12个HTML和css必须知道的重点难点问题

这12个问题，基本上就是HTML和CSS基础中的重点个难点了，也是必须要弄清楚的基本问题，其中定位的绝对定位和相对定位到底相对什么定位？这个还是容易被忽视的，浮动也是一个大坑，有很多细节。

这12个知识点是我个人认为的，下面我们就来看看这12个知识点。

**1.怎么让一个不定宽高的 DIV，垂直水平居中?**

使用Flex只需要在父盒子设置：

1. display: flex;
2. justify-content: center;
3. align-items: center;

使用 CSS3 transform

1. .parent{
2. position:relative
3. }
4. .children {
5. position: absolute;
6. top: 50%;
7. left: 50%;
8. transform: translate(-50%，-50%);
9. }

使用 display:table-cell 方法

1. .parent{
2. display:table-cell;
3. text-align:center;
4. vertical-align:middle;
5. }
6. .children {
7. display:inline-block;
8. vertical-align:middle;
9. }

**2. position 几个属性的作用**

position 的常见四个属性值: relative，absolute，fixed，static。一般都要配合"left"、"top"、"right" 以及 "bottom" 属性使用。

* **static:默认位置。**在一般情况下，我们不需要特别的去声明它，但有时候遇到继承的情况，我们不愿意见到元素所继承的属性影响本身，从而可以用Position:static取消继承，即还原元素定位的默认值。设置为 static 的元素，它始终会处于页面流给予的位置(static 元素会忽略任何 top、 bottom、left 或 right 声明)。一般不常用。
* **relative:相对定位。**相对定位是相对于元素默认的位置的定位，它偏移的 top，right，bottom，left 的值都以它原来的位置为基准偏移，而不管其他元素会怎么 样。注意 relative 移动后的元素在原来的位置仍占据空间。
* **absolute:绝对定位。**设置为 absolute 的元素，如果它的 父容器设置了 position 属性，并且 position 的属性值为 absolute 或者 relative，那么就会依据父容器进行偏移。如果其父容器没有设置 position 属性，那么偏移是以 body 为依据。注意设置 absolute 属性的元素在标准流中不占位置。
* **fixed:固定定位。**位置被设置为 fixed 的元素，可定位于相对于浏览器窗口的指定坐标。不论窗口滚动与否，元素都会留在那个位置。它始终是以 body 为依据的。 注意设置 fixed 属性的元素在标准流中不占位置。

**3.浮动与清除浮动**

3.1 浮动相关知识

**float属性的取值：**

* left：元素向左浮动。
* right：元素向右浮动。
* none：默认值。元素不浮动，并会显示在其在文本中出现的位置。

**浮动的特性：**

* 浮动元素会从普通文档流中脱离，但浮动元素影响的不仅是自己，它会影响周围的元素对齐进行环绕。
* 不管一个元素是行内元素还是块级元素，如果被设置了浮动，那浮动元素会生成一个块级框，可以设置它的width和height，因此float常常用于制作横向配列的菜单，可以设置大小并且横向排列。

**浮动元素的展示在不同情况下会有不同的规则：**

* 浮动元素在浮动的时候，其margin不会超过包含块的padding。PS：如果想要元素超出，可以设置margin属性
* 如果两个元素一个向左浮动，一个向右浮动，左浮动元素的marginRight不会和右浮动元素的marginLeft相邻。
* 如果有多个浮动元素，浮动元素会按顺序排下来而不会发生重叠的现象。
* 如果有多个浮动元素，后面的元素高度不会超过前面的元素，并且不会超过包含块。
* 如果有非浮动元素和浮动元素同时存在，并且非浮动元素在前，则浮动元素不会高于非浮动元素
* 浮动元素会尽可能地向顶端对齐、向左或向右对齐

**重叠问题**

* 行内元素与浮动元素发生重叠，其边框，背景和内容都会显示在浮动元素之上
* 块级元素与浮动元素发生重叠时，边框和背景会显示在浮动元素之下，内容会显示在浮动元素之上

**clear属性** clear属性：确保当前元素的左右两侧不会有浮动元素。clear只对元素本身的布局起作用。 取值：left、right、both

3.2 父元素高度塌陷问题

**为什么要清除浮动，父元素高度塌陷** 解决父元素高度塌陷问题：一个块级元素如果没有设置height，其height是由子元素撑开的。对子元素使用了浮动之后，子元素会脱离标准文档流，也就是说，父级元素中没有内容可以撑开其高度，这样父级元素的height就会被忽略，这就是所谓的高度塌陷。

3.3 清除浮动的方法

**方法1：给父级div定义 高度** 原理：给父级DIV定义固定高度（height），能解决父级DIV 无法获取高度得问题。 优点：代码简洁 缺点：高度被固定死了，是适合内容固定不变的模块。（不推荐使用）

**方法二：使用空元素，如**

**(.clear{clear:both})** 原理：添加一对空的DIV标签，利用css的clear:both属性清除浮动，让父级DIV能够获取高度。 优点：浏览器支持好 缺点：多出了很多空的DIV标签，如果页面中浮动模块多的话，就会出现很多的空置DIV了，这样感觉视乎不是太令人满意。（不推荐使用）

**方法三：让父级div 也一并浮起来** 这样做可以初步解决当前的浮动问题。但是也让父级浮动起来了，又会产生新的浮动问题。 不推荐使用

**方法四：父级div定义 display:table** 原理：将div属性强制变成表格 优点：不解 缺点：会产生新的未知问题。（不推荐使用）

**方法五：父元素设置 overflow：hidden、auto；** 原理：这个方法的关键在于触发了BFC。在IE6中还需要触发 hasLayout（zoom：1） 优点：代码简介，不存在结构和语义化问题 缺点：无法显示需要溢出的元素（亦不太推荐使用）

**方法六：父级div定义 伪类:after 和 zoom**

1. .clearfix:after{
2. content:'.';
3. display:block;
4. height:0;
5. clear:both;
6. visibility: hidden;
7. }
8. .clearfix {zoom:1;}

原理：IE8以上和非IE浏览器才支持 :after，原理和方法2有点类似， zoom(IE转有属性)可解决ie6,ie7浮动问题 优点：结构和语义化完全正确,代码量也适中，可重复利用率（建议定义公共类） 缺点：代码不是非常简洁（极力推荐使用）

**经益求精写法**

1. .clearfix:after {
2. content:”\200B”;
3. display:block;
4. height:0;
5. clear:both;
6. }
7. .clearfix { \*zoom:1; } 照顾IE6，IE7就可以了

详细关于浮动的知识请参看这篇文章： http://luopq.com/2015/11/08/CSS-float/

**4. BFC相关知识**

**定义**：BFC(Block formatting context)直译为"块级格式化上下文"。它是一个独立的渲染区域，只有 Block-level box 参 与， 它规定了内部的 Block-level Box 如何布局，并且与这个区域外部毫不相干。

**BFC布局规则** BFC 就是页面上的一个隔离的独立容器，容器里面的子元素不会影响到外面的元素。反之也如此。

* BFC这个元素的垂直方向的边距会发生重叠，垂直方向的距离由 margin决定，取最大值
* BFC 的区域不会与浮动盒子重叠（ 清除浮动原理）。
* 计算 BFC 的高度时，浮动元素也参与计算。

**哪些元素会生成 BFC**

* 根元素
* float 属性不为 none
* position 为 absolute 或 fixed
* display 为 inline-block， table-cell， table-caption， flex， inline-flex
* overflow 不为 visible

**5. box-sizing是什么**

设置CSS盒模型为标准模型或IE模型。标准模型的宽度只包括 content，二IE模型包括 border和 paddingbox-sizing属性可以为三个值之一：

* content-box，默认值，只计算内容的宽度，border和padding不计算入width之内
* padding-box，padding计算入宽度内
* border-box，border和padding计算入宽度之内

**6. px，em，rem 的区别**

**px** 像素(Pixel)。绝对单位。像素 px 是 相对于显示器屏幕分辨率而言的，是一个虚拟长度单位，是计算 机系统的数字化图像长度单位，如果 px 要换算成物理长度，需要指定精度 DPI。 **em** 是相对长度单位， 相对于当前对象内文本的字体尺寸。如当前对行内文本的字体尺寸未被人为设置， 则相对于浏览器的默认字体尺寸。它会继承父级元素的字体大小，因此并不是一个固定的值。 **rem** 是 CSS3 新增的一个相对单位(root em，根 em)，使用 rem 为元素设定字体大小时，仍然是相对大小， 但 相对的只是HTML根元素。

**7. CSS 引入的方式有哪些? link 和@import 的区别是?**

有四种：内联(元素上的style属性)、内嵌(style标签)、外链(link)、导入(@import) link和@import的区别：

* link是XHTML标签，除了加载CSS外，还可以定义RSS等其他事务； @import属于CSS范畴， 只能加载CSS。
* link引用CSS时，在 页面载入时同时加载； @import需要页面网页完全载入以后加载。
* link是XHTML标签， 无兼容问题； @import是在CSS2.1提出的， 低版本的浏览器不支持。
* link支持使用Javascript控制DOM去 改变样式；而 @import不支持。

**8. 流式布局与响应式布局的区别**

**流式布局** 使用非固定像素来定义网页内容， 也就是百分比布局，通过盒子的宽度设置成百分比来根据屏幕的宽度来进 行伸缩，不受固定像素的限制，内容向两侧填充。

**响应式开发** 利用CSS3 中的 Media Query(媒介查询)，通过查询 screen 的宽度来指定某个宽度区间的网页布局。

* 超小屏幕(移动设备) 768px 以下
* 小屏设备 768px-992px
* 中等屏幕 992px-1200px
* 宽屏设备 1200px 以上

由于响应式开发显得繁琐些，一般使用第三方响应式框架来完成，比如 bootstrap 来完成一部分工作，当然也 可以自己写响应式。

**区别**

| **-** | **流式布局** | **响应式开发** |
| --- | --- | --- |
| 开发方式 | 移动Web开发+PC开发 | 响应式开发 |
| 应用场景 | 一般在已经有PC端网站，开发移动的的时候只需要单独开发移动端 | 针对一些新建的网站，现在要求适配移动端，所以就一套页面兼容各种终端 |
| 开发 | 正对性强，开发效率高 | 兼容各种终端，效率低 |
| 适配 | 只适配移动设备，pad上体验相对较差 | 可以适配各种终端 |
| 效率 | 代码简洁，加载快 | 代码相对复杂，加载慢 |

**9. 渐进增强和优雅降级**

关键的区别是他们所侧重的内容，以及这种不同造成的工作流程的差异

* **优雅降级**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。。
* **渐进增强**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。

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
* 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

**10.CSS隐藏元素的几种方式及区别**

**display:none**

* 元素在页面上将彻底消失，元素本来占有的空间就会被其他元素占有，也就是说它会导致浏览器的重排和重绘。
* 不会触发其点击事件

**visibility:hidden**

* 和 display:none的区别在于， 元素在页面消失后，其占据的空间依旧会保留着，所以它 只会导致浏览器重绘而不会重排。
* 无法触发其点击事件
* 适用于那些元素隐藏后不希望页面布局会发生变化的场景

**opacity:0**

* 将元素的透明度设置为0后，在我们用户眼中，元素也是隐藏的，这算是一种隐藏元素的方法。
* 和 visibility:hidden的一个共同点是元素隐藏后依旧占据着空间，但我们都知道，设置透明度为0后，元素只是隐身了，它依旧存在页面中。
* 可以触发点击事件

**设置height，width等盒模型属性为0**

* 简单说就是将元素的 margin， border， padding， height和 width等影响元素盒模型的属性设置成 0，如果元素内有子元素或内容，还应该设置其 overflow:hidden来隐藏其子元素，这算是一种奇技淫巧。
* 如果元素设置了border，padding等属性不为0，很显然，页面上还是能看到这个元素的，触发元素的点击事件完全没有问题。如果全部属性都设置为0，很显然，这个元素相当于消失了，即无法触发点击事件。
* 这种方式既不实用，也可能存在着着一些问题。但平时我们用到的一些页面效果可能就是采用这种方式来完成的，比如jquery的slideUp动画，它就是设置元素的overflow:hidden后，接着通过定时器，不断地设置元素的height，margin-top，margin-bottom，border-top，border-bottom，padding-top，padding-bottom为0，从而达到slideUp的效果。

**其他脑洞方法**

* 设置元素的position与left，top，bottom，right等，将元素移出至屏幕外
* 设置元素的position与z-index，将z-index设置成尽量小的负数

**11.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。

**12. 行内元素有哪些？块级元素有哪些？ 空(void)元素有那些？**

这个问题面试偶尔被问到的时候有点懵逼~~~平时没在意。。。。

行内元素：a、b、span、img、input、strong、select、label、em、button、textarea 块级元素：div、ul、li、dl、dt、dd、p、h1-h6、blockquote 空元素：即系没有内容的HTML元素，例如：br、meta、hr、link、input、img

(完)