

# HTML



## H5学习资料

## 第一部分 HTML&CSS整理答案

6

1. 标签上title与alt属性的区别是什么? 6
2. <p>标签内的文字是什么颜色的? 6
3. css单位px,em,rem的区别。 6
4. 以下哪些是css伪类? (c,d) after和before是伪元素 6
5. 以下不是html5元素的有哪些? (b,c) 6
6. 下述有关css属性position的属性值的描述。语法错误的是? b 6
7. CSS实现隐藏页面元素的方法有哪些? 6
8. xhtml和html有什么区别? 6
9. 每个html文件里开头都有个Doctype。这是代表了什么意思? 7
10. strong与em的异同? 7
11. 行内元素有哪些? 块级元素有哪些? 7
12. css选择符有哪些? 哪些属性可以继承? 优先级算法如何计算? 8
13. HTML5文档类型和字符集是? 8
14. HTML5中canvas是什么? datalist是什么? 8
15. 旧版本的IE浏览器(如:IE6,IE7)的默认盒模型,W3C定义的盒模型,二者之间有什么不同? 8
16. 以下HTML代码片段,有几种方法让该div在浏览器中横向居中? 8
17. 以下css代码片段,在IE6,IE7,IE8,Firefox,Chrome下分别呈现什么颜色? 8
18. 如何做到让一个div隐藏但是却在html流中占位? 9
19. 下面的输出是? 9
20. 阅读以下代码回答:hello world 是什么颜色? 9
21. 请用CSS实现4个div的水平排列占满一行,要求4个div宽度相同,有10px的padding值。 9
22. <!DOCTYPE html>中DOCTYPE有什么作用, 去掉会有什么影响? 9
23. HTML行内元素有哪些, 块级元素有哪些? 画出CSS盒模型。 9
24. 内联和important哪个优先级更高? 9
25. 清除浮动的几种方式? 9
26. CSS3新增伪类选择器有哪些? 9
27. 用CSS3编写: div旋转90度, 0.5秒后变圆, 0.5秒后方法两倍。 10
28. 实现一个多列的等高布局。 10
29. css选择器中, 以下排序正确的为 (a) 10
30. 如何解决在移动端1px的线比实际粗 10
31. 如何计算移动端rem的值 10
32. zoom是什么,有什么用? 10
33. HTML5 为什么只需要写<! DOCTYPE HTML>? 10
34. 页面导入样式时, 使用link和@import有什么区别? 11
35. 浏览器的内核分别是什么? 常见兼容性问题? 11

36. 介绍下CSS的盒子模型?	11
37. CSS3新增伪类举例	11
38. 使用CSS预处理器吗? 喜欢哪个	11
39. H5界面的响应式	11

## 第二部分：JS面试题 12

1. DOMContentLoaded 与 OnLoad事件的区别。	12
2. “==”和“===”的不同。	12
3. JavaScript的数据类型都有哪些?	12
4. var foo = “11” + 2 - “1”;	12
5. split() join()的区别在哪?	12
6. 数组方法pop() push() unshift() shift()的区别在哪?	12
7. JavaScript的事件流模型都有哪些?	12
8. 请按执行顺序写出几次alert()语句a的值:	12
9. 请按要求操作数组a:	13
10. 以下技术您用到过哪些?	13
11. var stringArray = [“Hello”, “World”],给出您能想到的最好的方法输出“Hello World”。	13
12. typeof输出的类型有哪些 A.number B.string C.null D.undefined	13
13. 以下关于Array数组对象的说法不正确的是 (c)	13
14. .javascrip的基本数据类型有哪些? (a,b,e,f,g)	13
15. 下面有关浏览器中使用js跨域获取数据的描述。说法错误的是? a	13
16. 请写出打印出来的值。	14
17. 请分别使用原生js, jquery获取标签信息。(<div id=“div1”>这是一个DIV标签</div>)	14
18. 求下面两种情况下, a和b的值是多少?	14
19. 将数组[“a”, “b”, “c”, “d”]转化为字符串“d#c#b#a”;	14
20. 对应写出下列输出值。	14
21. 获取event目标对象的方法(要求兼容性)。	14
22. 经常使用的页面开发工具以及测试工具。	15
23. .经常使用什么脚本库, 开发或使用过什么应用或组建。	15
24. 请写出代码执行结果	15
25. 请写一段json字符串,包含所有的数据类型.	15
26. 请按执行顺序写出几次alert()语句a的值:	15
27. 请按执行顺序写出几次alert()语句a的值:	15
28. 怎样添加,移除,移动,复制,创建和查找节点?	15
29. 用js写一个正则匹配标签中是否包含一个class?	15
30. 完善以下代码:	15
31. 写一个函数,将一串字符串中空格替换”.”,如replace(“铝 神 欢 迎 你”)=“铝.神.欢迎.你”	16

32. 对作用域上下文和this的理解,看下列代码:	17
33. HTTP状态码有哪些, 分别代表什么意思?	17
34. 举例2种强制类型转换和1种隐式类型转换?	17
35. split() 和join()的区别	17
36. 事件绑定和普通事件有什么区别	17
37. call和apply的区别	17
38. 事件委托是什么, 使用事件委托有什么好处?	17
39. 如何阻止事件冒泡和默认事件	17
40. 跨域的方法	17
41. 编写一个数组去重的方法	18
42. 如何实现多标签页之间的通讯。	18
43. 简述什么是按需加载,然后实现一个图片按需加载的js。	18
44. 简述下闭包,有什么好处和坏处。	18
45. 什么是模块开发,怎么实现一个模块开发。	18
46. 怎么优化一个首屏的加载速度,可以从js和css等多方便考虑	18
47. 怎么实现一个对象的继承。	18
48. 简述下平时开发用到的技术栈。	19
49. 在JavaScript中, 以下哪个变量名是非法的(b)	19
50. 请选择结果为真的表达式 (c)	19
51. var a = new Object();	19
52. var a = 3;	19
53. 微信分享怎么实现	20
54. 以下localStorage的使用方式,哪些是正确的?abe	20
55. 以下关于Array数组对象的说法不正确的是?c	20
56. var stringArray=["Hello","World"],给出您能想到的最好的方法输出"Hello, World"	20
57. HTML5存储类型有哪些?	20
58. 说几条写JavaScript的基本规范	20
59. 写一个通用的事件侦听器函数	21
60. Node js的适用场景?	21
61. 介绍js的基本数据类型	21
62. JavaScript如何实现继承?	21
63. 谈谈对JSON的了解	22
64. 什么是闭包, 闭包的作用?	22
65. 请描述一下cookies sessionStorage和localStorage的区别?	22
66. 如何提高网页运行性能	22
67. Flash, Ajax各自的优缺点, 在使用中如何取舍?	22

68. JavaScript中如何检测一个变量是一个String类型？请写出函数实现。	22
69. 请写出使用过的UI框架、CSS框架、js框架（jquery除外）	22
70. 作PC端项目时，需要考虑哪些兼容性的问题？	22
71. PC端项目涉及到跨域，那么什么叫跨域？为什么需要跨域？跨域的方法有几种，都是什么原理？	26
72. 后台语言会哪些，到什么程度	26
73. 对于面向对象的理解	27
74. 对于Angular.js与node.js的认识，在工作中用到过吗？应用在哪里	27
75. 在开发中遇到了哪些问题？，如何解决的？	27
76. 移动端的框架都用到过哪些，在作移动端时，都用到过哪些，如何选择的？是否查看过某个框架的源码	27
77. 微信端的支付与地图功能	27

### 第三部分 jQuery 框架 28

### 第四部分 AJAX 28

1. HTTP状态码有哪些，分别代表什么意思？	28
2. ajax请求的时候get和post方式的区别	28
3. Ajax请求Json数据后如何解析	28
4. ajax出问题时怎么判断是前端的问题还是后台的	29
5. ajax请求到显示流程	29
6. 向后台请求数据，有几种方式	29
7. 什么是AJAX？同步异步的区别？	29
8. Ajax的原理	29
9. 什么是HTTP协议，在打开网页时，HTTP协议作了哪些工作，有哪些步骤？	29

## 第一部分 HTML&CSS整理答案

### 1. 标签上title与alt属性的区别是什么?

答: title是设置鼠标移动到图片上时显示的内容, 而alt是用于当图片没有正常显示时出现的提示文字, 另外alt还用于在seo中针对图片的优化说明.

### 2. <p>标签内的文字是什么颜色的?

```
<style>
.classA{color:blue !important;}
.classB{color:red;}
</style>
<body>
  <p class='classB classA' style='color:green'>123</p>
</body>
```

答: 文字是blue蓝色, !important, 可以更改默认的CSS样式优先级规则, 使该条样式属性声明具有最高优先级

### 3. css单位px,em,rem的区别。

答: px像素 (Pixel)。相对长度单位。像素px是相对于显示器屏幕分辨率而言的

px的特点:

1. IE无法调整那些使用px作为单位的字体大小;
2. 国外的大部分网站能够调整的原因在于其使用了em或rem作为字体单位;
3. Firefox能够调整px和em, rem, 但是96%以上的中国网民使用IE浏览器(或内核)。

em是相对长度单位。相对于当前对象内文本的字体尺寸。如当前对行内文本的字体尺寸未被人为设置, 则相对于浏览器的默认字体尺寸

EM特点

1. em的值并不是固定的;
2. em会继承父级元素的字体大小。

rem是CSS3新增的一个相对单位 (root em, 根em), 这个单位引起了广泛关注。这个单位与em有什么区别呢? 区别在于使用rem为元素设定字体大小时, 仍然是相对大小, 但相对的只是HTML根元素。这个单位可谓集相对大小和绝对大小的优点于一身, 通过它既可以做到只修改根元素就成比例地调整所有字体大小, 又可以避免字体大小逐层复合的连锁反应。目前, 除了IE8及更早版本外, 所有浏览器均已支持rem。对于不支持它的浏览器, 应对方法也很简单, 就是多写一个绝对单位的声明。这些浏览器会忽略用rem设定的字体大小。

### 4. 以下哪些是css伪类? (c,d) after和before是伪元素

A.after B.before C.hover D.nth-child

### 5. 以下不是html5元素的有哪些? (b,c)

A.footer B.center C.font D.frame

### 6. 下述有关css属性position的属性值的描述。语法错误的是? b

- A. static: 没有定位, 元素出现在正常的流中。
- B. fixed: 生成绝对定位的元素, 相对于父元素进行定位。
- C. relative: 生成相对定位的元素, 相对于元素本身正常位置进行定位。
- D. absolute: 生成绝对定位的元素, 相对于static定位意外的第一个祖先元素。

### 7. CSS实现隐藏页面元素的方法有哪些?

display:none; visibility: hidden; opacity:0; position:absolute;left:-10000px;

### 8. xhtml和html有什么区别?

xhtml比HTML严谨性会高点，然后基本语言都还是沿用的HTML的标签，只不过废除了部分表现层的标签，同事在标准上要求高了点比如标签的严格嵌套，标签结束等等

## 9. 每个html文件里开头都有个Doctype。这是代表了什么意思？

版本声明

## 10. strong与em的异同？

语义都是语气加重，strong在显示上是变粗，而em是斜体显示

## 11. 行内元素有哪些？块级元素有哪些？

块状：\* <ol>

\* <ul>

\* <dl>

\* <table>

\* <h1>...<h6>

\* <p>

\* <blockquote>

\* <dt>

\* <address>

\* <caption>

\* <div>

内联（行内）：

\* a - 锚点

\* abbr - 缩写

\* acronym - 首字

\* b - 粗体(不推荐)

\* bdo - bidi override

\* big - 大字体

\* br - 换行

\* cite - 引用

\* code - 计算机代码(在引用源码的时候需要)

\* dfn - 定义字段

\* em - 强调

\* font - 字体设定(不推荐)

\* i - 斜体

\* img - 图片

\* input - 输入框

\* kbd - 定义键盘文本

\* label - 表格标签

\* q - 短引用

\* s - 中划线(不推荐)

\* samp - 定义范例计算机代码

\* select - 项目选择

\* small - 小字体文本

\* span - 常用内联容器，定义文本内区块

\* strike - 中划线

\* strong - 粗体强调

\* sub - 下标

\* sup - 上标

\* textarea - 多行文本输入框

\* tt - 电传文本

- \* u - 下划线
- \* var - 定义变量

## 12. css选择符有哪些？哪些属性可以继承？优先级算法如何计算？

选择符: [http://www.360doc.com/content/14/0104/22/15643\\_342679505.shtml](http://www.360doc.com/content/14/0104/22/15643_342679505.shtml)

除了这些都可以继承: display、margin、border、padding、background、height、min-height、max-height、width、min-width、max-width、overflow、position、left、right、top、bottom、z-index、float、clear、table-layout、vertical-align、page-break-after、page-break-before和unicode-bidi

原则一: 继承不如指定

原则二: #id > .class > 标签选择符

原则三: 越具体越强大

原则四: 标签#id > #id ; 标签.class > .class

CSS优先级包含四个级别(标签内选择符, ID选择符, Class选择符, 元素选择符)以及各级别出现的次数!

根据这四个级别出现的次数计算得到CSS的优先级。

CSS优先级的计算规则如下:

\* 元素标签中定义的样式 (Style属性), 加1,0,0,0

\* 每个ID选择符(如 #id), 加0,1,0,0

\* 每个Class选择符(如 .class)、每个属性选择符(如 [attribute=])、每个伪类(如 :hover)加0,0,1,0

\* 每个元素选择符(如p)或伪元素选择符(如 :firstchild)等, 加0,0,0,1

然后, 将这四个数字分别累加, 就得到每个CSS定义的优先级的值,

然后从左到右逐位比较大小, 数字大的CSS样式的优先级就高。

## 13. HTML5文档类型和字符集是?

```
<!DOCTYPE html>
```

```
<meta charset="UTF-8">
```

## 14. HTML5中canvas是什么? datalist是什么?

canvas: 绘图标签, 可以使用它制作游戏和图像处理

datalist:<datalist> 标签定义选项列表。与 input 元素配合使用该元素, 来定义 input 可能的值

## 15. 旧版本的IE浏览器(如:IE6,IE7)的默认盒模型,W3C定义的盒模型,二者之间有什么不同?

ie的width是包含border和padding的,也称为怪异模式, 触发条件是在没写版本声明的时候  
而w3c的width就width不含border和padding

## 16. 以下HTML代码片段,有几种方法让该div在浏览器中横向居中?

```
<div style="width:200px;height:100px;"></div>
```

第一种: <div style="width:200px;height:100px; margin:0 auto"></div>

第二种: <div style="width:200px;height:100px; position:absolute; left:50%; margin-left:-100px;"></div>

第三种:<div style="width:200px;height:100px; position:absolute; left:50%; transform:translate(-50%,0)"></div>

## 17. 以下css代码片段,在IE6,IE7,IE8,Firefox,Chrome下分别呈现什么颜色?

```
.browsers{
background-color:#000000;/*黑色*/
background-color:#ff0000\9;/*红色*/
+background-color:#00ff00;/*绿色*/
_background-color:#0000ff;/*蓝色*/
}
```

在firefox和chrome中显示的是黑色

在ie8下是红色(\9是ie8的hack)

在ie7下是绿色 (+是ie7的hack)

在ie6下是蓝色(\_是ie6转有hack)



18. 如何做到让一个div隐藏但是却在html流中占位?

visibility: hidden;和opacity:0;

19. 下面的输出是?

```
<style>
.red{color:red;}
.blue{color:blue;}
</style>
```

```
<p class="blue red"></p>
```

蓝色

20. 阅读以下代码回答:hello world 是什么颜色?

```
<style>
.classA{color:red;}
.classB{color:yellow;}
</style>
<p class="classB classA">hello world</p>
yellow
```

21. 请用CSS实现4个div的水平排列占满一行,要求4个div宽度相同,有10px的padding值.

```
div{
    float: left;
    width: 25%;
    height: 100px;
    background-color: #ccc;
    padding: 10px;
    box-sizing: border-box;
}
```

22. <!DOCTYPE html>中DOCTYPE有什么作用, 去掉会有什么影响?

声明文档类型, 在ie6下如果去掉会进入怪异模式, width包含padding和border

~~23. HTML行内元素有哪些, 块级元素有哪些? 画出CSS盒模型。~~

~~上面有~~

24. 内联和important哪个优先级更高?

important更高

25. 清除浮动的几种方式?

第一种: clear:both;

第二种: overflow:hidden;

第三种: 父级一起浮动

第四种: 父级定义:after伪元素配合zoom解决ie6,7

第五种: 父级加position:absolute

26. CSS3新增伪类选择器有哪些?

: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。

## 27. 用CSS3编写：div旋转90度，0.5秒后变圆，0.5秒后方法两倍。

略

## 28. 实现一个多列的等高布局。

略

## 29. css选择器中，以下排序正确的为 (a)

- A、行内样式>id>class>标签
- B、行内像是>class>id>标签
- C、id>class>行内样式>标签
- D、标签>class>id>行内样式

## 30. 如何解决在移动端1px的线比实际粗 <http://www.cnblogs.com/lunarorbitx/p/5287309.html>

1. 设置为0.5px
2. 判断如果在retina屏上就使用tranform对它进行缩放

@media screen and (-webkit-min-device-pixel-ratio: 2){

```
.radius-border:before{
  content: "";
  pointer-events: none; /* 防止点击触发 */
  box-sizing: border-box;
  position: absolute;
  width: 200%;
  height: 200%;
  left: 0;
  top: 0;
  border-radius: 8px;
  border: 1px solid #999;
  -webkit-transform(scale(0.5));
  -webkit-transform-origin: 0 0;
  transform(scale(0.5));
  transform-origin: 0 0;
}
```

## 31. 如何计算移动端rem的值

如果html的font-size为10px,2rem就是20px.

## 32. zoom是什么,有什么用?

Zoom属性是IE浏览器的专有属性,它可以设置或检索对象的缩放比例,比如触发ie的hasLayout属性,清除浮动、清除margin的重叠等。

## 33. HTML5 为什么只需要写<! DOCTYPE HTML>?

首先写法是<!DOCTYPE html>

它的用途是声明本页面是一个html5的标准文档，写了他就可以使用html5的新标签等，如果不写在ie6下会进入怪异模式。 [http://www.w3school.com.cn/tags/tag\\_doctype.asp](http://www.w3school.com.cn/tags/tag_doctype.asp)

相比于之前的比如xhtml的版本声明中有三种类型（过渡型、严格型、框架集），而html5这个版本语法更加宽松，不需要声明太多的类型，只需要说是html文档就行，所以版本声明是<!DOCTYPE html>

### 34. 页面导入样式时，使用link和@import有什么区别？

首先解释一下link和@import：

link是用于加载一个css文件，而@import是用于在css文件中加载其他的css文件，主要是用在大型项目中，可以通过@import来管理这些css文件。

主要区别就是：link是一个标签，写在html文档中的，而@import是写在css样式表中的

### 35. 浏览器的内核分别是什么？常见兼容性问题？

Trident内核代表产品Internet Explorer，又称其为IE内核。Trident（又称为MSHTML），是微软开发的一种排版引擎。使用Trident渲染引擎的浏览器包括：IE、傲游、世界之窗浏览器、Avant、腾讯TT、Netscape 8、NetCaptor、Sleipnir、GOSURF、GreenBrowser和KKman等

Gecko内核代表作品Mozilla FirefoxGecko是一套开放源代码的、以C++编写的网页排版引擎。Gecko是最流行的排版引擎之一，仅次于Trident。使用它的最著名浏览器有Firefox、Netscape6至9

WebKit内核代表作品Safari、Chromewebkit 是一个开源项目，包含了来自KDE项目和苹果公司的一些组件，主要用于Mac OS系统，它的特点在于源码结构清晰、渲染速度极快。缺点是对网页代码的兼容性不高，导致一些编写不标准的网页无法正常显示。主要代表作品有Safari和Google的浏览器Chrome，国内的360极速版，猎豹也是用webkit作为内核，移动端ios和android主流用的也都是webkit内核的

Presto内核代表作品OperaPresto是由Opera Software开发的浏览器排版引擎，供Opera 7.0及以上使用。它取代了旧版Opera 4至6版本使用的Elektra排版引擎，包括加入动态功能，例如网页或其部分可随着DOM及Script语法的事件而重新排版，现在这个内核已被opera放弃，opera现在的内核是webkit的

<http://blog.csdn.net/chuyuqing/article/details/37561313/>

### 36. 介绍下CSS的盒子模型？

盒子模型的组成：内容(content)、填充(padding)、边框(border)、边界(margin)。

这些属性我们可以把它转移到我们日常生活中的盒子（箱子）上来理解，日常生活中所见的盒子也就是能装东西的一种箱子，也具有这些属性，所以叫它盒子模式

### ~~37. CSS3新增伪类举例~~

: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。

### 38. 使用CSS预处理器吗？喜欢哪个

less,sass

### 39. H5界面的响应式

- a) 就是一个网站能够兼容多个终端——而不是为每个终端做一个特定的版本
- b) 主要使用css3中的media媒体查询，以及百分比布局

## 第二部分：JS面试题

### 1. DOMContentLoaded 与 OnLoad事件的区别。

onload会在html,css,js,图片等资源文件加载完成后执行里面的代码，而DOMContentLoaded则只需在html,css,js加载完成后就执行里面的代码。

### 2. “==”和“===”的不同。

==用于一般比较，===用于严格比较，==在比较的时候可以转换数据类型，===严格比较，只要类型不匹配就返回false

### 3. JavaScript的数据类型都有哪些？

JavaScript中有5种简单数据类型（也称为基本数据类型）：Undefined、Null、Boolean、Number和String。还有1种复杂数据类型——Object，Object本质上是由一组无序的名值对组成的

### 4. var foo = “11” + 2 - “1”;

foo 输出的是？

typeof foo输出的是？

foo输出的是111,首先字符串11+数字2,执行字符串连接为112,而后减1，是将112执行隐式类型转换为数字112后再减1，所以最后结果是111

typeof foo，为number

### 5. split() join()的区别在哪？

split是将字符串分割为数组，而join是将数组中的值连接为字符串

### 6. 数组方法pop() push() unshift() shift()的区别在哪？

pop是删除数组最后一位

push是在数组最后一位插入一个新元素

unshift是在数组开头添加一个或更多元素

shift是将数组开头的元素删除

### 7. JavaScript的事件流模型都有哪些？

在各种浏览器中存在三种事件模型:原始事件模型(original event model),DOM2事件模型,IE事件模型.

原始事件模型简单来说就是用on+事件名的方式绑定的事件,比如onclick,onsubmit等

dom2事件模型就是通过addEventListener绑定的事件，可以通过event.stopPropagation()来停止事件的传播,调用preventDefault()来阻止浏览器的默认行为

IE模型也提供了一个event对象封装了事件的详细信息,但是IE不将该对象传入事件处理函数,由于在任意时刻只会存在一个事件,所以IE把它作为全局对象window的一个属性,IE中的事件传播模式对应于DOM2的第二和第三阶段,首先执目标元素的处理函数,然后向上传播到达document,ie中只能捕捉鼠标事件,而DOM2中可以捕捉所有的事件,IE中注册和删除事件处理函数的方法也不同于DOM2.

事件处理函数的注册和删除是通过元素的attachEvent("eventType","handler") and detachEvent("eventType","handler"),与dom2不同的是eventType有on前缀

### 8. 请按执行顺序写出几次alert()语句a的值：

```
function fn1() {
    var a = 1;
    function fn2() {
        alert(a++);
    }
    return fn2;
}
var f = fn1();
f();
f();
```

两次：第1次1，第二次2，var f=fn1()的时候只是调用了fn1这个函数并将fn2作为返回值返回，并未执行fn2里面的内容。

在第一个f()的时候就是执行fn2里面的内容，同时因为fn2是一个闭包函数，调用了fn1中的a，所以a为1，至于a++，因为a++为后自加，先引用a,再进行自加操作，所以第一次弹出为1,而第二个f(),因为a的值已经++过了，所以弹出2。

## 9. 请按要求操作数组a:

i. 如何删除数组a中第2个元素（即：“B”）？

```
var a = ["A", "B", "C", "D"];
a.splice(1,1);
```

ii. 如何删除数组a中最后一个元素？

```
var a = ["A", "B", "C", "D"];
a.pop()
```

iii. 如何将“NEW”插入数组a的“B”和“C”之间？

```
var a = ["A", "B", "C", "D"];
a.splice(2,0,"new");
```

## 10. 以下技术您用到过哪些？

EasyUI Bootstrap jQuery jQuery Mobile zepto ExtJS  
angular react backbone redux grunt gulp seats nodeJS  
swiper web pack requireJS

略

## 11. var stringArray = ["Hello", "World"],给出您能想到的最好的方法输出“Hello World”。

```
stringArray.join(" ");
```

## 12. typeof输出的类型有哪些 A.number B.string C.null D.undefined

## 13. 以下关于Array数组对象的说法不正确的是 (c)

A.对数组里数据的排序可以用sort函数，如果排序效果非预期，可以给sort函数加一个排序函数的参数。

B.reverse用于对数组数据的倒序排列。

C.向数组的最后位置加一个新元素，可以用pop方法。

D.unshift方法用于向数组删除第一个元素。

## 14. .javascript的基本数据类型有哪些？ (a,b,e,f,g)

A.number B.string C.object D.array E.null F.undefined G.boolean

## 15. 下面有关浏览器中使用js跨域获取数据的描述。说法错误的是？ a

A.域名，端口相同，协议不同，属于相同的域。

B.js可以使用jsonp进行跨域。

C.通过修改document.domain来跨子域。

D.使用window.name来进行跨域。

16. 请写出打印出来的值。

```
var flag = true;
if (flag) {
    var a = 10;
}
function fn() {
    var b = 20;
    c = 30;
}
fn();
console.log(a);
console.log(c);
console.log(b);
```

答:10,30,报错(b是在fn内部定义，全局并没有b这个变量)

17. 请分别使用原生js，jquery获取标签信息。（<div id="div1">这是一个DIV标签</div>）

```
document.getElementById("div1").innerHTML;
document.querySelector("#div1").innerText;
$("#div1").html()
```

18. 求下面两种情况下，a和b的值是多少？

题目1: var a = 1; function func1(){a=b=5;} alert(a); alert(b);

题目2: var a = 1; function func1(){var b = 1;a=b=5;}alert(a);alert(b);

题目1答：如果在alert(a)前没有调用func1的话就1，和报错，b没定义,如果调用了func1的话就是5,5

题目2答：如果在alert(a)前没有调用func1的话就1，和报错，b没定义,如果调用了func1的话就是5,和报错，因为b是在func1中定义的局部变量，在全局获取不到

19. 将数组["a","b","c","d"]转化为字符串"d#c#b#a";

```
['a','b','c','d'].join("#")
```

20. 对应写出下列输出值。

```
alert(typeof(null))
alert(typeof(NaN))
alert(typeof(undefined))
alert(typeof("undefined"))
alert(NaN == undefined)
alert(NaN == NaN)
```

答:

```
alert(typeof(null))//object
alert(typeof(NaN))//number
alert(typeof(undefined))//undefined
alert(typeof("undefined"))//string
alert(NaN == undefined)//false
alert(NaN == NaN)//false
```

21. 获取event目标对象的方法（要求兼容性）。

```
document.onclick = function (ev){
    var e = ev || window.event;
}
```

22. 经常使用的页面开发工具以及测试工具。

sublime, hbuilder, webstorm, 测试工具: mocha

23. 经常使用什么脚本库, 开发或使用过什么应用或组建。

jquery, zepto

24. 请写出代码执行结果

```
var a;  
alert(a);           ____undefined_____  
alert(typeof a);    undefined_____  
alert(b);           报错_____
```

25. 请写一段json字符串, 包含所有的数据类型.

```
{"a": "string", "b": 5, "c": [1, 2, 3, 4], "d": {"e": "obj"}}
```

26. 请按执行顺序写出几次alert()语句a的值:

```
var a = 10;  
function fn() {  
  alert(a);  
  var a = 100;  
  alert(a);  
}  
fn();  
alert(a);
```

undefined, 100, 10

27. 请按执行顺序写出几次alert()语句a的值:

```
function fn1() {  
  
  var a = 1;  
  
  function fn2() {  
    alert(a++);  
  }  
  
  fn2();  
}  
fn1();  
fn1();
```

fn2();  
}

fn1();  
fn1();

1, 1

28. 怎样添加, 移除, 移动, 复制, 创建和查找节点?

appendChild, removeChild, replaceChild, cloneNode, createElement, getElementById, getElementsByTagName

~~29. 用js写一个正则匹配标签中是否包含一个class?~~

```
var className = "active";  
var reg = new RegExp("\\b" + className + "\\b");
```

30. 完善以下代码:

```
<div style="height: 50px;" id="content">  
<p>铅神公司</p>  
</div>
```

(1). 请用CSS实现"铅神公司"在div内水平垂直居中?

(2).请用javascript将字体设置为红色.

答: (1)

第一种写法:

```
*{
    margin: 0;
}
div{
    text-align: center;
    line-height: 50px;
}
```

第二种写法:

```
*{
    margin: 0;
}
div{
    display: table;
    width: 100%;
}
div p{
    display: table-cell;
    width: 100%;
    text-align: center;
    vertical-align: middle;
}
```

第三种写法:

```
*{
    margin: 0;
}
div{
    position: relative;
}
div p{
    position: absolute;
    left: 50%;
    top: 50%;
    transform: translate(-50%,-50%);
}
```

(2)

**第一种:** content.style.color="red", 高级浏览器id不用获取

**第二种:** document.getElementById("content").style.color="red"

**第三种:** document.querySelector("#content").style.color="red";

31. 写一个函数,将一串字符串中空格替换".",如replace("铝 神 欢 迎 你")="铝.神.欢.迎.你"

第一种:

```
function re(s){
    return s.replace(/\s/g,".");
}
```

第二种:

```
function re(s){
```



```

        return s.split(" ").join(".");
    }
}

```

### 32. 对作用域上下文和this的理解,看下列代码:

```

var User = {

    count:1,

    getCount:function(){

        return this.count;

    }

};

console.log(User.getCount());//??

var func = User.getCount;

console.log(func());//??

```

1,undefined

### 33. HTTP状态码有哪些，分别代表什么意思？

如果是1开头的代表请求已被接受，需要继续处理

如果是2开头这一类型的状态码，代表请求已成功被服务器接收、理解、并接受

如果是3开头这类状态码代表需要客户端采取进一步的操作才能完成请求

如果是4开头这类的状态码代表了客户端看起来可能发生了错误，妨碍了服务器的处理

如果是5开头这类状态码代表了服务器在处理请求的过程中有错误或者异常状态发生，也有可能是服务器意识到以当前的软硬件资源无法完成对请求的处理

### 34. 举例2种强制类型转换和1种隐式类型转换？

强制 (parseInt,parseFloat,number)

隐式 (== - ===)

### 35. split() 和join()的区别

split是将字符串切割为数组，而join是将数组连接为字符串

### 36. 事件绑定和普通事件有什么区别

区别就是普通事件只支持单个事件，而事件绑定可以添加多个事件

### 37. call和apply的区别

call和apply都是改变函数内部的this指向，第一个参数也都放的是要改变this对象，区别主要在第二个参数上，call从第二个开始放的是正常的参数，而apply放的是数组

### 38. 事件委托是什么，使用事件委托有什么好处？

就是让别人来做，这个事件本来是加在某些元素上的，然而你却加到别人身上来做，完成这个事件。

也就是：利用冒泡的原理，把事件加到父级上，触发执行效果。 好处呢：1，提高性能。

### 39. 如何阻止事件冒泡和默认事件

阻止事件冒泡：ie:window.event.cancelBubble=true;而非ie:e.stopPropagation();

阻止默认事件：ie:window.event.returnValue = false;return false;而非ie:e.preventDefault();

### 40. 跨域的方法

1> document.domain + iframe （只有在主域相同的时候才能使用该方法）

2> window.name + iframe

3>html5中的postMessage()  
4>CORS  
5>jsonp  
6>websocket

跨域支持post请求的:

1>CORS  
2>window.postMessage  
3>html5的websocket  
4>iframe <http://www.jianshu.com/p/4773501f1adf>  
5>flash proxy

#### 41. 编写一个数组去重的方法

```
Array.prototype.unique3 = function(){  
    var res = [];  
    var json = {};  
    for(var i = 0; i < this.length; i++){  
        if(!json[this[i]]){  
            res.push(this[i]);  
            json[this[i]] = 1;  
        }  
    }  
    return res;  
}
```

#### 42. 如何实现多标签页之间的通讯。

调用 localStorage、cookies 等本地存储方式

#### 43. 简述什么是按需加载,然后实现一个图片按需加载的js。

图片的按需加载又称为延时加载,随着用户的滚动条滑动来按需获取图片。代码略,课件中有

#### 44. 简述下闭包,有什么好处和坏处。

闭包的好处有:

- 1.缓存
- 2.面向对象中的对象
- 3.实现封装,防止变量跑到外层作用域中,发生命名冲突
- 4.匿名自执行函数,匿名自执行函数可以减小内存消耗

闭包的坏处:

- 1.内存消耗
- 2.性能差

#### 45. 什么是模块开发,怎么实现一个模块开发。

一个模块就是实现特定功能的文件,有了模块,我们就可以更方便地使用别人的代码,想要什么功能,就加载什么模块。

可以使用模块加载器比如sea.js或者require.js

#### 46. 怎么优化一个首屏的加载速度,可以从js和css等多方便考虑

可以

- 1.可以使用延迟加载,通过js将第二屏出现的图片设置成延迟加载,优先加载第一屏的图片,如果不想自己写js推荐一个lazyload这个插件
- 2.尽可能压缩图片大小,可以利用在线图片压缩工具tinypng
- 3.将JS放到HTML最后的地方,能最后加载,优先保证基础信息阅读。可以参考雅虎优化原则

#### 47. 怎么实现一个对象的继承。

```
Object.prototype.extend = function(obj){
```

```
//在函数里，把obj属性复制到自身
for(var k in obj){
  if(obj.hasOwnProperty(k)){
    if(this[k] == undefined){
      this[k] = obj[k];
    }
  }
}
}

var kitty = {color:'yellow',climb:function(){alert('我会爬树')}};
var tiger = {color:'yellow and black'};
tiger.extend(kitty);
tiger.climb();
```

#### 48. 简述下平时开发用到的技术栈。

作者：何幻

链接：<http://www.zhihu.com/question/33179506/answer/75015013>

来源：知乎

著作权归作者所有，转载请联系作者获得授权。

浏览器环境IE6~8 IE9+ Chrome Firefox Safari Opera Edge  
 HTTP协议URI Cache Session Cookie Request Response  
 HTMLW3C HTML4.01 HTML5 DOM 语义化  
 CSSCSS1~3 选择器 盒模型 Flex 媒体检测 响应式 渲染引擎  
 JavaScriptEcmaScript3~5~6 Lexical-scope prototype-chain AJAX js引擎  
 编辑器Emacs Vim WebStorm Atom Sublime-Text  
 发布部署合并 压缩 单元测试 Node.js Grunt Gulp Yeoman Phantom JSLint  
 框架类库jQuery Zepto UnderScore Backbone Knockout React AngularJS  
 模块管理CommonJS Webpack  
 UI框架BootStrap SemanticUI jQueryUI Foundation  
 推送技术WebSocket 轮询 长连接  
 跨域技术iframe JSONP CORS  
 数据可视化D3 Echarts HighCharts Canvas  
 异步编程Promise \$.Deferred Generator  
 CSS预处理器LESS SASS Stylus  
 客户端模板Handlebars Haml Jade Mustache  
 相关语言CoffeeScript TypeScript Dart WebAssembly  
 WebApp/PCReact-native ionic

#### 49. 在JavaScript中，以下哪个变量名是非法的(b)

A、Name B、9name C、Name\_a D、Name9

#### 50. 请选择结果为真的表达式 (c)

A、null instanceof Object B、null === undefined C、null == undefined D、NaN == NaN

#### 51. var a = new Object();

```
a.value = 1;
b = a;
b.value = 2;
console.log(a.value);
输出正确的是： (b)
```

A、1 B、2 C、undefined D、以上都不正确

#### 52. var a = 3;

```
(function() {  
window.a = 2  
var a = 1;  
})()
```

console.log(a)  
输出为: (b)

A、1                  B、2                  C、3                  D、undefined

## 53. 微信分享怎么实现

分为3种:

第一种使用微信默认分享, 分享标题为分享页的title标签的内容, 如果是分享给朋友, 描述中为网页链接, 分享小图标为网页中第一张符合大小的图片

第二种是使用微信的jssdk, wx.onMenuShareTimeline分享到朋友圈和wx.onMenuShareAppMessage分享给朋友。

第三种如果是做微信广告, 部署到腾讯服务器下, 可以使用WeixinJSBridge调用分享接口

## 54. 以下localStorage的使用方式,哪些是正确的?abe

- A.localStorage.setItem("lastname", "smith");
- B.document.getElementById("result").innerHTML=localStorage.getItem("lastname");
- C.localStorage.lastname="Smith";
- D.document.getElementById("result").innerHTML=localStorage.lastname;
- E.localStorage.removeItem("lastname");

## 55. 以下关于Array数组对象的说法不正确的是?c

- A.对数组里数据的排序可以用sort函数,如果排序效果非预期,可以给sort函数加一个排序函数的参数.
- B.reverse用于对数组数据的排序排列.
- C.向数组的最后位置加一个新元素,可以用pop方法.
- D.unshift方法用于向数组删除第一个元素.

## 56. var stringArray=["Hello","World"],给出您能想到的最好的方法输出"Hello, World" stringArray.join(",");

## 57. HTML5存储类型有哪些?

cookies: 存储大小为4k

localStorage: 以键值对(Key-Value)的方式存储, 永久存储, 永不失效, 除非手动删除,大小为5m

sessionStorage:HTML5 的本地存储 API 中的 localStorage 与 sessionStorage 在使用方法是相同的, 区别在于 sessionStorage 在关闭页面后即被清空, 而 localStorage 则会一直保存

离线缓存(application cache):本地缓存应用所需的文件

Web SQL:关系数据库, 通过SQL语句访问

Web SQL 数据库 API 并不是 HTML5 规范的一部分, 但是它是一个独立的规范, 引入了一组使用 SQL 操作客户端数据库的 APIs。

## 58. 说几条写JavaScript的基本规范

JavaScript代码不应该被包含在HTML文件中,除非这是段特定只属于此部分的代码。在HTML中的JavaScript代码会明显增加文件大小,而且也不能对其进行缓存和压缩

<script src=filename.js>应尽量放到body的后面。这样可以减少因为载入script而造成其他页面内容载入也被延迟的问题。也没有必要使用language或者type属性。MIME类型是由服务器而非scripttag来决定的

缩进的单位为四个空格, 保证使用不同编辑器打开都是统一的

避免每行超过80个字符。当一条语句一行写不下时,请考虑折行。在运算符,最好是逗号后换行。在运算符后换行可以减少因为复制粘贴产生的错误被分号掩盖的几率。下一行应该缩进8个空格

所有的变量必须在使用前进行声明。JavaScript并不强制必须这么做,但是这么做可以让程序易于阅读,且也容易发现那些没声明的变量(它们会被编译成全局变量)。

将var语句放在函数的首部。

最好把每个变量的声明语句单独放到一行,并加上注释说明。

变量名应由26个大小写字母(A..Z,a..z),10个数字(0..9),和\_(下划线)组成。避免使用国际化字符(如中文),因为它们不是在任何地方都可以被方便的阅读和理解。不要在命名中使用\$(美元符号)或者(反斜杠)。

不要把\_(下划线)作为变量名的第一个字符。它有时用来表示私有变量,但实际上JavaScript并没提供私有变量的功能。如果私有变量很重要,那么使用私有成员的形式。应避免使用这种容易让人误解的命名习惯。

大多数的变量名和方法名应以小写字母开头。

必须与new共同使用的构造函数名应以大写字母开头。当new被省略时JavaScript不会有任何编译错误或运行错误抛出。忘记加new时会让不好的事情发生(比如被当成一般的函数),所以大写构造函数名是我们来尽量避免这种情况发生的唯一办法。

<http://www.cnblogs.com/niuniu/archive/2010/08/26/1808780.html>

## 59. 写一个通用的事件侦听器函数

```
function addEvent(obj,event,fn){
    //现代: addEventListener 不用加on
    //ie: touchEvent 需要加on
    if (window.addEventListener){
        obj.addEventListener(event,fn);
    }else{
        obj.attachEvent("on"+event, fn);
    }
}
```

## 60. Node js的适用场景?

### 1. RESTful API

这是NodeJS最理想的应用场景,可以处理数万条连接,本身没有太多的逻辑,只需要请求API,组织数据进行返回即可。它本质上只是从某个数据库中查找一些值并将它们组成一个响应。由于响应是少量文本,入站请求也是少量的文本,因此流量不高,一台机器甚至也可以处理最繁忙的公司的API需求。

### 2. 统一Web应用的UI层

目前MVC的架构,在某种意义上来说,Web开发有两个UI层,一个是在浏览器里面我们最终看到的,另一个在server端,负责生成和拼接页面。

不讨论这种架构是好是坏,但是有另外一种实践,面向服务的架构,更好的做前后端的依赖分离。如果所有的关键业务逻辑都封装成REST调用,就意味着在上层只需要考虑如何用这些REST接口构建具体的应用。那些后端程序员们根本不操心具体数据是如何从一个页面传递到另一个页面的,他们也不用管用户数据更新是通过Ajax异步获取的还是通过刷新页面。

### 3. 大量Ajax请求的应用

例如个性化应用,每个用户看到的页面都不一样,缓存失效,需要在页面加载的时候发起Ajax请求,NodeJS能响应大量的并发请求。总而言之,NodeJS适合运用在高并发、I/O密集、少量业务逻辑的场景。

## 61. 介绍js的基本数据类型

字符串、数字、布尔、数组、对象、Null、Undefined

## 62. JavaScript如何实现继承?

如果继承父类中的属性和方法使用call或者apply,如果是要继承父类原形的内容,直接将子类的prototype指向父类的prototype

示例:

```
//职工类
function 职员(姓名,性别,工作){
```

```

        //call,apply
        // Person.call(this,姓名,性别);
        Person.apply(this,arguments);

        this.job=工作;
        // this.性别=""
        this.name="润田";
    }

```

职员.prototype=Person.prototype;

## 63. 谈谈对JSON的了解

JSON的全称是"JavaScript Object Notation"，意思是JavaScript对象表示法，它是一种基于文本，独立于语言的轻量级数据交换格式

## 64. 什么是闭包，闭包的作用？

闭包就是一个函数引用另外一个函数的变量，因为变量被引用着所以不会被回收，因此可以用来封装一个私有变量

## 65. 请描述一下cookies sessionStorage和localStorage的区别？

1.与cookie不同的是：Web Storage数据完全存储在客户端，不需要通过浏览器的请求将数据传给服务器，因此x相比cookie来说能够存储更多的数据，大概5M左右。

2.LocalStorage和sessionStorage功能上是一样的，但是存储持久时间不一样。

LocalStorage：浏览器关闭了数据仍然可以保存下来，并可用于所有同源（相同的域名、协议和端口）窗口（或标签页），

sessionStorage：数据存储在窗口对象中，窗口关闭后对应的窗口对象消失，存储的数据也会丢失。

注意：sessionStorage 都可以用localStorage 来代替，但需要记住的是，在窗口或者标签页关闭时，使用sessionStorage 存储的数据会丢失。

3、使用 local storage和session storage主要通过在js中操作这两个对象来实现，分别为window.localStorage和window.sessionStorage. 这两个对象均是Storage类的两个实例，自然也具有Storage类的属性和方法。

## 66. 如何提高网页运行性能

<http://blog.csdn.net/wojiaopanpan/article/details/7616687>

## 67. Flash，Ajax各自的优缺点，在使用中如何取舍？

<http://blog.csdn.net/duandianyiwai/article/details/52127839>

## 68. JavaScript中如何检测一个变量是一个String类型？请写出函数实现。

```

function isString (s) {

    return typeof s === "string";
}

```

## 69. 请写出使用过的UI框架、CSS框架、js框架（jquery除外）

略

## 70. 作PC端项目时，需要考虑哪些兼容性的问题？

浏览器兼容问题一：不同浏览器的标签默认的外补丁和内补丁不同

问题症状：随便写几个标签，不加样式控制的情况下，各自的margin 和padding差异较大。

碰到频率:100%

解决方案：CSS里 `*{margin:0;padding:0;}`

备注：这个是最常见的也是最易解决的一个浏览器兼容性问题，几乎所有的CSS文件开头都会用通配符\*来设置各个标签的内外补丁是0。

浏览器兼容问题二：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

问题症状：常见症状是IE6中后面的一块被顶到下一行

碰到频率：90%（稍微复杂点的页面都会碰到，float布局最常见的浏览器兼容问题）

解决方案：在float的标签样式控制中加入 `display:inline`；将其转化为行内属性

备注：我们最常用的就是div+CSS布局了，而div就是一个典型的块属性标签，横向布局的时候我们通常都是用div float实现的，横向的间距设置如果用margin实现，这就是一个必然会碰到的兼容性问题。

浏览器兼容问题三：设置较小高度标签（一般小于10px），在IE6，IE7，遨游中高度超出自己设置高度

问题症状：IE6、7和遨游里这个标签的高度不受控制，超出自己设置的高度

碰到频率：60%

解决方案：给超出高度的标签设置`overflow:hidden`；或者设置行高`line-height` 小于你设置的高度。

备注：这种情况一般出现在我们设置小圆角背景的标签里。出现这个问题的原因是IE8之前的浏览器都会给标签一个最小默认的行高的高度。即使你的标签是空的，这个标签的高度还是会达到默认的行高。

浏览器兼容问题四：行内属性标签，设置`display:block`后采用float布局，又有横行的margin的情况，IE6间距bug

问题症状：IE6里的间距比超过设置的间距

碰到几率：20%

解决方案：在`display:block`；后面加入`display:inline`；`display:table`；

备注：行内属性标签，为了设置宽高，我们需要设置`display:block`；（除了input标签比较特殊）。在用float布局并有横向的margin后，在IE6下，他就具有了块属性float后的横向margin的bug。不过因为它本身就是行内属性标签，所以我们再加上`display:inline`的话，它的高宽就不可设了。这时候我们还需要在`display:inline`后面加入`display:table`。

浏览器兼容问题五：图片默认有间距

问题症状：几个img标签放在一起的时候，有些浏览器会有默认的间距，加了问题一中提到的通配符也不起作用。

碰到几率：20%

解决方案：使用float属性为img布局

备注：因为img标签是行内属性标签，所以只要不超出容器宽度，img标签都会排在一行里，但是部分浏览器的img标签之间会有个间距。去掉这个间距使用float是正道。（我的一个学生使用负margin，虽然能解决，但负margin本身就是容易引起浏览器兼容问题的用法，所以我禁止他们使用）

浏览器兼容问题六：标签最低高度设置min-height不兼容

问题症状：因为min-height本身就是一个不兼容的CSS属性，所以设置min-height时不能很好的被各个浏览器兼容

碰到几率：5%

解决方案：如果我们要设置一个标签的最小高度200px，需要进行的设置为：{min-height:200px; height:auto !important; height:200px; overflow:visible;}

备注：在B/S系统前端开发时，有很多情况下我们又这种需求。当内容小于一个值（如300px）时。容器的高度为300px；当内容高度大于这个值时，容器高度被撑高，而不是出现滚动条。这时候我们会面临这个兼容性问题。

浏览器兼容问题七：透明度的兼容CSS设置

做兼容页面的方法是：每写一小段代码（布局中的一行或者一块）我们都要在不同的浏览器中看是否兼容，当然熟练到一定的程度就没这么麻烦了。建议经常会碰到兼容性问题的新手使用。很多兼容性问题都是因为浏览器对标签的默认属性解析不同造成的，只要我们稍加设置都能轻松地解决这些兼容问题。如果我们熟悉标签的默认属性的话，就能很好的理解为什么会出现兼容问题以及怎么去解决这些兼容问题。

/\* CSS hack\*/

我很少使用hacker的，可能是个人习惯吧，我不喜欢写的代码IE不兼容，然后用hack来解决。不过hacker还是非常好用的。使用hacker我可以把浏览器分为3类：IE6；IE7和遨游；其他（IE8 chrome ff safari opera等）

◆IE6认识的hacker 是下划线\_和星号 \*

◆IE7 遨游认识的hacker是星号 \*

比如这样一个CSS设置：

height:300px;\*height:200px;\_height:100px;

IE6浏览器在读到height:300px的时候会认为高时300px；继续往下读，他也认识\*height，所以当IE6读到\*height:200px的时候会覆盖掉前一条的相冲突设置，认为高度是200px。继续往下读，IE6还认识\_height,所以他又会覆盖掉200px高的设置，把高度设置为100px；

IE7和遨游也是一样的从高度300px的设置往下读。当它们读到\*height200px的时候就停下了，因为它们不认识\_height。所以它们会把高度解析为200px，剩下的浏览器只认识第一个height:300px;所以它们会把高度解析为300px。因为优先级相同且想冲突的属性设置后一个会覆盖掉前一个，所以书写的次序是很重要的。

Pc端的项目兼容问题主要集中在ie6和ie7，而他们的市场份额已经非常低了，而且选择兼容他们会让我们代码变多，维护成本升高，不能使用css2.1或者3的高级命令等等。所以最好不要兼容ie6,7.

而如果要兼容ie8，那么在使用css3的时候一定要小心，因为ie8也是不兼容css3和html5的，所以建议在使用css3的时候要避免使用影响布局的css3属性.

在js中的兼容问题：

1)通过class获取标签：

```
function byClass(parents,className){
```

```
    if (parents.getElementsByTagName){
```



```

        return parents.getElementsByClassName(className);
    }

    var el=parents.getElementsByTagName("*");

    var arr=[];

    var reg=new RegExp("\\b"+className+"\\b");

    // console.log(reg);
    var arr=[];
    for (var i=0; i<el.length; i++){

        if (reg.test(el[i].className)){
            arr.push(el[i]);
        }
    }

    return arr;
}
2)ready函数
function ready(fn){
    if(document.addEventListener)
    {
        document.addEventListener('DOMContentLoaded', fn, false);
    }
    else
    {
        var oS=document.createElement('script');

        oS.defer=true;
        oS.onreadystatechange=function ()
        {
            if(oS.readyState=='complete')
            {
                fn();
            }
        };

        var oHead=document.getElementsByTagName('head')[0];

        oHead.appendChild(oS);
    }
}

```

### 3) 事件绑定

```

function addEvent(obj,event,fn){
    //现代: addEventListener 不用加on
    //ie: touchEvent 需要加on
    if (window.addEventListener){
        obj.addEventListener(event,fn);
    }else{
        obj.attachEvent("on"+event, fn);
    }
}

```

```

    }
}
4) 获取样式
function getStyle(el,styleName){

    if (el.currentStyle){
        return el.currentStyle[styleName];
    }else{
        return getComputedStyle(el, null)[styleName];
    }
}

```

```

5) ajax
function ajax(url,succFn,faultFn){

    if (window.XMLHttpRequest){
        var ajax=new XMLHttpRequest();
    }else{
        var ajax=new ActiveXObject("Msxml2.XMLHTTP");
    }

    ajax.open("get",url,true);

    ajax.send();

    ajax.onreadystatechange=function (){
        if (ajax.readyState=="4"){

            if (ajax.status>=200&&ajax.status<300 || ajax.status==304){

                if (succFn){
                    succFn(ajax.responseText);
                }
            }else{
                if (faultFn){
                    faultFn(ajax.status);
                }
            }
        }
    }
}

```

71. PC端项目涉及到跨域，那么什么叫跨域？为什么需要跨域？跨域的方法有几种，都是什么原理？

- 跨域简单的理解即为当前服务器目录下的前端需求要请求非本服务器(本域名)下的后台服务就称为跨域请求。跨域请求我们使用普通的ajax请求是无法进行的
- 由于同源策略，一般来说位于 server1.example.com 的网页无法与不是 server2.example.com的服务器沟通，或者说如果在server1.example.com下想获取server2.example.com的话就得用跨域请求
- 跨域的解决方法及解决：<http://blog.csdn.net/joyhen/article/details/21631833>

72. 后台语言会哪些，到什么程度

- node.php
- 常见的后台操作，数据库(mongodb,mysql)的基础操作，接口的编写，简易网站后台的编写

### 73. 对于面向对象的理解

- f) Javascript是一种基于对象的语言，在程序中遇到的所有东西几乎都是对象。但是，它又不是一种真正的面向对象编程（OOP）语言，因为它的语法中没有class类
- g) 在js中使用面向对象可以通过构造函数，所谓"构造函数"，其实就是一个普通函数，但是内部使用了this变量。对构造函数使用new运算符，就能生成实例，并且this变量会绑定在实例对象上
- h) 使编程的注意力重新从应用程序的逻辑回到其数据上来。换句话说，OOP将焦点从过程式编程转向最终建模的真实实体。这使得应用程序更接近我们周围的现实世界。
- i) OOP达到了软件工程的三个目标：重用性、灵活性和扩展性。采用面向对象方法可以使系统各部分各司其职、各尽所能；使其编程的代码更简洁、更易于维护，并且具有更强的可重用性

### 74. 对于Angular.js与node.js的认识，在工作中用过吗？应用在哪里

- j) CRUD的单页应用
- k) 后台，企业应用网站
- l) 交互密集型的应用

### 75. 在开发中遇到了哪些问题？，如何解决的？

略

### 76. 移动端的框架都用到过哪些，在作移动端时，都用到过哪些，如何选择的？是否查看过某个框架的源码

- c) bootstrap->响应式样式框架
- d) mui->移动端app开发框架
- e) ionic->使用JavaScript MVVM 框架和 AngularJS 来增强应用

### 77. 微信端的支付与地图功能

- f) 要使用微信支付功能需要公众号类型为服务号，并获得微信支付权限api
  - i. 需要设置测试的目录，在微信支付-» 开发配置下，添加白名单
  - ii. 设置正式支付目录
  - iii. 在微信浏览器里面打开H5网页中执行JS调起支付。接口输入输出数据格式为JSON
  - iv. 实例代码：

```
function onBridgeReady(){
    WeixinJSBridge.invoke(
        'getBrandWCPayRequest', {
            "appId" : "wx2421b1c4370ec43b", //公众号名称，由商户传入
            "timeStamp": "1395712654", //时间戳，自1970年以来的秒数
            "nonceStr" : "e61463f8efa94090b1f366cccfb444", //随机串
            "package" : "prepay_id=u802345jgfjsdfgsdg888",
            "signType" : "MD5", //微信签名方式
            "paySign" : "70EA570631E4BB79628FBCA90534C63FF7FADD89" //微信签名
        },
        function(res){
            if(res.err_msg == "get_brand_wcpay_request: ok") {} // 使用以上方式判断前端返回,微信团队郑重提示: res.err_msg将在用户支付成功后返回 ok，但并不保证它绝对可靠。
        }
    );
}

if (typeof WeixinJSBridge == "undefined"){
    if( document.addEventListener ){
        document.addEventListener('WeixinJSBridgeReady', onBridgeReady, false);
    }else if (document.attachEvent){
        document.attachEvent('WeixinJSBridgeReady', onBridgeReady);
        document.attachEvent('onWeixinJSBridgeReady', onBridgeReady);
    }
}
```

```

    }
  }else{
    onBridgeReady();
  }
}

```

#### g) 地图功能

使用微信内置地图查看位置接口

```

wx.openLocation({
  latitude: 0, // 纬度，浮点数，范围为90 ~ -90
  longitude: 0, // 经度，浮点数，范围为180 ~ -180。
  name: "", // 位置名
  address: "", // 地址详情说明
  scale: 1, // 地图缩放级别,整形值,范围从1~28。默认为最大
  infoUrl: "" // 在查看位置界面底部显示的超链接,可点击跳转
});

```

## 第三部分 jQuery 框架

## 第四部分 AJAX

### ~~1. HTTP状态码有哪些，分别代表什么意思？~~

如果是1开头的代表请求已被接受，需要继续处理

如果是2开头这一类型的状态码，代表请求已成功被服务器接收、理解、并接受

如果是3开头这类状态码代表需要客户端采取进一步的操作才能完成请求

如果是4开头这类的状态码代表了客户端看起来可能发生了错误，妨碍了服务器的处理

如果是5开头这类状态码代表了服务器在处理请求的过程中有错误或者异常状态发生，也有可能是服务器意识到以当前的软硬件资源无法完成对请求的处理

2.

### 3. ajax请求的时候get和post方式的区别

第一：ajax.open方法的第二个参数如果是get请求路径中是带有参数的，而post就只是路径

第二：如果是post方式在open方法和send方法之间还要设置请求的头：ajax.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");

第三：ajax.send方法如果是get的话参数留空就行，而post需要将参数以序列化的格式传入

### 4. Ajax请求Json数据后如何解析

~~第一种：使用eval~~

第二种：使用JSON.parse()

第三种: `new Function("return" + strJSON)();`

## 5. ajax出问题怎么判断是前端的问题还是后台的

首先测试api是否能正常调用,比如可以模拟一些数据发送一下,如果是get请求就直接浏览器地址栏中拼接上然后看返回结果。在确认api调用正常的情况下,基本就可以确定是属于前端的问题。

## 6. ajax请求到显示流程

- 1.创建ajax对象
- 2.调用open方法
- 3.调用send方法
- 4.添加onreadystatechange事件
- 5.判断返回的状态码是否为成功的状态码
- 6.通过调用ajax的responseText属性返回数据

## 7. 向后台请求数据,有几种方式

用的最多的就是get和post.

除了这两个还有:

HEAD

类似于get请求,只不过返回的响应中没有具体的内容,用于获取报头

PUT

上传当前请求文档的一个版本

DELETE

发送一个用来删除指定文档的请求

TRACE

发送请求的一个副本,以跟踪其处理进程

OPTIONS

返回所有可用的方法;可检查服务器支持哪些方法

CONNECT

用于ssl隧道的基于代理的请求

## 8. 什么是AJAX? 同步异步的区别?

通过在后台与服务器进行少量数据交换, AJAX 可以使网页实现异步更新。这意味着可以在不重新加载整个网页的情况下,对网页的某部分进行更新

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

## 9. Ajax的原理

- m) Ajax的原理简单来说通过XmlHttpRequest对象来向服务器发异步请求,从服务器获得数据,然后用javascript来操作DOM而更新页面
- n) XMLHttpRequest是ajax的核心机制,它是在IE5中首先引入的,是一种支持异步请求的技术。简单的说,也就是javascript可以及时向服务器提出请求和处理响应,而不阻塞用户。达到无刷新的效果。
- o) 兼容代码在兼容题目中

## 10. 什么是HTTP协议,在打开网页时,HTTP协议作了哪些工作,有哪些步骤?

- p) HTTP 是基于 TCP/IP 协议的应用层协议。它不涉及数据包 (packet) 传输,主要规定了客户端和服务端之间的通信格式,默认使用80端口。
- q) http使得互联网不仅可以传输文字,还能传输图像、视频、二进制文件,其次是规定了许多动词方法(命令): get,post,head,put,patch,options,delete.
- r) 在http1.1中还采用了分块传输编码的方式进行数据传输,在node中就是stream流
- s) http做的工作是:
  - i. 接收客户端的请求,允许的请求命令是: get,post,head,put,patch,options,delete
  - ii. 对客户端的请求做出回应,可以发送文字、图像、视频、二进制文件

- iii. 通过头信息，描述了一些元数据，比如：状态码（status code）、多字符集支持、多部分发送（multi-part type）、权限（authorization）、缓存（cache）、内容编码（content encoding）等，例子：
  - 1. Content-Encoding: gzip
  - 2. Content-Encoding: compress
  - 3. Content-Encoding: deflate
- iv. 而且可以将数据压缩后再发送Content-Encoding字段说明数据的压缩方法，在客户端中，我们也可以通过Accept-Encoding字段说明自己可以接受哪些压缩方法：例子：Accept-Encoding: gzip, deflate
- v. 允许持久连接，对于同一个域名，大多数浏览器允许同时建立6个持久连接
- t) 步骤：
  - i. 1.打开HTTP连接。一定要记住HTTP是一种无状态协议。正因为如此，对于每一个请求你都要建立一个新的连接。
  - ii. 3.设置HTTP请求头。这里面包含要传送的数据类型(二进制)和数据的总长。
  - iii. 4.发送请求。将二进制流写到服务器。
  - iv. 5.读取请求。目标servlet程序将被调用并接受HTTP请求数据。servlet程序就调用所有必要的参数选择相应的方法。注意，如果这是这个客户端的第一次请求，一个服务器对象的新的实例就会被创建。
  - v. 6.调用方法。方法将会被服务器端的对象调用。
  - vi. 7.初始化方法响应。如果调用的方法抛出一个异常，客户将接收到出错信息。否则，返回的类型(如果有)将会被发送。
  - vii. 8.设置HTTP响应头。在响应头中，一定会设置待发送数据的类型和长度。
  - viii. 9.发送响应。二进制数据流将从Web服务器发送并返回给客户端。