

## H5面试题

---

1、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

2、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

链接：<http://www.cnblogs.com/yizuierguo/archive/2009/07/15/1524106.html>

3、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

其实就是把网页中一些背景图片整合到一张图片文件中，再利用CSS的“background-image”，“background-repeat”，“background-position”的组合进行背景定位，background-position可以用数字精确的定位出背景图片的位置。

链接：<https://www.toptal.com/developers/css/sprite-generator>

4、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

5、清除浮动的几种方式，各自的优缺点

- 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）

- 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)
- 是用afert伪元素清除浮动(用于非IE浏览器)

## 6、数组方法pop() push() unshift() shift() splice() slice()

Push()尾部添加 pop()尾部删除

Unshift()头部添加 shift()头部删除

## 7、call和apply的区别

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

Object.apply(this,arguments)//arguments 为数组

//此知识点要结合JS继承

## 8、Javascript的typeof返回哪些数据类型

object number string boolean underfind null

## 9、规避javascript多人开发函数重名问题

- 命名空间
- 封闭空间 =》》》 闭包???
- Js模块化mvc (数据层、表现层、控制层)
- sea.js /require.js
- 变量转换成对象的属性

## 10、js继承三种方式

### 1)、js原型 (prototype) 实现继承

```
<script type="text/javascript">
function Person(name,age){
    this.name=name;
    this.age=age;
}
Person.prototype.sayHello=function(){
    alert("使用原型得到Name: "+this.name);
}
```

```

}
var per=new Person("马小倩",21);
per.sayHello(); //输出： 使用原型得到Name:马小倩

```

```

function Student(){}
Student.prototype=new Person("洪如彤",21);
var stu=new Student();
Student.prototype.grade=5;
Student.prototype.intr=function(){
    alert(this.grade);
}
stu.sayHello();//输出： 使用原型得到Name:洪如彤
stu.intr();//输出： 5

```

</script>

## 2) 构造函数实现继承

```

<script type="text/javascript">
    function Parent(name){
        this.name=name;
        this.sayParent=function(){
            alert("Parent:"+this.name);
        }
    }

    function Child(name,age){
        this.tempMethod=Parent;
        this.tempMethod(name);
        this.age=age;
        this.sayChild=function(){
            alert("Child:"+this.name+"age:"+this.age);
        }
    }

    var parent=new Parent("江剑臣");
    parent.sayParent(); //输出：“Parent:江剑臣”
    var child=new Child("李鸣",24); //输出：“Child:李鸣 age:24”
    child.sayChild();
</script>

```

## 3) call , apply实现继承

```

<script type="text/javascript">
    function Person(name,age,love){
        this.name=name;
        this.age=age;
        this.love=love;
        this.say=function say(){
            alert("姓名： "+name);
        }
    }

```

```

    }
}

//call方式
function student(name,age){
    Person.call(this,name,age);
}

//apply方式
function teacher(name,love){
    Person.apply(this,[name,love]);
    //Person.apply(this,arguments); //跟上句一样的效果,
    arguments
}

//call与aply的异同:
//1,第一个参数this都一样,指当前对象
//2,第二个参数不一样: call的是一个个的参数列表; apply的
    是一个数组 (arguments也可以)

var per=new Person("武凤楼",25,"魏荧屏");//输出: "武凤楼"
per.say();
var stu=new student("曹玉",18);//输出: "曹玉"
stu.say();
var tea=new teacher("秦杰",16);//输出: "秦杰"
tea.say();

</script>
Person.prototype.showName = function(){
    alert(this.name);
}

function Worker(name, job){
    Person.apply(this,arguments)
    this.job = job;
}

for(var i in Person.prototype){
    Worker.prototype = Person.prototype;
}
new Worker('sl', 'coders').showName();*/
Var w= new Worker('sl', 'coders');
w. showName();

```

- 11、 JavaScript中如何检测一个变量是一个String类型？请写出函数实现

```
typeof(obj) == 'string'
obj.constructor == String;
```

- 12、 请说出减低页面加载时间的方法

- 1、 压缩css、js文件
- 2、 合并js、css文件，减少http请求
- 3、 外部js、css文件放在最底下
- 4、 减少dom操作，尽可能用变量替代不必要的dom操作
- 5、 CSSSprites

- 13、 闭包

- 14、 你能解释一下JavaScript中的继承是如何工作的吗？

```
/*子构造函数中执行父构造函数，并用call\apply改变this
克隆父构造函数原型上的方法*/
```

- 15、 什么是FOUC？你如何来避免FOUC？

FOUC浏览器样式闪烁，之前也听说过一些类似的东西，比如样式突变等等，但这东西竟然有学名的。。什么是FOUC(文档样式短暂失效)?如果使用import方法对CSS进行导入,会导致某些页面在Windows 下的Internet Explorer出现一些奇怪的现象:以无样式显示页面内容的瞬间闪烁,这种现象称之为文档样式短暂失效(Flash of Unstyled Content),简称为FOUC。原因大致为：1，使用import方法导入样式表。2，将样式表放在页面底部3，有几个样式表，放在html结构的不同位置。其实原理很清楚：当样式表晚于结构性html加载，当加载到此样式表时，页面将停止之前的渲染。此样式表被下载和解析后，将重新渲染页面，也就出现了短暂的花屏现象。解决方法：使用LINK标签将样式表放在文档HEAD中。

```
/*由于css引入使用了@import 或者存在多个style标签以及css文件
在页面底部引入使得css文件加载在html之后导致页面闪烁、花屏
用link加载css文件，放在head标签里面*/
```

- 16、 你能描述一下渐进增强和优雅降级之间的不同吗？

```
.transition{
    -webkit-transition: all .5s;
```

```
-ms-transition: all .5s;  
-moz-transition: all .5s;  
-o-transition: all .5s;  
transition: all .5s;  
  
}
```

### 渐进增强

progressive enhancement: 针对低版本浏览器进行构建页面, 保证最基本的功能, 然后再针对高级浏览器进行效果、交互等改进和追加功能达到更好的用户体验。

### 优雅降级

graceful degradation: 一开始就构建完整的功能, 然后再针对低版本浏览器进行兼容。

### 区别:

优雅降级是从复杂的现状开始, 并试图减少用户体验的供给, 而渐进增强则是从一个非常基础的, 能够起作用的版本开始, 并不断扩充, 以适应未来环境的需要。降级(功能衰减)意味着往回看; 而渐进增强则意味着朝前看, 同时保证其根基处于安全地带。

17、html 文档流

18、你如何优化自己的代码?

/\*代码重用

避免全局变量(命名空间, 封闭空间, 模块化mvc..)

拆分函数避免函数过于臃肿

注释

19、

20、介绍一下你对浏览器内核的理解?

渲染引擎:

firefox使用gecko引擎

IE使用Trident引擎

2015年微软推出自己新的浏览器，原名叫斯巴达，后改名edge,使用edge引擎

opera最早使用Presto引擎，后来弃用

chrome\safari\opera使用webkit引擎

13年chrome和opera开始使用Blink引擎

js引擎：

老版本IE使用Jscript引擎

IE9之后使用Chakra引擎

edge浏览器仍然使用Chakra引擎

firefox使用monkey系列引擎

safari使用的SquirrelFish系列引擎

Opera使用Carakan引擎

chrome使用V8引擎。nodeJs其实就是封装了V8引擎

常见内核：

- Trident内核：

IE,MaxThon,TT,The World,360,搜狗浏览器等。[又称MSHTML]

- Gecko内核：

Netscape6及以上版本，FF,MozillaSuite/SeaMonkey等

- Presto内核：

Opera7及以上。 [Opera内核原为：Presto，现为：Blink;]

- Webkit内核：

Safari,Chrome等。 [ Chrome的：Blink (WebKit的分支) ]

主要分成两部分：渲染引擎(layout engineer或Rendering Engine)和JS引擎。

渲染引擎：负责取得网页的内容（HTML、XML、图像等等）、整理讯息（例如加入CSS等），以及计算网页的显示方式，然后会输出至显示器或打印机。浏览器的内核的不同对于网页的语法解释会有不同，所以渲染的效果也不相同。所有网页浏览器、电子邮件客户端以及其它需要编辑、显示网络内容的应用程序都需要内核。

JS引擎则：解析和执行javascript来实现网页的动态效果。

最开始渲染引擎和JS引擎并没有区分的很明确，后来JS引擎越来越独立，内核就倾向于只指渲染引擎。

## 22、行内元素有哪些？块级元素有哪些？空(void)元素有那些？

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

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

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

(3) 常见的空元素：

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

鲜为人知的是：

<area> <base> <col> <command> <embed>

<keygen> <param> <source> <track> <wbr>

## 23、页面导入样式时，使用link和@import有什么区别？

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



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

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

24、html5有哪些新特性、移除了那些元素？

- 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。
- 绘画 canvas；
- 用于媒介回放的 video 和 audio 元素；
- 本地离线存储 localStorage 长期存储数据，浏览器关闭后数据不丢失；
- sessionStorage 的数据在浏览器关闭后自动删除；
- 语意化更好的内容元素，比如 article、footer、header、nav、section；
- 表单控件，calendar、date、time、email、url、search；
- 新的技术webworker, websocket, Geolocation；

移除的元素：

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

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

25、HTML5的离线储存怎么使用，工作原理能不能解释一下？

在用户没有与因特网连接时，可以正常访问站点或应用，在用户与因特网连接时，更新用户机器上的缓存文件。

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

如何使用：

- 1、页面头部像下面一样加入一个manifest的属性；
- 2、在cache.manifest文件的编写离线存储的资源；

```
CACHE MANIFEST
```

```
#v0.11
```

```
CACHE:
```

```
js/app.js
```

```
css/style.css
```

```
NETWORK:
```

```
resource/logo.png
```

```
FALLBACK:
```

```
/ /offline.html
```

3、在离线状态时，操作window.applicationCache进行需求实现。

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

cookie是网站为了标示用户身份而储存在用户本地终端（Client Side）上的数据（通常经过加密）。

cookie数据始终在同源的http请求中携带（即使不需要），记会在浏览器和服务间来回传递。

sessionStorage和localStorage不会自动把数据发给服务器，仅在本地保存。

存储大小：

cookie数据大小不能超过4k。

sessionStorage和localStorage 虽然也有存储大小的限制，但比cookie大得多，可以达到5M或更大。

有期时间：

localStorage 存储持久数据，浏览器关闭后数据不丢失除非主动删除数据；

sessionStorage 数据在当前浏览器窗口关闭后自动删除。

cookie 设置的cookie过期时间之前一直有效，即使窗口或浏览器关闭

27、实现不使用 border 画出1px高的线，在不同浏览器的标准模式与怪异模式下都能保持一致的效果。

```
<div style="height:1px;overflow:hidden;background:red"></div>
```

28、CSS选择符有哪些？哪些属性可以继承？

1. 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)

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

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

## 29、CSS优先级算法如何计算?

\* 优先级就近原则, 同权重情况下样式定义最近者为准;

\* 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

## 30、CSS3新增伪类有那些?

- 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。
- 
- :after 在元素之前添加内容,也可以用来做清除浮动。
- :before 在元素之后添加内容
- :enabled
- :disabled 控制表单控件的禁用状态。
- :checked 单选框或复选框被选中。

### 31、如何居中div? 如何居中一个浮动元素? 如何让绝对定位的div居中?

给div设置一个宽度, 然后添加margin:0 auto属性

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%;
}
```

让绝对定位的div居中-水平居中

```
position: absolute;
width: 1200px;
background: none;
margin: 0 auto;
right: 0;
left: 0;
```

让绝对定位的div居中-水平居中+垂直居中

```
position: absolute;
width: 1200px;
background: none;
margin: auto auto;
top: 0;
left: 0;
bottom: 0;
right: 0;
```

```
.div_fixed{
    position: fixed;
    top: 50%;
    left: 50%;
    background-color: #000;
    width:50%;
    height: 50%;
    -webkit-transform: translateX(-50%) translateY(-50%);
}
```

让div内部的块元素居中

```
display: table-cell;
vertical-align: middle;
```

```
.div_fixed{
  position: fixed;
  top: 50%;
  left: 50%;
  background-color: #000;
  width:50%;
  height: 50%;
  -webkit-transform: translateX(-50%) translateY(-50%);
}
```

32、CSS里的visibility属性有个collapse属性值是干嘛用的？在不同浏览器下以后什么区别？

当一个元素的visibility属性被设置成collapse值后，对于一般的元素，它的表现跟hidden是一样的。但例外的是，如果这个元素是table相关的元素，例如table行，table group，table列，table column group，它的表现却跟display: none一样，也就是说，它们占用的空间也会释放。

但遗憾的是，各种浏览器对collapse值的处理方式不一样。

33、CSS3媒体查询和FLEX布局

34、使用 CSS 预处理器吗？喜欢那个？

SASS (SASS、LESS没有本质区别，只因为团队前端都是用的SASS)

35、JS事件冒泡

事件处理程序中的变量event保存着事件对象。而event.target属性保存着发生事件的目标元素。这个属性是DOM API中规定的，但是没有被所有浏览器实现。jQuery对这个事件对象进行了必要的扩展，从而在任何浏览器中都能够使用这个属性。通过.target，可以确定DOM中首先接收到事件的元素（即实际被单击的元素）。

1、事件冒泡情况

```
<div id="div01">
```

```

<div id="div_inner01"></div>

<input type="button" value="button03" id="btn01">

</div>

<script src="http://ajax.aspnetcdn.com/ajax/jquery/jquery-1.8.0.js">
</script>

<script>
    $(function () {
        $('#div01').click(function (event) {
            $('#div_inner01').toggleClass('hidden')
        })
        $('#btn01').click(function (event) {
            alert('ok');
        })
    })
</script>

```

## 2、解决

### 1) 判定当前事件源是否匹配

```

if(event.target==this){
    $('#div_inner01').toggleClass('hidden');
}

```

### 2) event.stopPropagation();

该函数只阻止事件向祖辈元素的传播，不会阻止该元素自身绑定的其他事件处理函数的函数。event.stopImmediatePropagation()不仅会阻止事件向祖辈元素的传播，还会阻止该元素绑定的其他(尚未执行的)事件处理函数的执行。——该内容尚未验证

由于on()函数并不是将事件处理函数直接绑定到自己身上，而是“委托”绑定到祖辈元素上，由祖辈元素来触发执行。on()函数会先一次性冒泡到文档

的顶部，然后为符合条件的元素触发事件。因此，`stopPropagation()`函数无法阻止on事件的冒泡。

### 36、默认事件

如果我们把单击事件处理程序注册到一个锚元素，而不是一个外层的<div>上，那么就要面对另外一个问题：当用户单击链接时，浏览器会加载一个新页面。这种行为与我们讨论的事件处理程序不是同一个概念，它是单击锚元素的默认操作。类似地，当用户在编辑完表单后按下回车键时，会触发表单的submit事件，在此事件发生后，表单提交才会真正发生。

如果我们不希望执行这种默认操作，那么在事件对象上调用`stopPropagation()`方法也无济于事，因为默认操作不是在正常的事件传播流中发生的。在这种情况下，`.preventDefault()`方法则可以在触发默认操作之前终止事件。

提示 当在事件的环境中完成了某些验证之后，通常会用到`.preventDefault()`。例如，在表单提交期间，我们会对用户是否填写了必填字段进行检查，如果用户没有填写相应字段，那么就需要阻止默认操作。

事件传播和默认操作是相互独立的两套机制，在二者任何一方发生时，都可以终止另一方。如果想要同时停止事件传播和默认操作，可以在事件处理程序中返回false，这是对在事件对象上同时调用`stopPropagation()`和`preventDefault()`的一种简写方式。

### 36、Jquery绑定多个事件，多个元素

```
$('#id').bind('click change',function (event) {...})
```

//多个事件用空格分隔

绑定多个元素到同一个事件（方法多种...）

```
$(document).on('click', '#header .fixed-feedback-bn, #sb-sec .feedback-bn', function () {
    // code... });
```

## 37、JS动态增加、删除节点

```

var li01=document.createElement('li');
li01.setAttribute('value','003');
li01.innerText='china';
var ul01=document.getElementById('ul01');
ul01.appendChild(li01);
ul01.removeChild(li01); //dom

//jquery 如下:
// $('<li value="004">japan</li>').appendTo($('#ul01'))
$('#ul01').append($('<li value="004">japan</li>'))

```

## 38、实现以下函数

```

plus(10,10)=>20

function plus(a,b) {
    return a+b;}

plus(10)(10)=>20

function plus(a) {
    function plus1(b) {
        console.log(a+b);
        return a+b;
    }
    return plus1;
}

plus(10)(10);

```



39、实现以下代码

```
[a,b,3,4,5].copy=>[a,a,b,b,3,3,4,4,5,5]
```

```
Array.prototype.copy=function () {  
    var b=this.map(function (i) {  
        return i+', '+i;  
    });  
    console.log(b);  
}  
var b=new Array(['a','b','c',1]);  
b.copy();
```

40、下面三条语句的区别

```
function show(){}  
var show=function(){}  
var show=new Function()
```

函数，函数表达式，函数对象

41、js对象和jquery对象相互转化

```
var js_obj=document.getElementById('id');  
var jq_obj=$('#id')  
js转jquery对象：  
    $(js_obj)  
jqeuery转js对象：
```

jq\_obj[0] 或者 jq\_obj.get(0) //这两种情况等价

## 42、JSONP

jsonp(JSON with Padding) 是 json 的一种"使用模式", 可以让网页从别的域名(网站)那获取资料, 即跨域读取数据。为什么我们从不同的域(网站)访问数据需要一个特殊的技术(JSONP)呢? 这是因为同源策略。

## 综合参考

1. for循环的变量作用域问题
2. js的闭包
3. call和apply的区别
4. ==和===的区别
5. background包含border吗
6. js怎样创建一个对象
7. js是单线程的还是多线程的?
8. js怎么处理不同浏览器的兼容问题
9. div怎么水平和垂直居中
10. angular的三个特性。有趣点
11. 你在项目中用到了bootstrap的哪些内容;
12. 你用到了哪些框架
13. ajax怎么调用后台数据的(什么方法) ajax的同步异步请求,
14. 怎么清除浮动
15. 网页的优化问题、代码怎么优化
16. 会写接口吗? 你们是怎么调用后台数据的
17. 对mvc的理解, 优缺点
18. div高度塌陷
19. 简述node.js优缺点
20. es6的新特性
21. express的特点
22. npm安装了哪些包
23. 省市联动的问题
24. 响应式布局和栅格布局
25. 使用jQuery的好处?
26. 左中右3个DIV, 如何让中间的DIV居中?
27. 浏览器的兼容问题你如何解决。
28. Ajax有哪些方法吗?
29. Position属性的值有哪些? 怎样使用?
30. Node.js中的模板有哪些? ejs, jade
31. MVC是什么? 如何使用?
32. Doctype有什么作用? 严格模式与混杂模式有什么区别?
33. 了解SASS和Less?
34. 对于前端自动化构建工具有了解吗?
35. 你常用那些开发工具, 有什么优点?
36. 浏览器的内核有哪些?
37. Js如何实现继承?

38. 什么是CSS Hack，作用是什么？
39. W3C标准以及对W3C的理解和认识？
40. 如何使用最少的代码做出当鼠标点击导航栏的时候背景颜色改变，并且让后面的导航栏也发生这种变化？
41. CSS预处理器？
42. XHTML和HTML有什么区别？
43. Js中如何判断数据的类型？
44. 解释一下“==”和“===”
45. Ajax请求时，如何解释Json数据？
46. 什么是事件冒泡，如何阻止？
47. JS中定义类的方法？
48. 事件绑定的几种方法？
49. 开发调试工具和方法有哪些？
50. 如何使用响应适布局？
51. CSS实现垂直水平居中？
52. 怎样得到一个节点的值。
53. 如何消除数组中重复的元素？
54. 如何统计字符串中某个字符的数量？
55. 伪类数组如何转换为标准数组？
56. 快速排序的方法？
57. 谈一谈JavaScript作用域链，JavaScript原型链？
58. 怎样给一个对象动态添加属性和方法？
59. 如何删除、添加、查找节点。
60. JavaScript跨域总结和解决方法。
61. 如何得到一个字符串的字节长度？
62. 什么是FOUC？如何避免FOUC？
63. 什么是语义化的HTML？