# DAY07 CSS

# 背景

### 7. 背景图的尺寸

|  |
| --- |
| background-size 没有设置任何尺寸时，默认使用图片本身的大小  background-size: 600px 300px; 取值可以是像素值  background-size: 100% 100%; 取值可以是百分比，宽高都是100%注意图片的失真问题  background-size: 100% auto; 宽度正好,高度自适应,会牺牲高度  background-size: auto 100%; 高度正好,宽度自适应,会牺牲宽度  background-size: contain; 至少显示一张完整的图,多余的空间平铺  background-size: cover; 铺满整个屏幕,多余的部分裁掉  注意：能使用比例合适的图尽量使用比例合适的图 |

# 1 版心

|  |
| --- |
| **版心：**  在页面布局中，指定一个确定宽度的，用来显示主体内容的区域  注意：PC端页面一定要写版心，版心宽度是设计提供的 |
| **常见的PC端分辨率:**  960px 1024px 1200px  PC端相对来说，内容较多，样式较复杂，需要设置一个固定的单位显示内容区域 |

# 2 元素的隐藏与显示

|  |
| --- |
| **方案1:修改颜色为透明色**  background-color: transparent;  background-color: rgba(0,0,0,0); |
| **方案2:修改元素的显示模式**  display:none; none表示不显示且释放布局空间  如果要显示回来就使用元素原有的display属性值，比如display:block; |
| **方案3:设置可见度为隐藏**  visibility: hidden; 隐藏  visibility: visible; 默认值，显示 |
| **方案4: 修改元素的不透明度**  opacity：0.3； 0完全透明 1完全不透明  注意:这里的不透明度包含元素与元素中的所有内容 |

# 3 元素浮动

### 为什么要用浮动

|  |
| --- |
| 布局中遇到的第一个困难就是块级元素的横向排列  行内元素虽然是横向排列，但设置宽高无效，且不能包裹更多标签  行内块元素虽然也是横向排列，但不建议做布局，因为行内块自带了很多不好去除的样式  所以，我们要使用浮动，可以让元素横向排列，方便排版 |

### float属性

|  |
| --- |
| float单词的含义是“浮动”，它最初的作用是为了制作“文本环绕效果”  float:left; 左浮动，是向父元素的左侧边界靠拢  float:right; 右浮动，是向父元素的右侧边界靠拢  注意：如果多个元素朝一个方向浮动，谁先浮动，谁就最靠近边界 |

### 元素浮动后产生的效果

|  |
| --- |
| 1. 一个父元素中的子元素浮动了，那么这个子元素就不再支撑父元素的高度 2. 浮动会破坏原有的文档流布局，变成浮动布局，元素可以横向排列 3. 如果一个父元素中所有的子元素都浮动了，父元素会高度坍塌/失高 4. 浮动会让元素”块状化”【行内 行内块会变成块级】 5. 子元素浮动后，外间距的重叠现象消失   兄弟间的垂直外间距相加显示 父子的贴边问题也会消失 |

# 解决高度坍塌

|  |
| --- |
| 当父元素中的所有子元素都浮动走，也就是所有的子元素都脱离文档流，释放布局空间后，不再支撑父元素的高度，父元素就会高度坍塌 |

### 父元素自定义高度

|  |
| --- |
| 父元素可以通过参考子元素的高度给自己设置一个高度 |
| **应用场景：**  已知子元素的高度，并且子元素高度相同时，适合此方案，比如:导航条  注意:当子元素高度未知时，无法自定义父级高度 |

### 给父元素设置BFC保护

|  |
| --- |
| 任何设置了BFC的元素，和浮动的元素相遇时，都可以实现”自动填充”的适应布局  也就是说，当父元素设置了BFC结界后，子元素浮动浮动，父元素会根据子元素实际浮动后的高度自动填充自己的高度 |
| **应用场景:**  子元素未知高度，且子元素不会超出父级范围  会触发BFC的代码:overflow:hidden/auto/scroll; |

### 父级元素称为同层元素

|  |
| --- |
| 父级想要和浮动的子元素称为同一层元素，相当于父元素也要脱离文档流  父元素浮动脱离文档流，就可能会影响到父元素的父元素，导致上层元素依然需要解决高度坍塌问题 |

### clear属性清除浮动影响

|  |
| --- |
| clear属性是专门用于解决float带来的高度坍塌问题   1. 清除指的是清除前面“哥哥们”浮动对父元素产生的不支撑效果 2. clear会让前面的浮动元素与当前元素换行显示，对后面的浮动元素无法起到作用 3. 使用clear属性的元素，自己不可以浮动，要保持在文档流中替前面的浮动元素清除影响 4. clear属性只对块级元素生效 |
| **使用方法：**  在父元素的最后添加一个宽高都为0的块级子元素，清除浮动影响  但这样的方式会造成后续在父元素下多一个孩子 |

### 使用伪元素

|  |
| --- |
| 给父元素最后生成一个空白的“假孩子”，变块级，清影响  .clearfix::after{ /\* 在指定元素的最后生成一个假孩子 \*/  content: ''; /\* 内容是空白的,宽高为0 \*/  display: block; /\* 块级元素才可以使用clear属性 \*/  clear: both; /\* 清除前面所有元素浮动造成的不支撑影响 \*/  }  哪个元素高度坍塌，就可以给哪个元素上加类名clearfix |
| 我们使用::after模拟出来的假孩子，在父元素的最后，很方便的帮我们清除高度坍塌影响 |

# 拓展

### 幽灵空白节点

|  |
| --- |
| 内联元素(行内+行内块)自带一个叫“幽灵空白节点”的文字节点  如果需要去掉幽灵空白节点，可以把该内联元素变为块级元素  注意：幽灵空白节点不是必须要去除的，看实际需求有没有必要 |

### BFC

|  |
| --- |
| 【面试题】  BFC的全称是Block Formatting Context(块级格式化上下文)  简单的来说，它是一个独立的渲染区域，这个区域和外部不相干了，相当于子元素在父元素内部折腾，但也不会超出父元素的范围  触发条件: 使用overflow的三个属性值auto/hidden/scroll都可以让父元素形成BFC结界  BFC结界一旦形成，就能让父元素重新计算内部浮动的子元素的实际高度，进行自动填充  BFC结界一旦形成，就能避免margin的父子贴边问题 |