20px 的右边界，IE6.0 显示的结果是最后一个 dl 出现了双倍边界。

解决方法定义 display:inline-block;

(2)、盒子上定义了 200px 的高，IE6.0 超出时可以实现高度自适应，而其他浏览器没有。解决方法：height:auto!important;height:200px;min-height:200px

JS:

```
p.onclick=function(e){
    window.event
    alert(e.type);
}
```

(1)、引用事件对象时，IE9+ 及其他的浏览器可以直接引用，而 IE8 及之前的浏览器需要使用 window.event，解决方法：假设参数为 e，写为 e=e || window.event

(2)、使用 DOM2 级添加事件时 IE 需使用 attachEvent 非IE 使用 addEventListener。解决方法：判断支持哪个方法使用哪个方法

(3)、IE8.0+ 会将换行与换行之间的空白当作文本节点，IE6 和 IE7 不会。解决方法：获取所有节点遍历判断，如果是文本节点则删除，最后返回所有的元素节点。

十九、清除浮动？

clear : both、clear:left、clear : right、float : none

伪类：after 进行后续控制的高度位零的伪类层清除

采用 CSS overflow : auto 的方式撑高

采用 CSS overflow : hidden 的方式产生怪异适应

采用 display : table 将对象变成 table 形式

采用 div 标签，以及 css 的 clear 属性

采用 br 标签，以及 css 的 clear 属性

采用 br 标签，以及其自身 HTML 的 clear 属性

第二部分 JavaScript

一、JavaScript 原型，原型链？有什么特点？

- (1)、使用 Apply()和 call()，用于子类只继承父类构造函数内部的属性和方法时。
- (2)、使用原型链。用于子类只继承不重置父类的属性时
- (3)、使用 Apply()或 call()和原型混合的模式。用于继承父类所有的属性和方法，且有可能重置父类上的属性或方法时。
- (4)、对象冒充

二、eval 是做什么的？

它的功能是把对应的字符串解析成 JS 代码并运行；

应该避免使用 eval，不安全，非常耗性能（2 次，一次解析成 js 语句，一次执行）。

三、null，undefined 的区别？

- * Undefined 类型只有一个值，即 undefined。当声明的变量还未被初始化时，变量的默认值为 undefined。
- * Null 类型也只有一个值，即 null。null 用来表示尚未存在的对象，常用来表示函数企图返回一个不存在的对象。

四、写一个通用的事件侦听器函数

```
// event(事件)工具集，来源：github.com/markyun
markyun.Event = {
  // 页面加载完成后
  readyEvent : function(fn) {
    if (fn==null) {
      fn=document;
    }
    var oldonload = window.onload;
    if (typeof window.onload != 'function') {
      window.onload = fn;
    } else {
      window.onload = function() {
        oldonload();
        fn();
      };
    }
  },
  // 视能力分别使用 dom0||dom2||IE 方式 来绑定事件
  // 参数： 操作的元素,事件名称 ,事件处理程序
  addEvent : function(element, type, handler) {
    if (element.addEventListener) {
      //事件类型、需要执行的函数、是否捕捉
      element.addEventListener(type, handler, false);
    } else if (element.attachEvent) {
      element.attachEvent('on' + type, function() {
        handler.call(element);
      });
    } else {
      element['on' + type] = handler;
    }
  },
  // 移除事件
  removeEvent : function(element, type, handler) {
    if (element.removeEnentListener) {
      element.removeEnentListener(type, handler,
false);
    } else if (element.datachEvent) {
      element.detachEvent('on' + type, handler);
    } else {

```

```
        element['on' + type] = null;
      }
    },
    // 阻止事件（主要是事件冒泡，因为 IE 不支持事件捕获）
    stopPropagation : function(ev) {
      if (ev.stopPropagation) {
        ev.stopPropagation();
      } else {
        ev.cancelBubble = true;
      }
    },
    // 取消事件的默认行为
    preventDefault : function(event) {
      if (event.preventDefault) {
        event.preventDefault();
      } else {
        event.returnValue = false;
      }
    },
    // 获取事件目标
    getTarget : function(event) {
      return event.target || event.srcElement;
    },
    // 获取 event 对象的引用，取到事件的所有信息，确保随时能使用 event；
    getEvent : function(e) {
      var ev = e || window.event;
      if (!ev) {
        var c = this.getEvent.caller;
        while (c) {
          ev = c.arguments[0];
          if (ev && Event == ev.constructor) {
            break;
          }
          c = c.caller;
        }
      }
      return ev;
    }
  };
};
```

五、Node.js 的适用场景？

高并发、聊天、实时消息推送

六、介绍 js 的基本数据类型。

number,string,boolean,object,undefine

七、Javascript 如何实现继承？

通过原型和构造器

八、["1", "2", "3"].map(parseInt) 答案是多少？

[1, NaN, NaN] 因为 parseInt 需要两个参数 (val, radix) 但 map 传了 3 个 (element, index, array)

九、如何创建一个对象？

```
function Person(name, age) {
  this.name = name;
  this.age = age;
}
```

```
this.sing = function() { alert(this.name) }  
  
}
```

十、谈谈 This 对象的理解。

this 是 js 的一个关键字，随着函数使用场合不同，this 的值会发生变化。

但是有一个总原则，那就是 this 指的是调用函数的那个对象。

this 一般情况下：是全局对象 Global。 作为方法调用，那么 this 就是指这个对象

十一、事件是？IE 与火狐的事件机制有什么区别？ 如何阻止冒泡？

(1)、我们在网页中的某个操作（有的操作对应多个事件）。例如：当我们点击一个按钮就会产生一个事件。是可以被 JavaScript 侦测到的行为。

(2)、事件处理机制：IE 是事件冒泡、火狐是 事件捕获；

(3)、ev.stopPropagation();

十二、什么是闭包（closure），为什么要用它？

(1)、闭包是有权访问另一个函数作用域中的变量的函数。

(2)、闭包是个函数，而它“记住了周围发生了什么”。表现为由“一个函数”体中定义了“另一个函数”

(3)、“闭包”是一个表达式（一般是函数），它具有自由变量以及绑定这些变量的环境（该环境“封闭了”这个表达式）。

1. 闭包有权访问函数内部的所有变量。

2. 当函数返回一个闭包时，这个函数的作用域将会一直在内存中保存到闭包不存在为止。

```
function f() {  
    var rs = [];  
    for (var i=0; i < 10; i++) {  
        rs[i] = function() {  
            return i;  
        };  
    }  
    return rs;  
}  
  
var fn = f();  
for (var i = 0; i < fn.length; i++) {  
    console.log('函数 fn[' + i + ']()返回值:' + fn[i]());  
}
```

十三、如何判断一个对象是否属于某个类？

使用 instanceof

if(a instanceof Person){

```
    alert('yes');  
}
```

十四、new 操作符具体干了什么呢？

(1)、创建一个空对象，并且 this 变量引用该对象，同时还继承了该函数的原型。

(2)、属性和方法被加入到 this 引用的对象中。

(3)、新创建的对象由 this 所引用，并且最后隐式的返回 this。

```
var obj = {};  
obj.__proto__ = Base.prototype;  
Base.call(obj);
```

十五、Javascript 中，有一个函数，执行时对象查找时，永远不会去查找原型，这个函数是？

hasOwnProperty

十六、js 延迟加载的方式有哪些？

defer 和 async、动态创建 DOM 方式（用得最多）、按需异步载入 js

十七、异步加载的方式有哪些？

(1) defer，只支持 IE

(2) async：

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

十八、document.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

十九、call() 和 .apply() 的区别？

(1)、call, apply 都属于 Function.prototype 的一个方法，它是 JavaScript 引擎内实现的，因为属于 Function.prototype，所以每个 Function 对象实例(就是每个方法)都有 call, apply 属性。既然作为方法的属性，那它们的使用就当然是针对方法的了，这两个方法是容易混淆的，因为它们的作用一样，只是使用方式不同。

(2)、语法：foo.call(this, arg1,arg2,arg3) == foo.apply(this, arguments) == this.foo(arg1, arg2, arg3);

(3)、相同点：两个方法产生的作用是完全一样的。

(4)、不同点：方法传递的参数不同。

二十、Jquery 与 jQuery UI 有啥区别？

*jQuery 是一个 js 库，主要提供的功能是选择器，属性修改和事件绑定等等。

*jQuery UI 则是在 jQuery 的基础上，利用 jQuery 的扩展性，设计的插件。

提供了一些常用的界面元素，诸如对话框、拖动行为、改变大小行为等等

二十一、那些操作会造成内存泄漏？

- (1) 内存泄漏指任何对象在您不再拥有或需要它之后仍然存在。
- (2) 垃圾回收器定期扫描对象，并计算引用了每个对象的其他对象的数量。如果一个对象的引用数量为 0（没有其他对象引用过该对象），或对该对象的惟一引用是循环的，那么该对象的内存即可回收。
- (3) setTimeout 的第一个参数使用字符串而非函数的话，会引发内存泄漏。
- (4) 闭包、控制台日志、循环（在两个对象彼此引用且彼此保留时，就会产生一个循环）

二十二、你有哪些性能优化的方法？

- (1) 减少 http 请求次数：CSS Sprites, JS、CSS 源码压缩、图片大小控制合适；网页 Gzip，CDN 托管，data 缓存，图片服务器。
- (2) 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
- (3) 用 innerHTML 代替 DOM 操作，减少 DOM 操作次数，优化 javascript 性能。
- (4) 当需要设置的样式很多时设置 className 而不是直接操作 style。
- (5) 少用全局变量、缓存 DOM 节点查找的结果。减少 IO 读取操作。
- (6) 避免使用 CSS Expression(css 表达式)又称 Dynamic properties(动态属性)。
- (7) 图片预加载，将样式表放在顶部，将脚本放在底部 加上时间戳。
- (8) 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。

二十三、一个页面从输入 URL 到页面加载显示完成，这个过程中都发生了什么？（流程说的越详细越好）

查找浏览器缓存

DNS 解析、查找该域名对应的 IP 地址、重定向（301）、发出第二个 GET 请求

进行 HTTP 协议会话

客户端发送报头(请求报头)

服务器回馈报头(响应报头)

html 文档开始下载

文档树建立，根据标记请求所需指定 MIME 类型的文件

文件显示浏览器这边做的工作大致分为以下几步：

加载：根据请求的 URL 进行域名解析，向服务器发起请求，接收文件（HTML、JS、CSS、图象等）。

解析：对加载到的资源（HTML、JS、CSS 等）进行语法规则解析，建议相应的内部数据结构（比如 HTML 的 DOM 树，JS 的（对象）属性表，CSS 的样式规则等等）

二十四、什么是 DOM，什么是 BOM？以及它们的用法？

BOM 即浏览器对象模型，浏览器对象模型（BOM）使 JavaScript 有能力与浏览器“对话”，由于现代浏览器已经（几乎）实现了 JavaScript 交互性方面的相同方法和属性，因此常被认为是 BOM 的方法和属性。所有浏览器都支持 window 对象。它表示浏览器窗口。

所有 JavaScript 全局对象、函数以及变量均自动成为 window 对象的成员。

全局变量是 window 对象的属性。

全局函数是 window 对象的方法。

甚至 HTML DOM 的 document 也是 window 对象的属性之一：

- Window 对象其实是 BOM 中所有的对象都源自 window 对象，有 location 对象，history 对象，方法有 .Resize（），alert（），.confirm（），.prompt（），.open（），.close（），.setInterval（），.setTimeout（），.window.open（）- 打开新窗口
- .window.close（）- 关闭当前窗口
- .window.moveTo（）- 移动当前窗口
- .window.resizeTo（）- 调整当前窗口的尺寸

DOM 的文档对象模型，最顶级的对象是 document。可以 js 通过操作 DOM，就是一个接口，可以访问 html 的标准方法。要改变页面的某个东西，JavaScript 就需要获得对 HTML 文档中所有元素进行访问的入口。这个入口，连同对 HTML 元素进行添加、移动、改变或移除的方法和属性，都是通过文档对象模型来获得的（DOM）。

二十五、如何让 ie6,7,8，兼容 html5 的标签？

我一直使用公司里提供的 jqside 插件，里面就是把 html5 的标签放到字符串，用字符串的 split 的方法变为数组，用 while 的方法，变量递减，用 dom 的 createElement 方法进行再 ie678 里创建标签。

```
if( $.isIE678 ){
    var html5 = "abbr,article,aside,audio,canvas,datalist,details,dialog,eventsource,figure,footer,header,hgroup,mark,menu,meter,nav,output,progress,section,time,video".split( ',' );
    for( var i = html5.length; i-->0; ) document.createElement( html5[i] );
}
```

二十六、JavaScript 有哪些数据类型？

- (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

二十七、多浏览器检测通过什么？

- (1) navigator.userAgent
- (2) 不同浏览器的特性，如 addEventListener

二十八、常使用的库有哪些？

jQuery、Extjs、YUI、iScroll、Zepto

二十九、常用的前端开发工具有哪些？

irebug、Yslow、JSMIn(js 压缩工具)、Chrome Developer Tools、Sublime、hbuilder 等

三十、开发过什么应用或组件？

弹框、拖动、表单验证、移动 web 相册、移动端图片滑动等

三十一、研究过哪些 JS 框架的源代码？

jQuery、iScroll

三十二、列举几个你工作中遇到的 CSS 和 JS 兼容问题，并写出解决方案

CSS：

(1)、每个浮动的 dl 上定义了 20px 的右边界，IE6.0 显示的结果是最后一个 dl 出现了双倍边界。

解决方法定义 display:inline-block;

(2)、盒子上定义了 200px 的高，IE6.0 超出时可以实现高度自适应，而其他浏览器没有。解决方法：height:auto!important;height:200px;min-height:200px

JS:

```
p.onclick=function(e){
    window.event
    alert(e.type);
}
```

(1)、引用事件对象时，IE9+ 及其他的浏览器可以直接引用，而 IE8 及之前的浏览器需要使用 window.event，解决方法：假设参数为 e，写为 e=e || window.event

(2)、使用 DOM2 级添加事件时 IE 需使用 attachEvent 非 IE 使用 addEventListener。
解决方法：判断支持哪个方法使用哪个方法

(3)、IE8.0+ 会将换行与换行之间的空白当作文本节点，IE6 和 IE7 不会。解决方法：获取所有节点遍历判断，如果是文本节点则删除，最后返回所有的元素节点。

三十三、setInterval()和 setTimeout()分别是什么？它们有何实质性的区别(它们的作用是什么)

setTimeout 方法是超时调用，也就是在什么时间以后干什么。干完了就拉倒。

setInterval 方法间歇调用，则表示间隔一定时间反复执行某操作。

setInterval() 方法可按照指定的周期（以毫秒计）来调用函数或计算表达式。

setInterval() 方法会不停地调用函数，直到 clearInterval() 被调用或窗口被关闭。由 setInterval() 返回的 ID 值可用作 clearInterval() 方法的参数

三十四、如何解决跨域问题？解释一下它是如何工作的？(为什么它不是真正的 AJAX)

解决跨域问题的两种方法：

(1)、使用 iframe

(2)、JSONP

如何工作的？----word 文档中有答案

因为 JSONP 只是借助了 script 标签的 src 属性实现的跨域，并没有做异步请求，所以不是真正的 AJAX

三十五、使用 arguments.callee 写个递归

```
var s=(function(n){
    if(n>0) return n+arguments.callee(n-1);
    return 0;
})(10);
alert(s)
```

三十六、得出变量 baz 的值

```
var foo=10;
console.log(baz);
var bar={
    foo:9,
    foo1:function(){
        return this.foo;
    },
    foo2:(function(){
        console.log(this);
        return this.foo;
    })()
}
undefined;
```

三十七、new array(3)和 new array('3')有什么区别？

前者是创建一个数组且长度为三，但是仍然可以将元素存储在规定长度以外的，注意：这时长度会随之改变。

后者为创建一个数组，并赋值。

三十八、看程序

```
ar x=1,y=0,z=0;
function add(n){n=n+1;return n;}
y=add(x);
function add(n){n=n+3;return n;}
z=add(x);
```

console.log(y)

得出__4__

三十九、 == 和 === 的差别？

"==" 与 "===" 是不同的,一个是判断值是否相等,一个是判断值及类型是否完全相等

第三部分 JQuery,zepto 篇

一、JQuery 的源码看过吗？能不能简单说一下它的实现原理

二、jquery 中如何将数组转化为 json 字符串,然后再转化回来？

jQuery 中没有提供这个功能,所以你需要先编写两个 jQuery 的扩展：

```
$.fn.stringifyArray = function(array) {
    return JSON.stringify(array)
}
```

```
$.fn.parseArray = function(array) {
    return JSON.parse(array)
}
```

然后调用：

```
$("").stringifyArray(array)
```

三、针对 jQuery 的优化方法？

*基于 Class 的选择性的性能相对于 Id 选择器开销很大,因为需遍历所有 DOM 元素。

*频繁操作的 DOM,先缓存起来再操作。用 JQuery 的链式调用更好。

比如: var str=\$("#a").attr("href");

*for (var i = size; i < arr.length; i++) {}

for 循环每一次循环都查找了数组 (arr) 的.length 属性,在开始循环的时候设置一个变量来存储这个数字,可以让循环跑得更快:

```
for (var i = size, length = arr.length; i < length; i++) {}
```

四、JQuery 一个对象可以同时绑定多个事件,这是如何实现的？

使用 bind 和 on 都可意思实现利用 event 对象的 type 属性判断不同的事件类型

五、zepto 和 jquery 的区别

- 1) Zepto 对象 不能自定义事件
 - 2) Zepto 的选择器表达式: [name=value] 中 value 必须用 双引号 " or 单引号 ' 括起来
 - 3) Zepto 是根据标准浏览器写的,所以对于节点尺寸的方法只提供 width() 和 height(),省去了 innerWidth(),innerHeight(),outerWidth(),outerHeight()
 - 4) Zepto 的 each 方法只能遍历 数组,不能遍历 JSON 对象
 - 5) Zepto 的 animate 方法参数说明 :详情点击->
 - 6) zepto 的 jsonp callback 函数名无法自定义
 - 7) DOM 操作区别
 - 8) 事件触发区别
- 使用 jQuery 时 load 事件的处理函数 不会 执行
- 使用 Zepto 时 load 事件的处理函数 会 执行。

六、jQuery 库中的\$()是什么？

\$() 函数是 jQuery() 函数的别称,乍一看这很怪异,还使 jQuery 代码晦涩难懂。一旦你适应了,你会爱上它的简洁。\$() 函数用于将任何对象包裹成 jQuery 对象,接着你就会被允许调用定义在 jQuery 对象上的多个不同方法。你甚至可以将一个选择器字符串传入 \$() 函数,它会返回一个包含所有匹配的 DOM 元素数组的 jQuery 对象。这个问题我已经见过好几次被提及,尽管它非常基础,它经常被用来区分一个开发人员是否了解 jQuery。

七、网页上有 5 个 <div> 元素,如何使用 jQuery 来选择它们？

\$("div"), 这样会返回一个包含所有 5 个 div 标签的 jQuery 对象

八、jQuery 里的 ID 选择器和 class 选择器有何不同？

ID 选择器使用 ID 来选择元素,比如 #element1,而 class 选择器使用 CSS class 来选择元素。当你只需要选择一个元素时,使用 ID 选择器,而如果你想要选择一组具有相同 CSS class 的元素,就要用 class 选择器。在面试过程中,你有很大几率会被要求使用 ID 选择器和 class 选择器来写代码

九、如何在点击一个按钮时使用 jQuery 隐藏一个图片？

这是一个事件处理问题。jQuery 为按钮点击之类的事件提供了很好的支持。你可以通过以下代码去隐藏一个通过 ID 或 class 定位到的图片。你需要知道如何为按钮设置事件并执行 hide() 方法

十、\$(document).ready() 是个什么函数？为什么要用它？

ready() 函数用于在文档进入 ready 状态时执行代码。当 DOM 完全加载 (例如 HTML 被完全解析 DOM 树构建完成时) jQuery 允许你执行代码。使用\$(document).ready()

的最大好处在于它适用于所有浏览器，jQuery 帮你解决了跨浏览器的难题

十一、JavaScript window.onload 事件和 jQuery ready 函数有何不同？

JavaScript window.onload 事件和 jQuery ready 函数之间的主要区别是，前者除了要等待 DOM 被创建还要等到包括大型图片、音频、视频在内的所有外部资源都完全加载。如果加载图片和媒体内容花费了大量时间，用户就会感受到定义在 window.onload 事件上的代码在执行时有明显的延迟。

另一方面，jQuery ready() 函数只需对 DOM 树的等待，而无需对图像或外部资源加载的等待，从而执行起来更快。使用 jQuery \$(document).ready() 的另一个优势是，你可以在网页里多次使用它，浏览器会按它们在 HTML 页面里出现的顺序执行它们，相反对于 onload 技术而言，只能在单一函数里使用。鉴于这个好处，用 jQuery ready() 函数比用 JavaScript window.onload 事件要更好些。

十二、jQuery 里的 each() 是什么函数？你是如何使用它的？

each() 函数就像是 Java 里的一个 Iterator，它允许你遍历一个元素集合。你可以传一个函数给 each() 方法，被调用的 jQuery 对象会在其每个元素上执行传入的函数

十三、你是如何将一个 HTML 元素添加到 DOM 树中的？

可以用 jQuery 方法 appendTo() 将一个 HTML 元素添加到 DOM 树中。这是 jQuery 提供的众多操控 DOM 的方法中的一个。你可以通过 appendTo() 方法在指定的 DOM 元素末尾添加一个现存的元素或者一个新的 HTML 元素

十四、你能用 jQuery 代码选择所有在段落内部的超链接吗？

选择所有嵌套在段落（<p>标签）内部的超链接（<a>标签）

十五、\$(this) 和 this 关键字在 jQuery 中有何不同？

\$(this) 返回一个 jQuery 对象，你可以对它调用多个 jQuery 方法，比如用 text() 获取文本，用 val() 获取值等等。而 this 代表当前元素，它是 JavaScript 关键词中的一个，表示上下文中的当前 DOM 元素。你不能对它调用 jQuery 方法，直到它被 \$() 函数包裹，例如 \$(this)。

十六、你如何使用 jQuery 来提取一个 HTML 标记的属性 例如. 链接的 href?

```
$(this).attr('href')
```

十七、你如何使用 jQuery 设置一个属性值？

attr()方法和 jQuery 中的其它方法一样，能力不止一样。如果你在调用 attr()的同时带上一个值 例如. attr(name, value), 这里 name 是属性的名称，value 是属性的新值。

十八、jQuery 中 detach() 和 remove() 方法的区别是什么？

尽管 detach() 和 remove() 方法都被用来移除一个 DOM 元素，两者之间的主要不同在于 detach() 会保持对过去被解除元素的跟踪，因此它可以被取消解除，而 remove() 方法则会保持过去被移除对象的引用。你也还可以看看 用来向 DOM 中添加元素的 appendTo() 方法。

十九、你如何利用 jQuery 来向一个元素中添加和移除 CSS 类？

通过利用 addClass() 和 removeClass() 这两个 jQuery 方法。动态的改变元素的 class 属性可以很简单例如. 使用类 ".active"来标记它们的未激活和激活状态，等等。

二十、使用 CDN 加载 jQuery 库的主要优势是什么？

除了报错节省服务器带宽以及更快的下载速度这许多的好处之外，最重要的是，如果浏览器已经从同一个 CDN 下载类相同的 jQuery 版本，那么它就不会再去下载它一次。因此今时今日，许多公共的网站都将 jQuery 用于用户交互和动画，如果浏览器已经有

了下载好的 jQuery 库，网站就能有非常好的展示机会。

二十一、jQuery.get() 和 jQuery.ajax() 方法之间的区别是什么？

ajax() 方法更强大，更具可配置性，让你可以指定等待多久，以及如何处理错误。get() 方法是一个只获取一些数据的专门化方法。

二十二、jQuery 中的方法链是什么？使用方法链有什么好处？

方法链是对一个方法返回的结果调用另一个方法，这使得代码简洁明了，同时由于只对 DOM 进行了一轮查找，性能方面更加出色。

二十三、你要是在一个 jQuery 事件处理程序里返回了 false 会怎样？

这通常用于阻止事件向上冒泡。

二十四、哪种方式更高效：document.getElementById("myId") 还是 \$("#myId")？

第一种，因为它直接调用了 JavaScript 引擎。

第四部分 HTML5+CSS3 篇

一、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？

(1) 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。

(2) 绘画 canvas

(3) 用于媒介回放的 video 和 audio 元素

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

(5) sessionStorage 的数据在浏览器关闭后自动删除

(6) 语义化更好的内容元素，比如 article、footer、header、nav、section

(7) 表单控件，calendar、date、time、email、url、search

(8) 新的技术 webworker, websocket, Geolocation

* 移除的元素

(1) 纯表现的元素：basefont、big、center、font、s、strike、tt、u；

(2) 对可用性产生负面影响的元素：frame、frameset、noframes；

支持 HTML5 新标签：

(1) IE8/IE7/IE6 支持通过 document.createElement 方法产生的标签，

(2) 可以利用这一特性让这些浏览器支持 HTML5 新标签，

如何处理

(1) 浏览器支持新标签后，还需要添加标签默认的样式：

(2) 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

二、语义化的理解？

- (1) 用正确的标签做正确的事情！
- (2) html 语义化就是让页面的内容结构化，便于对浏览器、搜索引擎解析；
- (3) 在没有样式 CCS 情况下也以一种文档格式显示，并且是容易阅读的。
- (4) 搜索引擎的爬虫依赖于标记来确定上下文和各个关键字的权重，利于 SEO。
- (5) 使阅读源代码的人对网站更容易将网站分块，便于阅读维护理解。

三、HTML5 的离线储存？

localStorage 长期存储数据，浏览器关闭后数据不丢失；

sessionStorage 数据在浏览器关闭后自动删除。

四、描述一段语义的 html 代码吧

(HTML5 中新增加的很多标签（如：<article>、<nav>、<header>和<footer>等）

就是基于语义化设计原则）

```
< div id="header">

< h1>标题< /h1>

< h2>专注 Web 前端技术< /h2>

< /div>
```

五、iframe 有那些缺点？

- (1) iframe 会阻塞主页面的 Onload 事件；
- (2) iframe 和主页面共享连接池，而浏览器对相同域的连接有限制，所以会影响页面的并行加载。
- (3) 使用 iframe 之前需要考虑这两个缺点。如果需要使用 iframe，最好是通过 javascript
- (4) 动态给 iframe 添加 src 属性值，这样可以可以绕开以上两个问题。

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

- (1) cookie 在浏览器和服务器间来回传递。 sessionStorage 和 localStorage 不会
- (2) sessionStorage 和 localStorage 的存储空间更大；
- (3) sessionStorage 和 localStorage 有更多丰富易用的接口；
- (4) sessionStorage 和 localStorage 各自独立的存储空间；

七、CSS3 有哪些新特性？

CSS3 实现圆角 (border-radius:8px) ， 阴影 (box-shadow:10px) ， 对文字加特效 (text-shadow、) ， 线性渐变 (gradient) ， 旋转 (transform)

transform:rotate(9deg) scale(0.85,0.90) translate(0px,-30px) skew(-9deg,0deg);//旋转,缩放,定位,倾斜

增加了更多的 CSS 选择器 多背景 rgba

八、手机浏览器独有的三个事件？

onTouchmove , ontouchend , ontouchstart , ontouchcancel。

九、为什么要用 Zepto?

query 适用于 PC 端桌面环境，桌面环境更加复杂，jquery 需要考虑的因素非常多，尤其在兼容性上面，相对于 PC 端，移动端的发杂都远不及 PC 端,手机上的带宽永远比不上 pc 端。pc 端下载 jquery 到本地只需要 1~3 秒 (90+K) ，但是移动端就慢了很多，2G 网络下你会看到一大片空白网页在加载，相信用户第二次就没打开的欲望了。zepto 解决了这个问题，只有不到 10K 的大小，2G 网络环境下也毫无压力，表现不逊色于 jquery。所以移动端开发首选框架，个人推荐 [zepto.js](#)。

jq mobi , zepoto 手机跨平台的手机框架

十、如何区分多个终端 (zepto) ？

用 detece 来判断：

```
// general device type

$.os.phone

$.os.tablet


// specific OS

$.os.ios

$.os.android

$.os.webos

$.os.blackberry

$.os.bb10

$.os.rimtabletos


// specific device type

$.os.iphone

$.os.ipad

$.os.touchpad

$.os.kindle


// specific browser

$.browser.chrome

$.browser.firefox

$.browser.silk

$.browser.playbook


// Additionally, version information is available as well.

// Here's what's returned for an iPhone running iOS 6.1.

!!$.os.phone // => true

!!$.os.iphone // => true

!!$.os.ios // => true

!!$.os.version // => "6.1"


!!$.browser.version // => "536.26"
```

十一、border-image 原理？

- (1) 调用边框图片：border-image 的 url 属性，通过相对或绝对路径链接图片。
- (2) 边框图片的建裁：border-image 的数值参数剪裁边框图片，形成九宫格。
- (3) 剪裁图片的边框：边框图片被切割成 9 部分，以——对应的关系放到 div 边框的九宫格中，然后再压缩 (或拉伸) 至边框 (border-width 或 border-image-width) 的宽度大小。
- (4) 执行重复属性：被填充至边框九宫格四个角落的的边框图片是不执行重复属性的。上下的九宫格执行水平方向的重复属性 (拉伸或平铺) ，左右的格子执行垂直方向的重复属性，而中间的那个格子则水平重复和垂直方向的重复都要执行。
- (5) 完成绘制，实现效果

十二、自定义手机 UI 组件？如何实现的？

Input,button,radio,checkbox。

把本身的 input 隐藏掉，给后面的 label 进行样式，并且用 label 的 for 属性，去指定

input 的 id。去点击 label 的时候，css3 的 :checked 和 :disabled 去确定状态，样式用 css 精灵去排版。

十三、图片切换的实现思路？

以全局监听的方式通过 a 标签的锚点进行 view 动态切换页面，只要把 a 标签带有 id 的 href 属性的值指到锚点，用 CSS3 的动画进行切换页面。

十四、HTML5 都有哪些新的 JS API？

1. 二维绘图 API，可以用在一个新的画布（Canvas）元素上以呈现图像、游戏图形或者其他运行中的可视图形。
2. 一个允许 web 应用程序将自身注册为某个协议或 MIME 类型的 API。
3. 一个引入新的缓存机制以支持脱机 web 应用程序的 API。
4. 一个能够播放视频和音频的 API，可以使用新的 video 和 audio 元素。
5. 一个历史纪录 API，它可以公开正在浏览的历史纪录，从而允许页面更好地支持 AJAX 应用程序中实现后退功能。
6. 跨文档/跨域的消息传递，它提供了一种方式，使得文档可以互相通信而不用考虑它们的来源域，在某种程度上，这样的设计是为了防止跨站点的脚本攻击。
7. 一个支持拖放操作的 API，用它可以与 draggable 特性相关联。
8. 一个支持编辑操作的 API，用它可以与一个新的全局 contenteditable 特性相关联。
9. 一个新的网络 API，它支持 web 应用程序在本地网络上互相通信，并在它们的源服务器上维持双向的通信。
10. 使用 JavaScript API 的键/值对实现客户端的持久化存储，同时支持嵌入的 SQL 数据库。
11. 服务器发送的事件，通过它可以与新的事件源（event-source）元素关联，新的事件源元素有利于与远程数据源的持久性连接，而且极大地消除了 Web 应用程序中对轮询的需求。

十五、新的 HTML5 文档类型和字符集是？

HTML5 文档类型很简单：<!doctype html>

HTML5 使用 UTF-8 编码示例：<meta charset=" UTF-8">

十六、HTML5 Canvas 元素有什么用？

Canvas 元素用于在网页上绘制图形，该元素标签强大之处在于可以直接在 HTML 上进行图形操作，

十七、什么是 HTML5？

（1）HTML5 是最新的 HTML 标准，他的主要目标是提供所有内容而不需要任何的像 flash，silverlight 等的额外插件，这些内容来自动画，视频，富 GUI 等

（2）HTML5 是万维网联盟（W3C）和网络超文本应用技术工作组（WHATWG）之间合作输出的

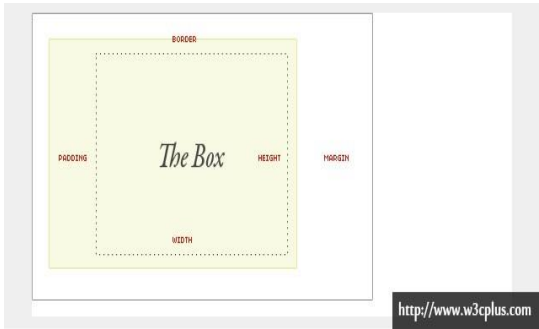
十八、HTML5 中的 datalist 是什么？

HTML5 中的 Datalist 元素有助于提供文本框自动完成特性，如下图所示：

十九、什么是 CSS 的盒子模型？这里有哪些 CSS 属性？

CSS 盒模型的基本原则是理解布局和大小等。它主要由下面部分组成：

- width 和 height（或缺乏，默认值和里面的内容）
- padding
- border



margin 对于盒模型来说是相关的但不严格。我将额外的提到这一点，但不包括。

二十、Sass、LESS 和 Stylus 是什么？大家为什么要使用他们？如何使用 Compass 相关的 Sass？

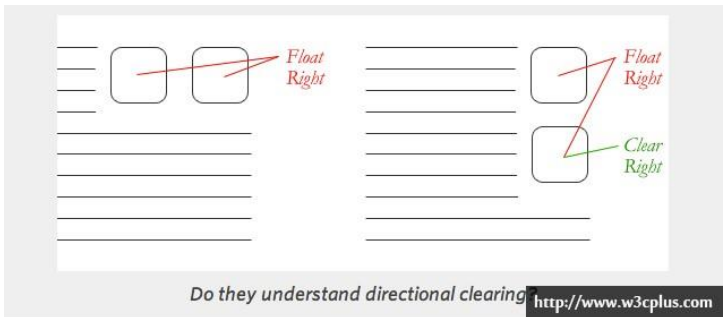
他们是 CSS 预处理器。他是 CSS 上的一种抽象层。他们是一种特殊的语法/语言编译成 CSS。他们使用 CSS 更容易管理，诸如变量、混合，以及用来处理前端（在其他事物之中）。他们容易做到最佳，像连接和压缩 CSS。

二十一、清除浮动的各种技巧是什么？

Float 是很普便的。众所周知，可以使用他制作布局和网格系统，并且兼容所有浏览器。大家都知道，浮动会让元素塌陷。即被浮动元素的父元素不具有高度。例如一个父元素包含了浮动元素，它将塌陷具有零高度。你可以按下面的方法处理：

- 使用 clearfix(知道 Micro clearfix 可以加分)
- 父元素浮动也是一种方法
- 父元素使用 overflow 属性并设置 visible 值

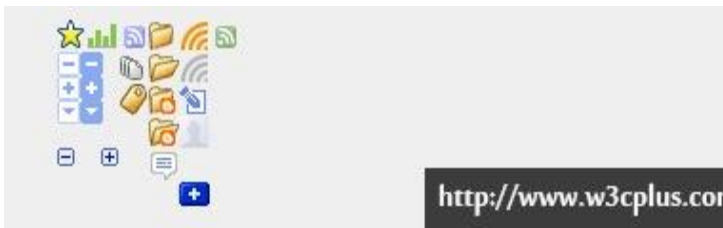
创建一个 BFC 可以加分。可能要嵌套一些东西，如：<br style="clear: both;">。



作为奖励的问题，你可以向他们问比较使用 inline-block 还是 float 创建一个网格。好的答案：无论哪种方式有问题。使用 inline-block，你需要处理空白问题，使用 float 你需要清除浮动。

二十二、Sprites 是什么和为什么要使用他们？你是如何创建他们？有可能替代 Sprites 的方法是什么？

Sprites 本质是多张图片拼成一张图片。为了提高性能的原因，使用它们。一般来说，最慢一个网站可以做的是请求一个资源。一个网站的请求需要越少，速度越快。快就等于好。多个请求合成一个请求就是好。



问他们如何让 Sprites 就搞定，他们其实非常熟悉他们。手动创建 Sprites 当然是一种可能性，但它并不是非常高效的。有一些工具可以提高效率，例如：SpriteCow、

SpriteMe、Spriting with Compass 或者 Grunticon。听到这些都是非常有趣的。

Sprites 是栅格图像。当问到替代品时，好的答案可能都是与事实相关，Sprites 通常为 Icons 和 Icons 不是需要光栅。SVG 图标，字体图标，字符编码等等。

二十三、CSS 中可访问是什么？

- 1) 隐藏内容在这里是一个重点。如果你们都试图在视觉上和他的内容隐藏不见，这里仅接受 display:none;。
- 2) CSS 控制颜色，因此颜色可访问性相关。专注样式用 CSS 直接控制也非常重要。
- 3) 与 HTML 和 JavaScript 相关的，有更多的可访问性的东西，所以提到那个东西是比较厉害的，但我认为这是集中考验 CSS 的一个有趣的题目。

二十四、inline、inline-block 和 block 区别是什么？

在不改变行内元素，能将他们的细节描述清楚可以加分。

二十五、什么是响应式设计？

它是关于网站的制作或网页制作的工作。不同的设备有不同的尺寸和不同的功能。响应式设计是让所有的人能在这些设备上让网站运行正常。一部分是媒体查询和不同的视觉效果。一部分是不同的资源（如不同的 Javascript 来处理触摸与点击自动适应屏幕的对比）。

二十六、你做过网格布局吗？你有什么想法吗？

为什么他们需要一个网格系统？他们是如何创建一个网格系统？是自己创建还是使用一个网格工具？他们喜欢网格工具吗？他们喜欢用什么样的类名定义网格？他们是移动端先行还是桌面系统先行？网格是一个帮助还是一个障碍呢？网格能自动运用到任何项目中吗？

二十七、SVG 的好处是什么？

SVG 是一种基于矢量的图像格式。这是一个有效的格式（文件小）。你可以对他们进行缩放和任何尺寸下保持清晰（提及到小尺寸下光栅占优势可以加分）。你可以使用 CSS 和 JavaScript 来修饰他们以及使用 SVG 指定的滤镜做很多事情，比如说模糊效果。

二十八、你是负责编辑设计的，使用非标准的 Web 字体，你会怎么做？

一个并非主要的方法@font-face，让他们谈论下它是如何工作的？谈论一下他如何是使用 CSS 技术以及谈论是如何服务，提供了体和让它变得更容易等等。Google 字体，Typekit，Font Deck，Cloud Typography 等等。

二十九、使固定宽度设计为流体

这有一个设计。它具有一个“800px”的固定宽度。你有更好的方法，让他适合任何屏幕大小吗？

第五部分 AJAX、JSON 篇

一、JSON 的了解？

JSON(JavaScript Object Notation) 是一种轻量级的数据交换格式。

它是基于 JavaScript 的一个子集。数据格式简单，易于读写，占用带宽小

```
{'age':12,'name':'back'}
```

二、Ajax 是什么？

是指一种创建交互式网页应用的网页开发技术

三、同步和异步的区别？

同步需要等待返回结果才能继续，异步不必等待，一般需要监听异步的结果

同步是在一条直线上的队列，异步不在一个队列上 各走各的

四、如何解决跨域问题？

jsonp、iframe、window.name、window.postMessage、服务器上设置代理页面

五、AJAX 应用和传统 Web 应用有什么不同？

在传统的 Javascript 编程中，如果想得到服务器端数据库或文件上的信息，或者发送客户端信息到服务器，需要建立一个 HTML form 然后 GET 或者 POST 数据到服务器端。用户需要点击“Submit”按钮来发送或者接受数据信息，然后等待服务器响应请求，页面重新加载。因为服务器每次都会返回一个新的页面，所以传统的 web 应用有可能很慢而且用户交互不友好。使用 AJAX 技术，就可以使 Javascript 通过 XMLHttpRequest 对象直接与服务器进行交互。通过 HTTP Request，一个 web 页面可以发送一个请求到 web 服务器并且接受 web 服务器返回的信息(不用重新加载页面)，展示给用户的还是通一个页面，用户感觉页面刷新，也看不到 Javascript 后台进行的发送请求和接受响应。

六、AJAX 是什么? AJAX 的交互模型(流程)? AJAX 跨域的解决办法？

Asynchronous JavaScript and XML (异步 JavaScript 和 XML)，是一种创建交互式网页应用的网页开发技术。简单的说它是多种技术的组合，目的就是让前台的数据交互变得更快，不用刷新页面就可以完成数据的更新。

优点很明显，利于用户体验，不会打断用户的操作，在不刷新页面的情况下更新内容，减小服务器压力也是它很硬性的一个优点；而缺点，除了倍受追捧的 SEO 问题，安全问题、前进后退（这个虽然可以用其他方法解决，但 AJAX 本身的机制还是没变）、破坏程序的异常机制以及对新兴手持设备支持不完美的问题都是它现存的一些缺点。

readyState 五种状态：

- 请求未初始化，还没有调用 open()。
- 请求已经建立，但是还没有发送，还没有调用 send()。
- 请求已发送，正在处理中（通常现在可以从响应中获取内容头）。
- 请求在处理中；通常响应中已有部分数据可用了，没有全部完成。
- 响应已完成；您可以获取并使用服务器的响应了。

status 常见的几种状态

- 100——客户必须继续发出请求
- 101——客户要求服务器根据请求转换 HTTP 协议版本
- 200——成功
- 201——提示知道新文件的 URL
- 300——请求的资源可在多处得到
- 301——删除请求数据
- 302-----缓存问题
- 404——没有发现文件、查询或 URI
- 500——服务器产生内部错误

七、如何解决跨域问题？解释一下它是如何工作的？(为什么它不是真正的 AJAX)

解决跨域问题的两种方法：

- (1)、使用 iframe
- (2)、JSONP

如何工作的？----word 文档中有答案

因为 JSONP 只是借助了 script 标签的 src 属性实现的跨域，并没有做异步请求，所以不是真正的 AJAX

八、AJAX 的全称是什么？

AJAX 的全称是 Asynchronous JavaScript And XML。

九、AJAX 的优缺点都有什么？

- 1．最大的一点是页面无刷新，用户的体验非常好。

- 2、使用异步方式与服务器通信，具有更加迅速的响应能力。
- 3、可以把以前一些服务器负担的工作转嫁到客户端，利用客户端闲置的能力来处理，减轻服务器和带宽的负担，节约空间和宽带租用成本。并且减轻服务器的负担，ajax 的原则是“按需取数据”，可以最大程度的减少冗余请求，和响应对服务器造成的负担。

- 4、基于标准化的并被广泛支持的技术，不需要下载插件或者小程序。

ajax 的缺点

- 1、ajax 不支持浏览器 back 按钮。
- 2、安全问题 AJAX 暴露了与服务器交互的细节。
- 3、对搜索引擎的支持比较弱。
- 4、破坏了程序的异常机制。
- 5、不容易调试。

使用 Ajax 的最大优点，就是能在不更新整个页面的前提下维护数据。这使得 Web 应用程序更为迅捷地回应用户动作，并避免了在网络上发送那些没有改变过的信息。

对应用 Ajax 最主要的缺点就是，它可能破坏浏览器后退按钮的正常行为

因为 Ajax 中采用了 xml 技术，所以在 Ajax 中也可能问到 XML 的问题

十、Ajax 的工作原理？

Ajax 的核心是 JavaScript 对象 XmlHttpRequest。该对象在 Internet Explorer 5 中首次引入，它是一种支持异步请求的技术。简而言之，XmlHttpRequest 使您可以使用 JavaScript 向服务器提出请求并处理响应，而不阻塞用户。在创建 Web 站点时，在客户端执行屏幕更新为用户提供了很大的灵活性。

十一、Ajax 和 javascript 的区别？

javascript 是一种在浏览器端执行的脚本语言，Ajax 是一种创建交互式网页应用的开发技术

十二、将项目的 XML 数据转为 JSON 数据

```
<?xml version="1.0" encoding="utf-8" ?>
<users>
  <user>
    <username>张三 XML</username>
    <age>33</age>
  </user>
  <user>
    <username>李四 XML</username>
    <age>34</age>
  </user>
</users>
```

十三、Ajax 的最大的特点是什么。

Ajax 可以实现动态不刷新（局部刷新）

就是能在不更新整个页面的前提下维护数据。这使得 Web 应用程序更为迅捷地回应用户动作，并避免了在网络上发送那些没有改变过的信息。

十四、Ajax 技术体系的组成部分有哪些？

HTML，css，dom，xml，xmlHttpRequest，javascript

十五、AJAX 请求总共有多少种 CALLBACK

Ajax 请求总共有八种 Callback

onSuccess

onFailure

onUninitialized

onLoading

onLoaded

onInteractive

onComplete

onException

十六、介绍一下 XMLHttpRequest 对象的常用方法和属性（回答的越多越好）

open(“method”，“URL”）建立对服务器的调用，第一个参数是 HTTP 请求方式可以为 GET，POST 或任何服务器所支持的您想调用的方式。

第二个参数是请求页面的 URL。

send()方法，发送具体请求

abort()方法，停止当前请求

readyState 属性 请求的状态 有 5 个可取值 0=未初始化，1=正在加载

2=以加载，3=交互中，4=完成

responseText 属性 服务器的响应，表示为一个串

responseXML 属性 服务器的响应，表示为 XML

status 服务器的 HTTP 状态码，200 对应 ok 400 对应 not found

十七、GET 和 POST 的区别？什么时候用 GET 什么时候用 POST？

区别：

（1）get 请求需注意缓存问题，post 请求不需担心这个问题

（2）post 请求必须设置 Content-Type 值为 application/x-form-www-urlencoded

（3）发送请求时，因为 get 请求的参数都在 url 里，所以 send 函数发送的参数为 null，而 post 请求在使用 send 方法时，却需赋予其参数；

（4）POST 比 GET 发送的数据量更大

使用：GET 一般用在获取服务端数据，POST 一般用于向服务器发送数据

十八、什么是 JSONP

JSONP 是 JSON with Padding 的略称。它是一个非官方的协议，它允许在服务端集成 Script tags 返回至客户端，通过 javascript callback 的形式实现跨域访问，这仅仅是 JSONP 简单的实现形式）

- 1、HTML 中定义表格的宽度用 80px 与 80%的区别是什么？

px 表示像素，%表示占整个页面宽度的百分比。

- 2、简述 DIV 元素和 SPAN 元素的区别。

DIV 默认情况下是分行显示，SPAN 在同行显示。

- 3、在 HTML 页面布局中，position 的值有几种？默认值又是什么？

1）position 的五种值分别是 static，relative，absolute，fixed，inherit

2）position 的缺省值是 static

- 4、怎样使一个宽为 200px 和高为 200px 的层垂直居中于浏览器中？写出 CSS 样式代码。

这里使用百分比和绝对定位，与外补丁负值的方法，负值的大小为其自身宽度高度一半。

- 5、CSS 规范中，.（句点）后面跟一个名称代表什么含义？#（井号）后面跟一个名称代表什么含义？如果要对一个元素设置 CSS 样式（内嵌样式），应将 CSS 样式写在它的什么属性内？

1）.（句点）后面跟一个名称表示文档中所有 class 属性值包含这个名称的应用其样式，

2) # (井号) 后面跟个名称表示文档中 ID 为此名称的元素应用其样式。

3) CSS 样式写在 style 属性内。

6、CSS 具体含义是什么？在网页制作中为什么要使用 CSS 技术？

CSS(Cascading Style Sheets)也叫层叠样式表，它是用来控制网页中一个或多个元素的外观。在网页制作中，使用样式表可以减少很多具有相同外观元素的重复设置工作，从而给网页的设计和维持带来了很大的方便；具体应用是：改变样式表中元素显示样式的定义，则在使用该样式表的所有网页中相应元素的显示样式都会自动发生改变。

7、CSS+DIV 开发 Web 页面的优势有哪些？

1) CSS+DIV，这个网页设计模式中，DIV 承担了网页的内容，CSS 承担了网页的样式。这样就使网页的内容和样式的分离开来。有利于页面的维护升级。

2) 有助于提高搜索引擎亲和力（快速找到需要的数据，而不是像在 TABLE 中一层层的查找）

3) 有助于页面的重构(换皮肤如 blog，直接套用另外一套样式就可以实现，而不用改动网页脚本。)

8、一个 HTML 文档必须包含三个元素，它们是 head、body 和哪个？

html

9、HTML 中文档内联注释应该是以结尾

10、引用 CSS 样式，有哪几种方法？

1) 导入一个已经存在的 CSS 文件：

2) 直接在 HARD 里声明一个 CSS 代码段：

3) 直接在网页元素里声明使用 CSS 样式：style=" color:red;"

11、JavaScript 中 setTimeout 与 setInterval 两个函数的区别？

setTimeout 只执行一次，setInterval 没有次数限制。

12、编写 JavaScript 脚本生成 1-6 之间的整数？

```
var NowFrame;
```

```
NowFrame=Math.random()*6+1 //随机生成一个 1-6 之间小数
```

```
NowFrame=parseInt(NowFrame) //把 1-6 之间的小数转化为整数
```

13、设置什么属性可以让一个 DIV 隐藏和显示？值分别是什么？

1) 设置 display 属性的值为 none 时，该层隐藏。

2) 设置 display 属性的值为 block 时，该层显示。

14、在 JavaScript 脚本中，isNaN 的作用是什么？

isNaN 的作用是判断值是否为数字

15、JavaScript 中获取某个元素有哪几种方式？

document.getElementById();此方法用于获得名为 ID 值的元素

document.myform.xxx;此方法按照层次结构来获取

document.getElementsByTagName();此方法用于获得所有的名字相同的元素

16、Ajax 的优缺点都有什么？

Ajax 的优点：

1) 页面无刷新，用户体验非常好。

2) 使用异步方式与服务器通信，具有更加迅速的响应能力。

3) 可以把一些服务器负担的工作转到客户端，利用客户端闲置的能力来处理，减轻服务器负担，节约空间和宽带租用成本。并且减轻服务器的负担，ajax 的原则是“按需取数据”，可以最大程度的减少冗余请求和响应对服务器造成的负担。

4) 基于标准化并被广泛支持的技术，不需要下载插件或者小程序。

Ajax 的缺点：

1) Ajax 不支持浏览器 back 按钮。

2) 安全问题，Ajax 暴露了与服务器交互的细节。

3) 对搜索引擎的支持比较弱。

4) 破坏了程序的异常机制。

5) 不容易调试。

17、简述一下 Ajax 的工作原理。

Ajax 的核心是 JavaScript 对象 XMLHttpRequest。该对象在 Internet Explorer 5 中首次引入，它是一种支持异步请求的技术。简而言之，XMLHttpRequest 使您可以使用 JavaScript 向服务器提出请求并处理响应，而不阻塞用户。

18、JavaScript 中的数据类型有几种？分别是什么？

JavaScript 中的数据类型有 7 种，分别是：

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

19、在 JavaScript 脚本中，表达式 2+5+" 8"得到的结果是什么？并说明原因。

因为 2 和 5 是数字类型，“8”是字符串类型，所以相加得到的结果是 78。

20、编写 JavaScript 脚本隐藏一个超级链接真实的地址。

代码示例如下：

```
onMouseOut='window.status="Done";'href="http://www.affiliate-url-here.com" target="_blank">Link Text Here
```

21.js 中定义变量问题，请问定义变量时有 var 关键字和没有 var 关键字有什么差别

在函数内部，有 var 和没 var 声明的变量是不一样的。

有 var 声明的是局部变量，没 var 的，声明的全局变量，所以可以借此向外暴露接口东东。 \

在全局作用域内声明变量时，

有 var 和没 var 看起来都一样，

我们知道，声明的全局变量，就是 window 的属性，

使用 var 语句重复声明语句是合法且无害的。

如果重复声明且带有赋值，那么就一般的赋值语句没差别。

如果尝试读取没有声明过的变量，Js 会报错。

JavaScript 的函数作用域内，声明的变量或内部函数，

在函数体内都是可见的。意味着，函数在定义之前可能已经可用。

函数定义有两种方式，一种是函数定义表达式，一种是函数声明语句。