

发件人: 1101632336@qq.com
主题:
日期: 2017年3月12日 下午5:47
收件人:

前端面试题

2017年3月12日 星期日
下午5:44

1. 浏览器内核
IE / Trident / -ms-
Chrome, Safari / WebKit / -webkit-
Firefox / Gecko / -moz-
2. 行内块元素的兼容性
`display:inline-block;*display:inline;*zoom:1;`
3. 清除浮动的方式
父级div定义height
结尾处加空<div>clear:both
(推荐)使用:after+zoom
`.clearfolat:after { clear: both; }
.clearfolot { zoom: 1; }`
给父级div定义overflow:hidden/auto
4. box-sizing常用的属性有哪些?分别有什么作用?
box-sizing用于设置盒模型的组成方式,有三个属性:content-box/border-box/inherit
content-box: 元素的高度和宽度不包括盒模型的margin与padding
border-box: 元素的高度和宽度包括盒模型的margin与padding
inherit: 通过父元素的定义继承
5. Doctype作用?标准模式与兼容模式各有什么区别?
<!DOCTYPE>告知浏览器的解析器用什么文档标准解析这个文档
标准模式的排版和js解析器都是以浏览器支持的最高标准运行
兼容模式采用宽松的向后兼容,保证老旧的浏览器也能正常工作
6. 为什么H5只需要写<!DOCTYPE HTML>就可以声明
因为H4需要对DTD进行引用,而H5并不需要对DTD进行引用,只需要声明即可
7. 页面导入样式时,使用link和@import有什么区别
link属于XHTML标签,除了可以引用CSS外还可以定义rel属性连接等功能;@import之能引用css
link引用的css会和页面同时加载;@import导入的css只会在页面加载完后再加载
@import是css2.1提出的,IE5以上才能识别,link无兼容性问题
8. 请介绍你对浏览器内核的理解
浏览器内核分为两部分,渲染引擎和js引擎;
渲染引擎用于对页面取得页面资源和内容,整理样式,计算网页的显示方式,将页面正确的显示出来;
js引擎用于解析和执行Javascript语言来实现网页动态和功能
9. HTML5有哪些新特性,如何处理兼容性问题
H5比H4新增了canvas绘图、语义化标签、video和audio元素、localStorage和sessionStorage、新的表单类型(email, url, date, time, url, search, calendar)
旧版IE可以通过document.createElement方法产生新标签来支持H5特性,也可以使用HTML5开源框架实现
10. H5语义化标签的理解
H5语义化标签有nav、footer、article、header、section等,虽然他们功能等同于div,但是对内容的结构化,让代码更加有条理,更容易让浏览器解析,有利于SEO
11. Javascript基本数据类型
Undefined, Null, Boolean, Number, String
12. js的内置对象
数据封装类型: Array, Object, Boolean, Number, String
其他: Function, Math, Date
13. this的理解
this总是指向函数的直接调用者,如果有new那么this指向new出来的那个对象,在事件中this指向触发事件的对象
14. eval的作用是什么
将字符串解析成js代码并运行
15. DOM结点的操作
createElement(), createTextNode(), appendNode(), removeNode(), replaceNode(), insertBefore()
16. null和undefined的区别
null表示空的对象,undefined表示无的原始值,转为数值类型为NaN
undefined:
变量声明了但没赋值,调用函数时需要提供的参数没有提供,函数没有返回值返回undefined
null:

作为函数的参数表示没有传入对象,null也是原型链的终点

17. new操作具体做了什么

创建一个空对象,并将this指向该对象,同时继承函数的原型
将属性和方法加入this指向的对象中,并隐式的返回this

18. call()和apply()的作用和区别

作用: 调用某一对象的某一方法,用另一个对象替代它进行调用
call的第一个参数传入替代对象,后面的参数为传入的参数列
apply的第二个参数传入一个数组,在数组中声明传入的参数列

19. HTTP状态码

100 - 继续,正在处理请求
200 - 正确接受并处理
301 - 永久重定向 302 - 临时重定向 304 - 未发生改变
401 - 没有权限访问 403 - 禁止访问 404 - 资源不存在
500 - 服务器处理错误 503 - 服务器暂时无法处理请求(过载或维护)

20. 性能优化的方法

将css,js代码压缩,将图片等资源文件控制在合理的范围内
减少请求发送次数,使用缓存避免重复数据获取
使用前端模板引擎替代后台渲染
尽可能少使用DOM操作,大量修改样式时通过修改类而不是频繁的修改样式
样式预加载,将样式放在顶部,将js放在底部

21. 什么叫优雅降级和渐进增强

优雅降级指的是让web站点在所有新式浏览器都可以正常工作,如果旧式浏览器,则判断是否可以正常工作,如果不能则需要为浏览器增加替补方案
渐进增强指的是先从所有浏览器都支持的最基本功能开始做起,逐步添加只有新式浏览器才支持的但无害于基础浏览器的增强样式与功能

22. 哪些常见操作会造成内存泄漏

内存泄漏指的是任何对象在不拥有或不是用时依旧存在,垃圾回收期会定时扫描对象,如果对象没有被应用则该对象会被内存所回收;
setTimeout如果第一个参数不是函数而是字符串就会发生内存泄漏

23. 请简述cookie、sessionStorage和localStorage的区别

cookie会在浏览器和服务端间进行传递,用于身份的储存
sessionStorage和localStorage不会传递到服务器中,只会在浏览器中进行存储,并且存储容量比cookie大得多,在浏览器关闭窗口后,sessionStorage存储的信息将会清楚,而localStorage如果不手动清楚,则会一直保存

24. 使用iframe的缺点

iframe会阻塞onload事件的发生,并且搜索引擎无法读取iframe的内容,不利于SEO
iframe和页面会共享连接,浏览器在相同域连接有限制,则会影响性能
通过Javascript给iframae动态添加src即可解决问题

25. 如何实现浏览器多标签间的通信

可以通过WebSocket进行通信,也可以调用本地存储的方式进行

26. 如何在页面上实现一个圆形可点击区域

通过圆角div,svg,canvas绘图都可以实现

27. 介绍一下标准盒模型,低版本IE盒模型有什么区别

标准的w3c盒模型包括内容、内边距、边框、外边距构成
IE的盒模型将margin和padding部分算到了内容中

28. css优先级如何计算

就近原则,同样权重下近者优先
权重的优先级为: !important > 行内 > id > class > tag

29. 为什么要使用CSS Sprites

CSS Sprites就是将网页中的一些小背景图片整合到一张图片中去,再利用background-image和background-position对需要的部分进行定位

30. display:none和visibility:hidened的区别

display:none - 隐藏对应的元素,并且不会为该元素分配空间
visibility:hidden - 隐藏对应元素,但是为其分配的空间仍旧保存

31. 如何使用js操作cookie

```
setCookie(name, value, day) {
    var d = new Date();
    d.setTime(d.getTime() + day*24*60*60*1000);
    var expires = "expires=" + d.getGMTString();
    document.cookie = name + "=" + value + ";" + expires;
}

getCookie(name) {
    var name = name + "=";
    var arr = document.cookie.split(";");
    for (var i = 0; i < arr.length; ++i) {
        var tmp = arr.trim();
        if (tmp.indexOf(name) == 0) {
            return tmp.substring(name.length, tmp.length);
        }
    }
    return "";
}
```

```

removeCookie(name) {
    var time = new Date();
    time.setTime(time.getTime() - 1);
    var expires = time.toGMTString() + ";";
    document.cookie = name + "=" + "expires=" + expires;
}

```

32. ajax有哪些优缺点,如何解决跨域问题

优点: ajax通过异步请求,提升了用户体验,又花了浏览器与服务器的传输,分担了服务器的部分性能,可以实现动态不刷新

缺点: ajax暴露了与服务器的交互细节,对搜索引擎支持较差,不易于SEO,不容易调试

解决跨域问题

在ajax方法中添加jsonp:jsonpcallback属性

通过\$.jsonp方法进行ajax交互

添加header头("Access-Control-Allow-Origin: *")

33. Javascript原型,原型链,有什么特点?

原型对象也是普通的对象,自带隐式的__proto__属性,原型也有自己的原型,构成了一个原型链

原型链是由一些继承和共享属性的对象组成的

34. 请解释js的同源策略

同源策略是客户端脚本的重要安全度标准,目的是防止某个文档某脚本通过不同源装载

35. 什么是闭包

闭包就是函数与其相关引用变量构成的实体

特点: 使用闭包可以减少全局变量的使用,还可以减少函数参数的数量,更利于封装

闭包中函数对自由变量是引用而不是复制,父函数没调用一次都会产生不同的闭包

36. js中的继承是如何实现的,如何避免在原型链上的对象共享

浅拷贝

```

function extend(p, c) {
    var c = c || {};
    for (var prop in p) {
        c[prop] = p[prop];
    }
}

```

深拷贝

```

function extendDeeply(p, c) {
    var c = c || {};
    for (var prop in p) {
        if (typeof p[prop] === "object") {
            c[prop] = (p[prop].constructor === Array)?[]:{};
            extendDeeply(p[pros]. c[pros]);
        } else {
            c[prop] = p[prop];
        }
    }
}

```

37. 原生ajax调用过程

创建XMLHttpRequest对象

准备发送的数据

调用XMLHttpRequest对象的open方法

调用XMLHttpRequest对象的send方法

获取异步调用返回的数据

使用js操作dom实现局部更新

38. 一个页面从输入url到页面加载显示完成,其中都发生了什么

查找浏览器缓存

DNS解析,查找该域名对应IP地址

进行http协议会话

发送请求报头

服务器反馈相应报头

html文档开始下载

文档显示

39. css3动画

使用@keyframes属性定义动画,用animation来为节点绑定对应的动画

```

@keyframes myAnimation {
    from: xxx
    to: xxx
}


animation的属性: 绑定@keyframes的名称,动画时长,速度曲线,播放次数,是否逆向播放等



transition是用于过渡css的属性变更,有变更属性名,变更时长,速度曲线,合适开始四个属性



### 40. html5多线程



```

var worker = new Worker("worker.js");
worker.postMessage("client send...");
onmessage = fucntion(event) {
 var data = event.data;
 postMessage("send to client...")
 close();
}

```


```

```

},
worker.onmessage = function(event) {
    var data = event.data;
}

```

41. js事件模型和事件流

js事件就是当js和dom连接的桥梁,当dom被触发特定事件时,js就会被调用特定的代码给出响应
 事件流指的是触发元素节点到根节点之间,事件会按照特定的顺序进行传播
 两种事件模型: 捕获型事件会从根节点到指定节点传播,冒泡型事件会在指定节点向根节点进行传播
 DOM事件流采用捕获+冒泡事件处理,在事件捕获阶段自根节点向下进行传播,传播到特定节点后进入冒泡阶段再向上进行传播
 事件代理的处理方式就是避免在所有子元素添加大量监听,可以通过给父元素添加监听,然后判断触发者来实现,这样可以避免定义大量的监听函数导致性能下降

42. javascript执行如何解析

首先是全局预处理阶段,解析器创建池法环境,扫面全局声明式定义的函数以及var形式定义的变量,将它们的名称以及对应的应用放到池法环境当中;然后进入全局执行阶段,分别对预处理阶段处理的成员赋值;函数预处理阶段会在函数内部建立池法环境,扫描其内部声明的函数以及变量,将其名称存放到池法环境中,且该池法环境无法被外界调用,然后执行函数执行阶段,分别对函数内部的池法环境中的成员进行赋值

43. CSS兼容性处理

DOCTYPE影响css的兼容性,使用@import导入样式表也会因为兼容性遇到问题
 可以使用各个浏览器内核的前缀为不同的浏览器定义不同的样式
 IE不支持!important属性,需要在声明的属性前加*来替代!important
 IE下的盒模型与其他主流浏览器不相同
 IE与Firefox在透明度不相同
 Firefox的默认字体是13px,IE是16px

44. 使用css3实现360度旋转

```

.div {
    transition: All 0.5s;
}
.div:hover {
    transform: rotate(360deg);
}

```

45. 选择器有哪些

基本选择器(id,class,tag,*)
 层次选择器
 过滤选择器
 表单选择器

46. 如何区分不同浏览器

```
var userAgent = navigator.userAgent; //取得浏览器的userAgent字符串
```

47. 函数声明的方式有哪些,有什么区别?

有三种方式: 函数声明,函数表达式,匿名函数
 函数声明会在js解析最开始创建池法环境时就会将函数的声明存放其中,而函数的表达式定义只会在真正调用该函数时才会读取,匿名函数不需要声明函数名,代码执行到匿名函数会自动调用

48. BOM的常用属性

location对象(href,search,hash,host,hostname,pathname)
 history对象(go,back,forward)
 navigator(userAgent,cookieEnabled)
 screen(width,height,availHeight,availWidth)

49. PC与移动端开发有什么不同

页面承载量不同,交互方式不同,使用场景不同,网络接入不同

50. 如何设置响应式布局

```
<meta name="viewport" content="width=device-width,initial-scale=1,maximum-scale=1" />
```

51. Bootstrap响应式布局是如何实现的

```

@media (min-width:768px) {}
@media (max-width:769px) and (min-width:992px) {}
@media (max-width:992px) and (min-width: 1200px) {}
@media (max-width: 1200px) {}

```

52. 为什么频繁操作dom节点的代价比较大

DOM本身也是一个对象,操作DOM对象过后,会触发浏览器的layout和paint行为,这些行为非常的耗费性能

53. http1.0和http1.1有什么区别?http2有什么新特性

http1.0默认是短链接,http1.1默认为长链接,http2.0新增了多路复用的技术,可以同一连接发送多个请求