# Web前端第一阶段考试题

# 1、HTMl5新增特性有哪些？（\*\*）

(1) 新增了语义化标签

(2) 新增了音频（audio）视频（video）标签

(3) 新增了canvas和svg绘图

(4) 新增了地理定位（navigator.geolocation）

(5) 新增了拖拽API (drapable)

(6) 新增了多线程技术web worker

(7) 新增了即时通讯web socket

# 2、 CSS3 新增特性（\*\*\*）

(1) 增加了transition 渐变 animation动画

(2) 选择器、阴影、弹性布局flex

(3) 媒体查询等

# 3、W3C标准有哪些？（\*）

(1) 标准分为三个层面，结构(html)、行为(js)、表现(css)

## 标准有:

(3) 文档类型需要声明（!DOCTYPE）

(4) 编码格式需要声明 <meta charset=”utf-8”>

(5) javascript代码需要定义

(6) <script language=”javascript” type = ”text/javascript”></script>

(7) css代码需要定义

(8) <style type=”text/css”></style>

(9) 标签名字和标签内部属性名字小写等

# 4、你遇到哪些浏览器兼容问题？您怎么处理浏览器兼容性问题的（\*\*）

所谓的浏览器兼容性问题，是指因为不同的浏览器对同一段代码有不同的解析，造成页面显示效果不统一的情况。在大多数情况下，我们的需求是，

无论用户用什么浏览器来查看我们的网站或者登陆我们的系统，都应该是统一的显示效果。所以浏览器的兼容性问题是前端开发人员经常会碰到和必须

要解决的问题

参考地址: <https://www.cnblogs.com/wxf-h/p/10513342.html>

分为以下几大类：

- **\*\*css兼容性问题\*\***

  1. cursor:hand    VS    cursor:pointer

  ```

  firefox 不支持hand，但ie支持 pointer

  解决方法:统一使用pointer

  ```

  2. innerText在IE中能正常工作，但在FireFox中却不行

  3. CSS透明

  4. 盒子模型

  5. 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

  6. 图片默认有间距

  7. 万能清除浮动

  8. IE捉迷藏的问题

  9. 溢出显示省略号

  10. IE6下图片下有空隙产生

  11. 超链接访问过后hover样式就不出现的问题

## 具体如下：

## CSS里的兼容性问题

### 1.cursor:hand VS cursor:pointer

firefox 不支持hand，但ie支持 pointer

解决方法:统一使用pointer

### 2. innerText在IE中能正常工作，但在FireFox中却不行

解决方法:

需用textContent

if(navigator.appName.indexOf("Explorer")>-1){

document.getElementById('element').innerText = "my text";

}else{

document.getElementById('element').textContent = "my text";

}

### 3. CSS透明

IE：filter:alpha(opacity=60)。

FF：opacity:0.6。

### 4.盒子模型

标准 w3c 盒子模型的范围包括 margin、border、padding、content，并且 content 部分不包含其他部分； IE盒子模型的范围也包括

margin、border、padding、content。和标准 w3c 盒子模型不同的是：ie 盒子模型的 content 部分包含了 border 和 padding。

因此，问题主要表现在css中的width是否计算border和padding的问题，这个是默认的盒子解析模型不同导致的。

IE6：中包括border和padding(box-sizing: border-box;)

IE7和非IE：width宽度不包括border和padding(box-sizing: content-box;)

解决方式： 根据使用方式，写好box-sizing属性。

### 5.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

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。

### 6.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。

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。

### 7.万能清除浮动

[复制代码](javascript:void(0);)

.clearfix:after {

content:".";

display:block;

height:0;

clear:both;

visibility:hidden;

}

[复制代码](javascript:void(0);)

### 8.IE捉迷藏的问题

　　当div应用复杂的时候每个栏中又有一些链接，DIV等这个时候容易发生捉迷藏的问题。 有些内容显示不出来，当鼠标选择这个区域是发现

内容确实在页面。

解决办法：对#layout使用line-height属性或者给#layout使用固定高和宽。页面结构尽量简单。

### 9.溢出显示省略号

[复制代码](javascript:void(0);)

select {

-o-text-overflow:ellipsis;

text-overflow:ellipsis;

white-space:nowrap;

overflow:hidden;

width:200px;

}

[复制代码](javascript:void(0);)

这个是在越出长度后会自行的截掉多出部分的文字，并以省略号结尾，很好的一个技术。只是目前Firefox并不支持。

### 10.IE6下图片下有空隙产生

解决这个BUG的技巧有很多,可以是改变html的排版,或者设置img为display:block。

### 11.超链接访问过后hover样式就不出现的问题

解决技巧

a:link {}

a:visited {}

a:hover {}

a:active {}

# 5、CSS伪类选择器有哪些?（\*\*）

(1) :before 在...之前

(2) :after 在...之后

(3) :active 点击

(4) :hover 鼠标悬停

(5) :first-letter 所有文字当中的第一个

(6) :first-line 所选元素当中的第一行

(7) :first-child 第一个子元素

(8) :nth-child() 获取子元素

(9) :nth-of-type 获取指定类型的子元素

等等几十个

# 6、常用的适配手段有哪些（\*\*）

1.媒体查询（\*\*\*）

2.弹性布局(\*\*\*)

3.使用各种UI框架(比如bootstrap)，UI框架一般都做了适配工作（\*\*）

4.手淘适配方案 （使用flexiable插件完成适配）(\*)

# 7、说说你对flex布局的理解（\*\*）

参考: https://www.cnblogs.com/jpwz/p/12483053.html

Flex 是 Flexible Box 的缩写，意为"灵活的盒子"或"弹性的盒子"，所以 flex 布局一般也叫作"弹性布局"。

- display 属性

- flex-direction 属性

- flex-wrap 属性

- flex-flow

- justify-content 属性

- align-items 属性

- align-content 属性

# 8、rem和em、VH、VW、PX的区别（\*\*）

em是相对于父级标签的字体大小来定义的

rem是相对于html标签的字体大小来定义的

# 9、canvas和svg的区别（\*）

都可以用来绘制图形，但是canvas是基于js来绘制的，svg是基于xml来绘制的.SVG绘制出来的是矢量图。

# 10、什么是BFC？BFC的布局规则是什么？如何创建BFC？BFC应用？

`BFC` 是 `Block Formatting Context `的缩写，即块格式化上下文。`BFC`是CSS布局的一个概念，是一个环境，里面的元素不会影响外面的元素。 布局规则：Box是CSS布局的对象和基本单位，页面是由若干个Box组成的。元素的类型和display属性，决定了这个Box的类型。不同类型的Box会参与不同的`Formatting Context`。

创建：浮动元素 `display：inline-block position:absolute`

应用: 1.分属于不同的`BFC`时,可以防止`margin`重叠 2.清除内部浮动 3.自适应多栏布局

# 12、清除浮动的方式有哪些

(1)浮动元素后新增标签，并且设置属性clear:both ;

(2)浮动元素的父标签设置属性overflow:hidden;

(3)设置伪类选择器 XX:after{clear:both}

# 13、网站页面常见优化的手段

(1)优化图片资源的格式（优先级选择webp > jpeg > png > bmp）

(2)使用压缩文件 min.js、min.css

(3)使用浏览器缓存（cookie ， localStorage）

(4)尽量减少请求次数

(5)使用懒加载，避免一次性请求过多

# 14、 CSS选择器优先级（\*\*\*）

- 内联样式表的权值为 1000

- ID 选择器的权值为 100

- Class 类选择器的权值为 10

- HTML 标签选择器的权值为 1

通过使用加法计算出来权重的大小

# 15、 九宫格实现方式

- flex布局 （css3）

- grid布局 （css3新增）

- float

- table （）

参考地址：<https://www.cnblogs.com/sunshq/p/10118425.html>

# 16、让一个盒子在父元素的中进行左右垂直居中

参考地址： <https://www.cnblogs.com/yingtoumao/p/11541256.html>

1. div{ margin:0 auto;}
2. div{width:500px;float:left; position:relative;left:50%; margin-left:-250px;}
3. div{width:500px; position:absolute;left:50%; margin-left:-250px;}

div{width:500px;position: absolute;left:50%; transform:tranlateX(-50%);}

1. div{ display: flex;align-items: center;justify-content: center;}

# 17、浏览器w3c标准模式和怪异模式有何区别，描述并可举例说明？

答 ：比如div{width:500px; padding-left:30px; padding-right:40px; margin:0 20px; border:10px solid red;}

1. w3c标准模式盒模型: 属于外增式 盒子的大小=width(500)+2padding(30+40) +2border(20+20)

盒子所占的空间=盒子的大小+2margin

(2)怪异模式盒模型 属于内减式 盒子的大小=width(500)

盒子所占的空间=盒子的大小+2margin

# 18、请简述"margin塌陷现象"的原因，并写出解决办法。

答：

（1）当两个盒子是兄弟关系并垂直显示的时候，下面盒子margin-top ,上面盒子margin-bottom,

外边距以设置的最大值为准（外边距合并的第一种情况）

(2)当两个盒子是嵌套关系，如果对里面的盒子设置margin-top,父盒子会同时下移（塌陷的第二种情况）

给父盒子设置边框

给父盒子设置overflow:hidden;

# 19、页面上的定位主要有哪几种，分别是有哪些特点？

答：5种 静态定位 相对定位 绝对定位 固定定位 粘性定位

1. position:static; 静态定位(普通流定位) -------------- 默认定位
2. position:relative;相对定位 相对本元素的左上角进行定位，top,left,bottom,right都可以有值。虽然经过定位后，位置可能会移动，但是本元素并没有脱离文档流，还占有原来的页面空间
3. position:abosolute; 绝对定位，相对于祖代中有relative(相对定位)并且离本元素层级关系上是最近的元素的左上角进行定位，如果在祖代元素中没有有relative定位的，就默认相对于body进行定位，脱离标准流
4. position:fixed; 固定定位是以窗口为参考。当出现滚动条时，对象不会随着滚动
5. position：sticky

# 21 浏览器渲染机制、重绘、重排

## 网页生成过程：

- `HTML`被HTML解析器解析成`DOM` 树

- `css`则被css解析器解析成`CSSOM `树

- 结合`DOM`树和`CSSOM`树，生成一棵渲染树(`Render Tree`)

- 生成布局（`flow`），即将所有渲染树的所有节点进行平面合成

- 将布局绘制（`paint`）在屏幕上

\*\*重排(也称回流):\*\* 当`DOM`的变化影响了元素的几何信息(`DOM`对象的位置和尺寸大小)，浏览器需要重新计算元素的几何属性，将其安放在界面中的正确位置，这个过程叫做重排。 触发：

1. 添加或者删除可见的DOM元素

2. 元素尺寸改变——边距、填充、边框、宽度和高度

\*\*重绘：\*\* 当一个元素的外观发生改变，但没有改变布局,重新把元素外观绘制出来的过程，叫做重绘。 触发：

- 改变元素的`color、background、box-shadow`等属性

## \*\*重排优化建议：\*\*

1. 分离读写操作

2. 样式集中修改

3. 缓存需要修改的`DOM`元素

4. 尽量只修改`position：absolute`或`fixed`元素，对其他元素影响不大

5. 动画开始`GPU`加速，`translate`使用`3D`变化

`transform` 不重绘，不回流 是因为`transform`属于合成属性，对合成属性进行`transition/animate`动画时，将会创建一个合成层。这使得动画元素在一个独立的层中进行渲染。当元素的内容没有发生改变，就没有必要进行重绘。浏览器会通过重新复合来创建动画帧。