Day10笔记

作业！

看视频号中的浮动布局！！！！！！！！！！

文本, 白板

描述已自动生成2. 浮动

- 子元素在父元素中浮动，会使其脱离文档流，

导致父元素对其的管理能力

- 浮动属性

- `float：none`默认不浮动

- `float：left`左浮动

- `float：right`右浮动

(1)浮动过程：

- 脱离文档流不再撑起父元素

- 不再自占一行

- 在元素浮动的“当前行”向左或向右对齐

(2)元素浮动特点

1.元素浮动后对父级的高度有影响，不再撑起父级的高度

2.元素浮动后可以在父元素的左侧或右侧排列

3.元素浮动后不自占一行

4.元素浮动后会对后面的文档流中的兄弟元素产生影响会遮盖其后面的兄弟元素

5.内联元素浮动后，自动变成块级元素

6.浮动元素仍受父元素影响，还是在父元素的范围内

7.当父元素内所有元素均浮动，元素们宽度相加大于父元素宽度时，会换行

8.当元素浮动时虽然不自占一行，但会向前占位，前方空值的位置都会属于自己所有，因此后方的元素换行时，不能占据元素向前占位的地域（见元素占位示意图）

9.当元素浮动时，原则会盖住其后方在文档流中的兄弟元素。但文字、图片等（行内元素和行内块）不会被盖住，其后兄弟元素的文字不会被盖住而是环绕浮动元素四周显示。解决方案包裹文字的元素浮动在前一个元素侧面。（见文字环绕示意图）

图表, 瀑布图

描述已自动生成

文本

描述已自动生成

3. 清除浮动

- 清除浮动并不是去除浮动，而是将浮动带来的负面效果解决掉。因为浮动会产生高度坍塌

- 浮动的影响父元素无高度

- 父元素无高度后，父元素的兄弟元素会向上，发生被遮盖

（1）利用高度解决

给父元素高度，弊端是未必知道子元素的高度

（2）父元素也浮动

父元素浮动，会影响父元素的兄弟元素

（3）父元素溢出隐藏

overflow：hidden/auto，父元素中的子元素可能会被隐藏（利用BFC 块级格式化上下文中，使用该属性可保证浮动元素也可以被计算入父级的高度）

（4）用伪元和clear属性素解决

给父元素增加一个在尾部的伪元素

|  |
| --- |
| .clearfix:after {  content: "";  display: block;  clear: both;  }  .clearfix {  \*zoom: 1;  } /\*IE/7/6\*/ |

【练习】

日程表

描述已自动生成

4. 浮动流体布局

PC 端页面会有一个内容部分的固定宽度（例如：1200px、1190px、1004px、960px 等等）。

液晶还没普及的年代，网页设计需要考虑 800×600 的分辨率。之后，液晶屏开始普及，1024×768，使用率接近 50%(国内)，所以很多网站都是以 1024 的宽度为标准进行设计的。

现在的液晶屏大小已经变得多样化，4k 时代到来，高分辨率，让 pc 端页面布局的宽度设计变图表, 树状图

描述已自动生成得更加要复合用户群。

(1) 分析 pc 端页面布局特点

pc 端布局都存在一个用于适应最小屏幕显示的居中元素。一般会把内容的结构都放在这个居中的元素中，左右两侧留白。

图形用户界面, 应用程序, 网站

描述已自动生成因此在 pc 端布局中，可以在用于划分某个部分的元素内部加入一个用于居中的元素。如：

(2) CSS reset

HTML标签在浏览器中都有默认的样式，不同的浏览器的默认样式之间存在差别。开发时浏览器的默认样式可能会给我们带来多浏览器兼容性问题，影响开发效率。所以解决的方法就是一开始就将浏览器的默认样式全部去掉，更准确说就是通过重新定义标签样式。这就是CSS reset。

|  |
| --- |
| /\* 星号的效率过低，非练习时可以自定义某些元素去掉margin: 0;padding: 0; \*/  \* {  margin: 0;  padding: 0;  }  a {  text-decoration: none;  }  ul {  list-style: none;  }  /\* 清除浮动 \*/  .clearFloat:after {  content: "";  display: block;  clear: both;  }  .clearFloat {  \*zoom: 1;  }  /\* 如：针对学子商城居中1000宽度 \*/  .center {  width: 1000px;  margin: 0 auto;  } |

十八、定位

1.position 属性

- `position`属性是定位属性，用于指定一个元素在文档中的定位方式。

- top，right，bottom 和 left 属性值则决定了该元素的距离四边的位置，可以为负值。

- 开发中多用绝对长度单位 px 进行调整，如果在移动端中，可以使用 rem、vw 等单位进行调整。

- 常用取值：

- `relative` 相对定位

- `absolute` 绝对定位

- `fixed` 固定定位

2.相对定位

- 相对定位 position: relative 不脱离文档流。

- 可使用 top，right，bottom 和 left 做偏移。

- 元素相对的位置是自己本来的位置，移动不改变页面布局。

- 相对定位属性不会影响周围的布局，但会出现新的层叠顺序。

- 当四个方向值发生重合时，以 top 和 left 为优先。

图示

中度可信度描述已自动生成

3.绝对定位

- absolute 绝对定位，元素将脱离文档流，其他元素不为该元素预留空间。

- 它的移动参照为祖先元素的偏移，来确定元素位置。元素会逐层向上寻找自己的参照元素，当找到的最近一层祖先元素具有 position 属性时，该元素就会以这个祖先元素的原点为参照点。

- 可使用 top，right，bottom 和 left 做偏移。

- 当四个方向值发生重合时，以“上 top”和“左 left”为优先。

- 绝对定位的元素可以设置外边距，且不会与其他边距合并。

- 应用场景背景图中的文字和按钮，轮播图中的控制符

文本

描述已自动生成