

## 11.17 面试题

---

1. 除了audio和video，html5还有那些媒体标签？

在新窗口中打开(O)

2. html5存储类型有哪些区别？

3. html5有哪些新增的表单元素？

4. html5关于布局和定位的标签有哪些？

5. html5废弃了哪些html4标签？

6. 常见的浏览器兼容性问题有哪些？

7. 如果做多屏幕的适配需要注意哪些方面？

8. 媒体查询和响应式布局有什么区别？

9. 响应式布局有哪些标签？

---

### HTML5中的多媒体标签

`<video>` 标记定义一个视频

`<audio>` 标记定义音频内容

`<source>` 标记定义媒体资源

`<canvas>` 标记定义图片

`<embed>` 标记定义外部的可交互的内容或插件 比如：flash

---

### HTML5有哪些不同类型的存储？

HTML5支持本地存储，在之前版本中是通过Cookie实现的。

HTML5本地存储速度快而且安全。

有两种不同的对象可用来存储数据：

`localStorage` 适用于长期存储的数据，浏览器关闭后数据不会丢失。

`sessionStorage` 存储的数据在浏览器关闭后自动删除。

---

# HTML5中新增表单元素与属性

## 1. form属性

在HTML5中，可以把表单从属元素写在页面任何位置，然后给元素指定一个form属性，属性值为表单的id，这样就可以声明该元素的从属于指定表单。

```
<form id="text">
  <input type="text">
</form>
<textarea id="text"></textarea>
```

## 2. formaction属性

在HTML5中，可以给所有的提交按钮增加不同的formaction属性，点击不同的按钮，将不同的按钮提交到不同的页面。

```
<input type="submit" formaction="a.php">
<input type="submit" formaction="b.php">
```

## 3. formmethod属性

在HTML5中，可以使用formaction属性来对每个表单元素分别指定不同的提交页面，同时也可以使用formmethod属性来对每个表单元素分别指定不同的提交方法。

```
<input type="submit" formaction="a.php" formmethod="post">
<input type="submit" formaction="b.php" formmethod="get">
```

## 4. placeholder属性

placeholder是指文本框或处于为输入状态并且未获得光标焦时，降低显示输入提示文字不透明度。

```
<input type="text" placeholder="点击收索">
```

## 5. autofocus属性

给文本框、选择框或者按钮控件自动获得光标焦点。

```
<input type="text" autofocus>
```

## 6. list属性

在HTML5中，为单行文本增加一个list属性，该属性的值为某个datalist元素的id。datalist元素类似于选择框（select），该元素本身并不显示，而是文本框获得焦点时以提示输入的方式显示。

```
<input type="text" list="w3cmm">
<datalist id="w3cmm" style="display:none">
  <option>HTML</option>
  <option>CSS</option>
  <option>JavaScript</option>
</datalist>
```

为了避免在不支持该元素的浏览器中的错误显示，用CSS等设定它为不显示。

## 7. autocomplete属性

辅助输入所用的自动完成功能，节省输入时间，同时也十分方便的功能。对于autocomplete属性，可以指定“on”、“off”、“（不指定）”这三种值。

```
<input type="text" autocomplete="on" list="w3cmm">
```

---

## html5中关于布局的标签

HTML5中推出了很多专门用于布局的新标签。

- `<header>` 标签  
`<header>` 标签定义文档的页眉（介绍信息），用法：

```
<header>
  <h1>页眉</h1>
</header>
```

- `<nav>` 标签  
`<nav>` 标签定义导航链接的部分，用法：

```
<nav>
  <ul>
    <li><a href="#">菜单1</a></li>
    <li><a href="#">菜单2</a></li>
    <li><a href="#">菜单3</a></li>
  </ul>
</nav>
```

- **<section> 标签**

**<section>** 标签中定义文档中的节（section、区段）。比如章节、页眉、页脚或文档中的其他部分。用法：

```
<section>
  <h1>标题1</h1>
  <p>文章段落内容</p>
</section>
```

- **<article> 标签**

**<article>** 标签定义外部内容。

外部内容可以是来自一个外部的新闻提供者的一篇新的文章，或者来自blog的文本，或者来自其他地方的文本。亦或是来自其他外部源内容。

总之可以理解为article所表示的就是文章内容。

用法：

```
<article>
  <a href="#">文章内容链接</a>
  <br>
  <p>文章内容文章内容文章内容</p>
</article>
```

- **<aside> 标签**

**<aside>** 标签定义article以外的内容。aside的内容应该与article的内容相关。它类似于布局中的辅助侧边栏。用法：

```
<aside>
  <h4>标题</h4>
  <p>段落内容段落内容</p>
</aside>
```

- **<footer> 标签**

**<footer>** 标签定义section或者document的页脚。在典型的情况下，该元素会包含创

作者的姓名、文档的创作日期/联系信息，也就是类似于版权信息的东西。

用法：

```
<footer>页脚文字，版权信息等</footer>
```

## html5中废除的html4标签

### 1. 能用CSS代替的标签

basefont、big、center、font、s、strick、tt、u.....这些元素纯粹是为了画面展示服务的，HTML5中提倡把画面展示性功能放在CSS中统一实现。

### 2. 不能再使用frame框架

frameset、frame、noframes。

HTML5中不支持frame框架，只支持iframe框架，或者用服务器方创建的由多个页面组成的符合页面的形式，删除以上这三个标签。

### 3. 只有浏览器支持的元素

applet、bgsound、blink、marquee等标签

### 4. 其他被废除的元素

1. 废除rb，使用ruby代替；
2. 废除acronym，使用abbr替代；
3. 废除dir，使用ul替代；
4. 废除isindex，使用form与input相结合的方式替代；
5. 废除listing，使用pre替代；
6. 废除xmp，使用code替代；
7. 废除呢些tid，使用guids替代；
8. 废除plaintext使用“text/plian”(无格式正文)MIME类型替代。

## 常见的浏览器兼容性问题

- 不同浏览器的默认的外间距和内间距不同。
- 块属性标签float后，又有横行的margin情况下，在IE6中显示的margin比设置的大
- 设置较小高度标签（一般小于10px），在IE6、IE7等浏览器中高度超出自己设置的高度。
- 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，IE间距bug。
- 图片默认有间距。
- 标签最低高度设置 `min-height` 不兼容。
- 透明度的兼容CSS设置。

## 多屏幕的适配需要注意哪些方面？

- 在manifest里定义你的程序支持的屏幕类型。
  - 对不同大小的屏幕提供不同 `layout` 。
  - 对不同的屏幕提供不同的图片。
  - 多屏幕适配的4条黄金原则
    - 在layout文件中设置空间尺寸是应采用wrap\_content , fill\_parent和dp。
    - 在程序代码中不要出现具体的像素值。
    - 不要使用AbsoluteLayout ( android1.5已废弃 ) 。相应的应该使用RelativeLayout。
    - 对不同的屏幕提供合适大小的图片。
- 

## 响应式布局和移动端布局的区别？

响应式布局是为了适配不同的终端而生，移动端布局是针对智能手机为主，也就是H5站（m站）。

简单说一下它们的不同之处：

1. 适配的群体不同，响应式适配各种终端，而移动端大部分以智能手机为主。
2. 响应式布局能根据不同的终端设备实现不同的页面布局，而移动端布局大部分是单列布局。
3. 响应式布局有可能造成冗余的代码比较多（传统的响应式布局，仅依赖于媒体查询，控制不同的页面布局），移动端布局冗余代码较少，针对性较强。

相同之处：

1. 在智能手机上，响应式看到的页面效果能和移动端一样效果（其他忽略，仅从效果而言）。
2. 两者都要面对适配的问题。

可以理解为：响应式布局 = PC布局 + 移动端布局