

前端开发就业面试题

转行、零基础学习前端开发，欢迎联系李老师 QQ: 1853802745

HTML+CSS (26 题)

1.对 WEB 标准以及 W3C 的理解与认识

标签闭合、标签小写、不乱嵌套、提高搜索机器人搜索几率、使用外链 css 和 js 脚本、结构行为表现的分离、文件下载与页面速度更快、内容能被更多的用户所访问、内容能被更广泛的设备所访问、更少的代码和组件，容易维护、改版方便，不需要变动页面内容、提供打印版本而不需要复制内容、提高网站易用性；

2.xhtml 和 html 有什么区别

HTML 是一种基本的 WEB 网页设计语言，XHTML 是一个基于 XML 的置标语言
最主要的不同：

XHTML 元素必须被正确地嵌套。

XHTML 元素必须被关闭。

标签名必须用小写字母。

XHTML 文档必须拥有根元素。

3.Doctype? 严格模式与混杂模式-如何触发这两种模式，区分它们有何意义?

用于声明文档使用那种规范 (html/xhtml) 一般为 严格 过度 基于框架的 html 文档
加入 XMI 声明可触发，解析方式更改为 IE5.5 拥有 IE5.5 的 bug

4.行内元素有哪些?块级元素有哪些?CSS 的盒模型?

块级元素: div p h1 h2 h3 h4 form ul

行内元素: a b br i span input select

Css 盒模型:内容, border ,margin, padding

5.CSS 引入的方式有哪些? link 和 @import 的区别是?

内联 内嵌 外链 导入

区别 : 同时加载

前者无兼容性，后者 CSS2.1 以下浏览器不支持

Link 支持使用 javascript 改变样式，后者不可

6.CSS 选择符有哪些?哪些属性可以继承?优先级算法如何计算?内联和 important 哪个优先级高?

标签选择符 类选择符 id 选择符

继承不如指定 Id>class>标签选择

后者优先级高

7.前端页面有哪三层构成，分别是什么?作用是什么?

结构层 Html 表示层 CSS 行为层 js

8.css 的基本语句构成是?

选择器{属性 1:值 1;属性 2:值 2;.....}

9.你做的页面在哪些浏览器测试过?这些浏览器的内核分别是什么?

ie(ie 内核) 火狐 (Gecko) 谷歌 (webkit) opear(Presto)

10.写出几种 IE6 BUG 的解决方法

1.双边距 BUG float 引起的 使用 display

2.3 像素问题 使用 float 引起的 使用 display:inline -3px

3.超链接 hover 点击后失效 使用正确的书写顺序 link visited hover active

4.ie z-index 问题 给父级添加 position:relative

5.Png 透明 使用 js 代码 改

6.Min-height 最小高度 ! Important 解决'

7.select 在 ie6 下遮盖 使用 iframe 嵌套

8.为什么没有办法定义 1px 左右的宽度容器 (IE6 默认的行高造成的, 使用 over:hidden,zoom:0.08 line-height:1px)

11.标签上 title 与 alt 属性的区别是什么?

Alt 当图片不显示是 用文字代表。

Title 为该属性提供信息

12.描述 css reset 的作用和用途。

Reset 重置浏览器的 css 默认属性 浏览器的品种不同, 样式不同, 然后重置, 让他们统一

13.解释 css sprites, 如何使用。

Css 精灵 把一堆小的图片整合到一张大的图片上, 减轻服务器对图片的请求数量

14.浏览器标准模式和怪异模式之间的区别是什么?

盒子模型 渲染模式的不同

使用 window.top.document.compatMode 可显示为什么模式

15.你如何对网站的文件和资源进行优化?期待的解决方案包括:

文件合并

文件最小化/文件压缩

使用 CDN 托管

缓存的使用

16.什么是语义化的 HTML?

直观的认识标签 对于搜索引擎的抓取有好处

17.清除浮动的几种方式, 各自的优缺点

1.使用空标签清除浮动 clear:both (理论上能清楚任何标签, , , 增加无意义的标签)

2.使用 overflow:auto (空标签元素清除浮动而不得不增加无意义代码的弊端,,使用 zoom:1 用于兼容 IE)

3.是用 afert 伪元素清除浮动(用于非 IE 浏览器)

18、如何居中 div?如何居中一个浮动元素? 给 div 设置一个宽度, 然后添加 margin:0 auto 属性

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

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%; top:50%;}
```

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

1.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. 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。

19、CSS3 新增伪类举例:

p:first-of-type 选择属于其父元素的首个元素的每个元素。

p:last-of-type 选择属于其父元素的最后元素的每个元素。p:only-of-type 选择属于其父元素唯一的元素的每个元素。p:only-child 选择属于其父元素的唯一子元素的每个元素。

p:nth-child(2) 选择属于其父元素的第二个子元素的每个元素。:enabled :disabled 控制表单控件的禁用态。:checked 单选框或复选框被选中。

20、HTML5 的离线储存有几种方式?

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。

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

cookie 在浏览器和服务器间来回传递。sessionStorage 和 localStorage 不会 sessionStorage 和 localStorage 的存储空间更大;sessionStorage 和

localStorage 有更多丰富易用的接口;sessionStorage 和 localStorage 各自独立的存储空间;

22、简述一下 src 与 href 的区别

href 是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, 用于超链接。

src 是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，图片和框架等元素也如此，类似于将所指向资源嵌入当前标签内。这也是为什么将 js 脚本放在底部而不是头部。

23、简述同步和异步的区别

同步是阻塞模式，异步是非阻塞模式。

同步就是指一个进程在执行某个请求的时候，若该请求需要一段时间才能返回信息，那么这个进程将会一直等待下去，直到收到返回信息才继续执行下去；

异步是指进程不需要一直等下去，而是继续执行下面的操作，不管其他进程的状态。当有消息返回时系统会通知进程进行处理，这样可以提高执行的效率。

24.HTML5 为什么只需要写 !DOCTYPE HTML?

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。

25.CSS 清除浮动的几种方法（至少两种）

使用带 clear 属性的空元素

使用 CSS 的 overflow 属性；

使用 CSS 的 :after 伪元素；

使用邻接元素处理；

26..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

Javascript

1、什么是 JavaScript？（这是基本题，对很多程序员来说也是送分题！）

JavaScript 是客户端和服务端脚本语言，可以插入到 HTML 页面中，并且是目前较热门的 Web 开发语言。同时，JavaScript 也是面向对象编程语言。

类似的基本题目还包括：JavaScript 都有哪些类型？JavaScript 是谁发明的？.....

2、列举 Java 和 JavaScript 之间的区别？

Java 是一门十分完整、成熟的编程语言。相比之下，JavaScript 是一个可以被引入 HTML 页面的编程语言。这两种语言并不完全相互依赖，而是针对不同的意图而设计的。Java 是一种面向对象编程（OOPS）或结构化编程语言，类似的如 C++ 或 C，而 JavaScript 是客户端脚本语言，它被称为非结构化编程。

3. JavaScript 和 ASP 脚本相比，哪个更快？

JavaScript 更快。JavaScript 是一种客户端语言，因此它不需要 Web 服务器的协助来执行。另一方面，ASP 是服务器端语言，因此总是比 JavaScript 慢。值得注意的是，JavaScript 现在也可用于服务器端语言（nodejs）。

4、什么是负无穷大？

负无穷大是 JavaScript 中的一个数字，可以通过将负数除以零来得到。

5、如何将 JavaScript 代码分解成几行吗？

在字符串语句中可以通过在第一行末尾使用反斜杠 “\” 来完成

例：document.write("This is \a program");

如果不是在字符串语句中更改为新行，那么 JavaScript 会忽略行中的断点。

例：

```
var x=1, y=2,
```

```
z=
```

```
x+y;
```

上面的代码是完美的，但并不建议这样做，因为阻碍了调试。

6、什么是未声明和未定义的变量？

未声明的变量是程序中不存在且未声明的变量。如果程序尝试读取未声明变量的值，则会遇到运行时错误。未定义的变量是在程序中声明但尚未给出任何值的变量。如果程序尝试读取未定义变量的值，则返回未定义的值。

7、如何编写可动态添加新元素的代码？

```
1<html>
2<head> <title>t1</title>
3<script type="text/javascript">
4function addNode() { var newP = document.createElement("p");
5var textNode = document.createTextNode(" This is a new text node");
6newP.appendChild(textNode);
7document.getElementById("firstP").appendChild(newP); }
8</script> </head>
9<body> <p id="firstP">firstP<p> </body>
</html>
```

8、什么是全局变量？这些变量如何声明，使用全局变量有哪些问题？

全局变量是整个代码长度可用的变量，也就是说这些变量没有任何作用域。var 关键字用于声明局部变量或对象。如果省略 var 关键字，则声明一个全局变量。

例：// Declare a global globalVariable = “Test” ;

使用全局变量所面临的问题是本地和全局变量名称的冲突。此外，很难调试和测试依赖于全局变量的代码。

9、解释 JavaScript 中定时器的工作？如果有，也可以说明使用定时器的缺点？

定时器用于在设定的时间执行一段代码，或者在给定的时间间隔内重复该代码。这通过使用函数 setTimeout, setInterval 和 clearInterval 来完成。

setTimeout (function, delay) 函数用于启动在所述延迟之后调用特定功能的定时器。

`setInterval (function, delay)` 函数用于在提到的延迟中重复执行给定的功能，只有在取消时才停止。

`clearInterval (id)` 函数指示定时器停止。

定时器在一个线程内运行，因此事件可能需要排队等待执行。

10、`ViewState` 和 `SessionState` 有什么区别？

“`ViewState`” 特定于会话中的页面。

“`SessionState`” 特定于可在 Web 应用程序中的所有页面上访问的用户特定数据。

11、什么是 `===` 运算符？

`===` 被称为严格等式运算符，当两个操作数具有相同的值而没有任何类型转换时，该运算符返回 `true`。

12、说明如何使用 JavaScript 提交表单？

要使用 JavaScript 提交表单，请使用

`document.form [0].submit ()` ;

13、元素的样式/类如何改变？

可以通过以下方式完成：

`document.getElementById(“myText”).style.fontSize = “20px”;`

或

`document.getElementById(“myText”).className = “anyclass”;`

14、JavaScript 中的循环结构都有什么？

`For`、`While`、`do-while loops`

15、如何在 JavaScript 中将 `base` 字符串转换为 `integer`？

`parseInt()` 函数解析一个字符串参数，并返回一个指定基数的整数。`parseInt ()` 将要转换的字符串作为其第一个参数，第二个参数是给定字符串的基础。

为了将 `4F`（基数 16）转换为整数，所使用的代码是：`parseInt (“4F”, 16);`

16、说明 “`==`” 和 “`===`” 之间的区别？

“`==`” 仅检查值相等，而 “`===`” 是一个更严格的等式判定，如果两个变量的值或类型不同，则返回 `false`。

17、`3 + 2 + “7”` 的结果是什么？

由于 3 和 2 是整数，它们将直接相加。由于 7 是一个字符串，它将会被直接连接，所以结果将是 57。

18、说明如何检测客户端机器上的操作系统？

为了检测客户端机器上的操作系统，应使用 `navigator.appVersion` 字符串（属性）。

19、JavaScript 中的 `NULL` 是什么意思？

`NULL` 用于表示无值或无对象。它意味着没有对象或空字符串，没有有效的布尔值，没有数值和数组对象。

20、`delete` 操作符的功能是什么？

`delete` 操作符用于删除程序中的所有变量或对象，但不能删除使用 `VAR` 关键字声明的变量。

21、JavaScript 中有哪些类型的弹出框？

`Alert`、`Confirm` and、`Prompt`

22、`Void (0)` 怎么用？

`Void (0)` 用于防止页面刷新，并在调用时传递参数 “zero”。

`Void (0)` 用于调用另一种方法而不刷新页面。

23、如何强制页面加载 JavaScript 中的其他页面？

必须插入以下代码才能达到预期效果：

```
1<script language="JavaScript" type="text/javascript" >
2<!-- location.href="http://newhost/newpath/newfile.html"; //--></script>
```

24、escape 字符是用来做什么的？

使用特殊字符（如单引号，双引号，撇号和&符号）时，将使用转义字符（反斜杠）。在字符前放置反斜杠，使其显示。

例：

```
1 document.write "I'm a 'good' boy"
2 document.write "I'm a \"good\" boy"
```

25、什么是 JavaScript Cookie？

Cookie 是用来存储计算机中的小型测试文件，当用户访问网站以存储他们需要的信息时，它将被创建。

26、解释 JavaScript 中的 pop（）方法？

pop（）方法与 shift（）方法类似，但不同之处在于 Shift 方法在数组的开头工作。此外，pop（）方法将最后一个元素从给定的数组中取出并返回。然后改变被调用的数组。

例：

```
var cloths = [ "Shirt", "Pant", "TShirt" ];
cloths.pop();
//Now cloth becomes Shirt,Pant
```

27、在 JavaScript 中使用 innerHTML 的缺点是什么？

如果在 JavaScript 中使用 innerHTML，缺点是：内容随处可见；不能像“追加到 innerHTML”一样使用；即使你使用 += like “innerHTML = innerHTML + 'html'”旧的内容仍然会被 html 替换；整个 innerHTML 内容被重新解析并构建成元素，因此它的速度要慢得多；innerHTML 不提供验证，因此我们可能会在文档中插入有效的和破坏性的 HTML 并将其中断。

28、break 和 continue 语句的作用？

Break 语句从当前循环中退出。

continue 语句继续下一个循环语句。

29、在 JavaScript 中，datatypes 的两个基本组是什么？

Primitive

Reference types

原始类型是数字和布尔数据类型。引用类型是更复杂的类型，如字符串和日期。

30、如何创建通用对象？

通用对象可以创建为：

```
var l = new object();
```

31、operator 类型用来做什么？

'typeof' 是一个运算符，用于返回变量类型的字符串描述。

32、哪些关键字用于处理异常？

try... Catch-finally 用于处理 JavaScript 中的异常。

```

1 Try{
2 Code
3 }
4 Catch(exp){
5 Code to throw an exception
6 }
7 Finally{
8 Code runs either it finishes successfully or after catch
9 }

```

33、JavaScript 中不同类型的错误有几种？

有三种类型的错误：

Load time errors: 该错误发生于加载网页时，例如出现语法错误等状况，称为加载时间错误，并且会动态生成错误。

Run time errors: 由于在 HTML 语言中滥用命令而导致的错误。

Logical Errors: 这是由于在具有不同操作的函数上执行了错误逻辑而发生的错误。

34、在 JavaScript 中使用的 Push 方法是什么？

push 方法用于将一个或多个元素添加或附加到数组的末尾。使用这种方法，可以通过传递多个参数来附加多个元素。

35、什么是 JavaScript 中的 unshift 方法？

Unshift 方法就像在数组开头工作的 push 方法。该方法用于将一个或多个元素添加到数组的开头。

36、对象属性如何分配？

属性按以下方式分配给对象：

obj["class"] = 12;

或

obj.class = 12;

37、获得 CheckBox 状态的方式是什么？

alert (document.getElementById ('checkbox1') .checked) ;

如果 CheckBox 被检查，此警报将返回 TRUE。

38、解释 window.onload 和 onDocumentReady？

在载入页面的所有信息之前，不运行 onload 函数。这导致在执行任何代码之前会出现延迟。

onDocumentReady 在加载 DOM 之后加载代码。这允许早期的代码操纵。

39、你将如何解释 JavaScript 中的闭包？ 什么时候使用？

Closure 是与函数返回时保留在内存中的函数相关的本地声明变量。

40、一个值如何附加到数组？

可以以给定的方式将值附加到数组：

arr[arr.length] = value;

41、解释 for-in 循环？

for-in 循环用于循环对象的属性。

for-in 循环的语法是：

```
1 for (variable name in object){  
2 statement or block to execute  
3 }
```

在每次循环中，来自对象的一个属性与变量名相关联，循环继续，直到对象的所有属性都被耗尽。

42、描述 JavaScript 中的匿名函数？

被声明为没有任何命名标识符的函数被称为匿名函数。一般来说，匿名函数在声明后无法访问。

匿名函数声明：

```
1 var anon = function() {  
2 alert('I am anonymous');  
3 };  
4 anon();
```

43、.call () 和.apply () 之间有什么区别？

函数.call () 和.apply () 在使用上非常相似，只是有一点区别。当程序员知道函数参数的编号时，使用.call ()，因为它们必须在调用语句中被提及为参数。另一方面，当不知道数字时使用.apply ()，函数.apply () 期望参数为数组。

.call () 和.apply () 之间的基本区别在于将参数传递给函数。它们的用法可以通过给定的例子进行说明。

```

1 var someObject = {
2   myProperty : 'Foo',
3   myMethod : function(prefix, postfix) {
4     alert(prefix + this.myProperty + postfix);
5   }
6 };
7 someObject.myMethod('<', '>'); // alerts '<Foo>'
8 var someOtherObject = {
9   myProperty : 'Bar'
10};
11someObject.myMethod.call(someOtherObject, '<', '>'); // alerts '<Bar>'
12someObject.myMethod.apply(someOtherObject, ['<', '>']); // alerts '<Bar>'

```

44、定义事件冒泡？

JavaScript 允许 DOM 元素嵌套在一起。在这种情况下，如果单击子级的处理程序，父级的处理程序也将执行同样的工作。

45、什么样的布尔运算符可以在 JavaScript 中使用？

“And”运算符（&&），'Or'运算符（||）和'Not'运算符（!）可以在 JavaScript 中使用。

*运算符没有括号。

46、一个特定的框架如何使用 JavaScript 中的超链接定位？

可以通过使用“target”属性在超链接中包含所需帧的名称来实现。

```
<a href="newpage.htm" target="newframe">>New Page</a>
```

47、在 web-garden 和 web-farm 之间有何不同？

web-garden 和 web-farm 都是网络托管系统。唯一的区别是 web-garden 是在单个服务器中包含许多处理器的设置，而 web-farm 是使用多个服务器的较大设置。

48、如何分配对象属性？

将属性分配给对象的方式与赋值给变量值相同。例如，表单对象的操作值以下列方式分配为

```
“submit”：Document.form.action = “submit”
```

49、在 JavaScript 中读取和写入文件的方法是什么？

可以通过使用 JavaScript 扩展（从 JavaScript 编辑器运行），打开文件的示例来完成：

```
fh = fopen(getScriptPath(), 0);
```

50、在 JavaScript 中如何使用 DOM？

DOM 代表文档对象模型，并且负责文档中各种对象的相互交互。DOM 是开发网页所必需的，其中包括诸如段落，链接等对象。可以操作这些对象以包括添加或删除等操作，DOM 还需要向网页添加额外的功能。除此之外，API 的使用比其他更有优势。

51、JavaScript 中如何使用事件处理程序？

事件是由用户生成活动（例如单击链接或填写表单）导致的操作。需要一个事件处理程序来管理所有这些事件的正确执行。事件处理程序是对象的额外属性。此属性包括事件的名称以及事件发生时采取的操作。

52、解释延迟脚本在 JavaScript 中的作用？

默认情况下，在页面加载期间，HTML 代码的解析将暂停，直到脚本停止执行。这意味着，如果服务器速度较慢或者脚本特别沉重，则会导致网页延迟。在使用 **Deferred** 时，脚本会延迟执行直到 HTML 解析器运行。这减少了网页加载时间，并且它们的显示速度更快。

53、JavaScript 中的各种功能组件是什么？

JavaScript 中的不同功能组件是：

First-class 函数：JavaScript 中的函数被用作第一类对象。这通常意味着这些函数可以作为参数传递给其他函数，作为其他函数的值返回，分配给变量，也可以存储在数据结构中。

嵌套函数：在其他函数中定义的函数称为嵌套函数。

54、解释 unshift（）方法？

该方法在数组启动时起作用，与 **push（）** 不同。它将所需数量的元素添加到数组的顶部。例如：

```
1 var name = [ "john" ];  
2 name.unshift( "charlie" );  
3 name.unshift( "joseph", "Jane" );  
4 console.log(name);
```

输出如下所示：

```
["joseph "," Jane ", " charlie ", " john "]
```

55、decodeURI（）和 encodeURI（）是什么？

EncodeURI（） 用于将 URL 转换为十六进制编码。而 **DecodeURI（）** 用于将编码的 URL 转换回正常。

56、为什么不建议在 JavaScript 中使用 innerHTML？

innerHTML 内容每次刷新，因此很慢。在 **innerHTML** 中没有验证的余地，因此，更容易在文档中插入错误代码，从而使网页不稳定。

57、如何在不支持 JavaScript 的旧浏览器中隐藏 JavaScript 代码？

在 **<script>** 标签之后的代码中添加 “**<!--**”，不带引号。

在 **<script>** 标签之前添加 “**// ->**” 代码中没有引号。

旧浏览器现在将 JavaScript 代码视为一个长的 HTML 注释。而支持 JavaScript 的浏览器则将“<!--”和“//>”作为一行注释。

58、javascript 的 typeof 返回哪些数据类型

Object、number、function、boolean、undefined

59、列举 3 种强制类型转换和 2 种隐式类型转换？

强制（parseInt,parseFloat,number）

隐式（== - ===）

60、split() join() 的区别

前者是切割成数组的形式，后者是将数组转换成字符串

61、数组方法 pop() push() unshift() shift()

Push()尾部添加 pop()尾部删除

Unshift()头部添加 shift()头部删除

62、事件绑定和普通事件有什么区别

普通添加事件的方法不支持添加多个事件，最下面的事件会覆盖上面的，而事件绑定（addEventListener）方式添加事件可以添加多个

63、IE 和 DOM 事件流的区别

1.执行顺序不一样、

2.参数不一样

3.事件加不加 on

4.this 指向问题

64、IE 和标准下有哪些兼容性的写法

Var ev = ev || window.event

document.documentElement.clientWidth || document.body.clientWidth

Var target = ev.srcElement || ev.target

65、ajax 请求的时候 get 和 post 方式的区别

一个在 url 后面 一个放在虚拟载体里面

有大小限制

安全问题

应用不同 一个是论坛等只需要请求的，一个是类似修改密码的

67、call 和 apply 的区别

Object.call(this,obj1,obj2,obj3)

Object.apply(this,arguments)

68、ajax 请求时，如何解释 json 数据

使用 eval parse 鉴于安全性考虑 使用 parse 更靠谱

69、写一个获取非行间样式的函数

```
function getStyle(obj,attr,value)
{
  if(!value)
  {
    if(obj.currentStyle)
    {
      return obj.currentStyle[attr]
    }
    else
    {
      obj.getComputedStyle(attr,false)
    }
  }
  else
  {
    obj.style[attr]=value
  }
}
```

70、事件委托是什么

让利用事件冒泡的原理，让自己的所触发的事件，让他的父元素代替执行！

<http://www.webasily.com/?p=78> 例子可见此链接

71、闭包是什么，有什么特性，对页面有什么影响

闭包就是能够读取其他函数内部变量的函数。

<http://blog.csdn.net/gaoshanwudi/article/details/7355794> 此链接可查看（问这个问题的不是一个公司）

72、如何阻止事件冒泡和默认事件

cancelBubble return false

73、添加 删除 替换 插入到某个接点的方法

obj.appendChild()

obj.insertBefore

obj.replaceChild

obj.removeChild

74、解释 jsonp 的原理，以及为什么不是真正的 ajax

动态创建 script 标签，回调函数

Ajax 是页面无刷新请求数据操作

75、javascript 的本地对象，内置对象和宿主对象

本地对象为 array obj regexp 等可以 new 实例化

内置对象为 global Math 等不可以实例化的

宿主为浏览器自带的 document,window 等

76、document load 和 document ready 的区别

Document.onload 是在结构和样式加载完才执行 js

Document.ready 原生种没有这个方法，jquery 中有 \$.ready(function)

77、“==” 和 “===” 的不同

前者会自动转换类型

后者不会

78、javascript 的同源策略

一段脚本只能读取来自于同一天来源的窗口和文档的属性，这里的同一天来源指的是主机名、协议和端口号的组合

79、编写一个数组去重的方法

```
function oSort(arr)
```

```
{
```

```
var result ={};
```

```
var newArr=[];
```

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

```
{
```

```
if(!result[arr])
```

```
{
```

```
newArr.push(arr)
```

```
result[arr]=1
```

```
}
```

```
}
```

```
return newArr
```

```
}
```