思路

第一个问题：

不知道怎么开始写?

第二个问题：

不知道用什么标签?

第三个问题：

不知道用什么属性?

如果不清楚块元素行内元素，块状标签，内联标签都有什么?

看下面：

CSS display 属性

属性值

1.内联（行内）元素

2.块级（块状）元素

3.行内块元素

4.同一类型的元素，到底该用哪个。

其实同一类型的元素，属性是一样的，显示出的效果也一样，用哪个都可以。可是为了代码的可读性，规范化...

块级元素和行内元素的分类

居中问题

1.行内元素水平垂直居中的方法

1.1 水平居中

1.2 垂直居中

1.2.1单行文本的垂直居中

1.2.2 多行文本的垂直居中

1.3 小结

2.2 绝对定位配合margin的方式实现水平垂直居中

2.3 绝对定位配合margin的方式实现水平垂直居中

2.5 小结

前端这些页面的写法 一开始就分为三个问题

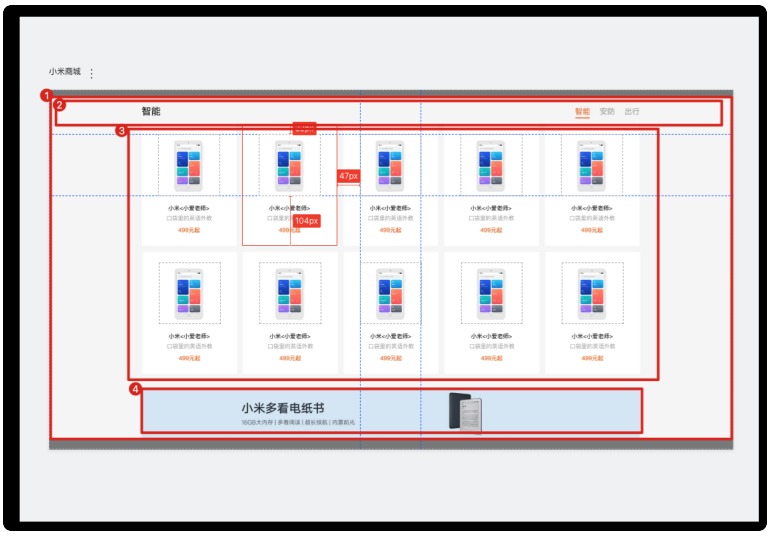
1. 不知道怎么开始写?
2. 不知道用什么标签?
3. 不知道用什么属性?

首先不知道怎么开始写跟css没关系 css控制的是属性 怎么开始写和用什么标签这些都是html的知识

第一个问题:

不知道怎么开始写?

你遇到的其余的所有的页面一开始本质是一样的 先划分框架 从大到小从左到右从上到下 不管是小页面还是大页面都是如此 拿下面这个举例



- 先划分区域 我这里按照从大到小从左到右从上到下
- 分了4个 然后每个里面单独你再分 现在你的1里面有234 然后再看2

就这样依次往下 html里对应的结构也是如此 终究页面都是一个一个盒子组合而成的

第二个问题:

不知道用什么标签?

- 标签问题 你把刚才的1分析完成之后 首先最外层的盒子不知道用什么的时候就用div 其余的比如有列表特征的排的很整齐的用列表标签ul li 是专门的图片的就图片标签 文字的话有很多p标签span标签 都可以超链接就a标签等等这些就需要什么用什么就可以

第三个问题:

不知道用什么属性?

- 属性问题 就需要结合css 前面html标签那些你相当于把一个房子的框架盖好了，然后你要开始装修这个房子，就和装修一样 你的选择有很多，同样css里也有很多很多属性，在你开始写一个标签的属性之前，你需要先了解有的标签是自带属性的，比如ul li它就会让文字竖着排列且有小黑点，那你用它的时候你就需要考虑这个标签自身的属性会对页面或者对你的需求产生什么样的影响，如果这个默认属性有影响 ，那就要去掉，去掉之后呢比如你的需求是几个元素横着排列的，但是你用完ul li之后它自己的是竖着排列的，这个时候就需要再思考 能让元素横着排列的属性有哪些 ，结合着去用，所以 属性的问题就是根据需求那你觉得需要用什么属性就用什么属性，如果不对就再改掉就好，这里主要的标签属性需要记住的就是块元素属性可以设置宽高但是会独占一行 行内元素特点不能设置宽高但是可以一排排很多直到放不下，因为页面里基本上都是有块元素和行内元素组成的 你需要认识到自己设置的标签 （块元素或者行内元素）它自身的特点 再添加其余所需的

如果不清楚块元素行内元素，块状标签，内联标签都有什么?

看下面:

HTML可以将元素分类方式分为内联（行内）元素、块级（块状）元素和行内块元素三种。

注：HTML是标签语言，那么既然是标签，就可以自己定义一些自己元素（如<wode>自定义的元素</wode>等），自定义元素浏览器默认解析为内联元素，为防止不同浏览器解析不同的问题，建议通过css的display属性来规定自定义元素的属性。相信使用过UI框架的朋友，会看到一些html手册以外的一些元素，其实这些都是一些自定义的元素。

首先需要说明的是，这三者是可以互相转换的，使用css的display属性能够将三者任意转换:

CSS display 属性

属性值

值	描述
none	此元素不会被显示。
block	此元素将显示为块级元素，此元素前后会带有换行符。
inline	默认。此元素会被显示为内联元素，元素前后没有换行符。
inline-block	行内块元素。（CSS2.1 新增的值）
list-item	此元素会作为列表显示。
run-in	此元素会根据上下文作为块级元素或内联元素显示。
compact	CSS 中有值 compact，不过由于缺乏广泛支持，已经从 CSS2.1 中删除。
marker	CSS 中有值 marker，不过由于缺乏广泛支持，已经从 CSS2.1 中删除。
table	此元素会作为块级表格来显示（类似 <table>），表格前后带有换行符。
inline-table	此元素会作为内联表格来显示（类似 <table>），表格前后没有换行符。
table-row-group	此元素会作为一个或多个行的分组来显示（类似 <tbody>）。
table-header-group	此元素会作为一个或多个行的分组来显示（类似 <thead>）。
table-footer-group	此元素会作为一个或多个行的分组来显示（类似 <tfoot>）。
table-row	此元素会作为一个表格行显示（类似 <tr>）。
table-column-group	此元素会作为一个或多个列的分组来显示（类似 <colgroup>）。
table-column	此元素会作为一个单元格列显示（类似 <col>）

table-cell	此元素会作为一个表格单元格显示（类似 <td> 和 <th>）
table-caption	此元素会作为一个表格标题显示（类似 <caption>）
inherit	规定应该从父元素继承 display 属性的值。

HTML | 复制代码

```html
<html>
<head>
<style type="text/css">
p {
  display: inline
}
div {
  display: none
}
</style>
</head>

<body>
<p>本例中的样式表把段落元素设置为内联元素。</p>

<p>而 div 元素不会显示出来！</p>

<div>div 元素的内容不会显示出来！</div>
</body>
</html>
```

显示结果：本例中的样式表把段落元素设置为内联元素。 而 div 元素不会显示出来！

1.内联（行内）元素

行内元素最常使用的就是span，其他的只在特定功能下使用，修饰字体<b>和<i>标签，还有<sub>和<sup>这两个标签可以直接做出平方的效果，而不需要类似移动属性的帮助，很实用。

行内元素特征：(1)设置宽高无效

(2)对margin仅设置左右方向有效，上下无效；padding设置上下左右都有效，即会撑大空间

(3)不会自动进行换行

2.块级（块状）元素

块状元素代表性的就是div，其他如p、nav、aside、header、footer、section、article、ul-li、address等等，都可以用div来实现。不过为了可以方便程序员解读代码，一般都会使用特定的语义化标签，使得代码可读性强，且便于查错。

块状元素特征：(1)能够识别宽高

(2)margin和padding的上下左右均对其有效

(3)可以自动换行

(4)多个块状元素标签写在一起，默认排列方式为从上至下

3.行内块元素

行内块状元素综合了行内元素和块状元素的特性，但是各有取舍。因此行内块状元素在日常的使用中，由于其特性，使用的次数也比较多。

行内块状元素特征：(1)不自动换行

(2)能够识别宽高

(3)默认排列方式为从左到右

4.同一类型的元素，到底该用哪个。

其实同一类型的元素，属性是一样的，显示出的效果也一样，用哪个都可以。可是为了代码的可读性，规范化，建议特定的场景用特定的元素。

比如：盒子容器用div，文字段落用p，反过来效果一样，但不符合规范。

块级元素和行内元素的分类

html中的块级元素：

标签	描述
<address>	定义地址。
<article>	定义文章。

<aside>	定义页面内容之外的内容。
<audio>	定义声音内容。
<blockquote>	定义长的引用。
<canvas>	定义图形。
<caption>	定义表格标题。
<dd>	定义定义列表中项目的描述。
	定义文档中的节。
<dl>	定义定义列表。
<dt>	定义定义列表中的项目。
<details>	定义元素的细节。
<fieldset>	定义围绕表单中元素的边框。
<figcaption>	定义 figure 元素的标题。
<figure>	定义媒介内容的分组，以及它们的标题。
<footer>	定义 section 或 page 的页脚。
<form>	定义供用户输入的 HTML 表单。
<h1> to <h6>	定义 HTML 标题。
<header>	定义 section 或 page 的页眉。
<hr>	定义水平线。
<legend>	定义 fieldset 元素的标题。
	定义列表的项目。
<menu>	定义命令的列表或菜单。
<meter>	定义预定义范围内的度量。
<nav>	定义导航链接。
<noframes>	定义针对不支持框架的用户的替代内容。
<noscript>	定义针对不支持客户端脚本的用户的替代内容。

	定义有序列表。
<output>	定义输出的一些类型。
<p>	定义段落。
<pre>	定义预格式文本。
<section>	定义 section。
<table>	定义表格。
<tbody>	定义表格中的主体内容。
<td>	定义表格中的单元。
<tfoot>	定义表格中的表注内容（脚注）。
<th>	定义表格中的表头单元格。
<thead>	定义表格中的表头内容。
<time>	定义日期/时间。
<tr>	定义表格中的行。
	定义无序列表。

html中的行内元素：

标签	**描述**
<a>	定义锚。
<abbr>	定义缩写。
<acronym>	定义只取首字母的缩写。
<b>	定义粗体字
<bdo>	定义文字方向。
<big>	定义大号文本。
 	定义简单的折行。
<button>	定义按钮 (push button)。
<cite>	定义引用(citation)。

<code>	定义计算机代码文本。
<command>	定义命令按钮。
<dfn>	定义定义项目。
<del>	定义被删除文本。
	定义强调文本。
<embed>	定义外部交互内容或插件。
<i>	定义斜体字。
<img>	定义图像。
<input>	定义输入控件。
<kbd>	定义键盘文本。
<label>	定义 input 元素的标注。
<map>	定义图像映射。
<mark>	定义有记号的文本。
<objec>	定义内嵌对象。
<progress>	定义任何类型的任务的进度。
<q>	定义短的引用。
<samp>	定义计算机代码样本。
<select>	定义选择列表（下拉列表）。
<small>	定义小号文本。
	定义文档中的节。
	定义强调文本。
<sub>	定义下标文本。
<sup>	定义上标文本。
<textarea>	定义多行的文本输入控件。
<time>	定义日期/时间。

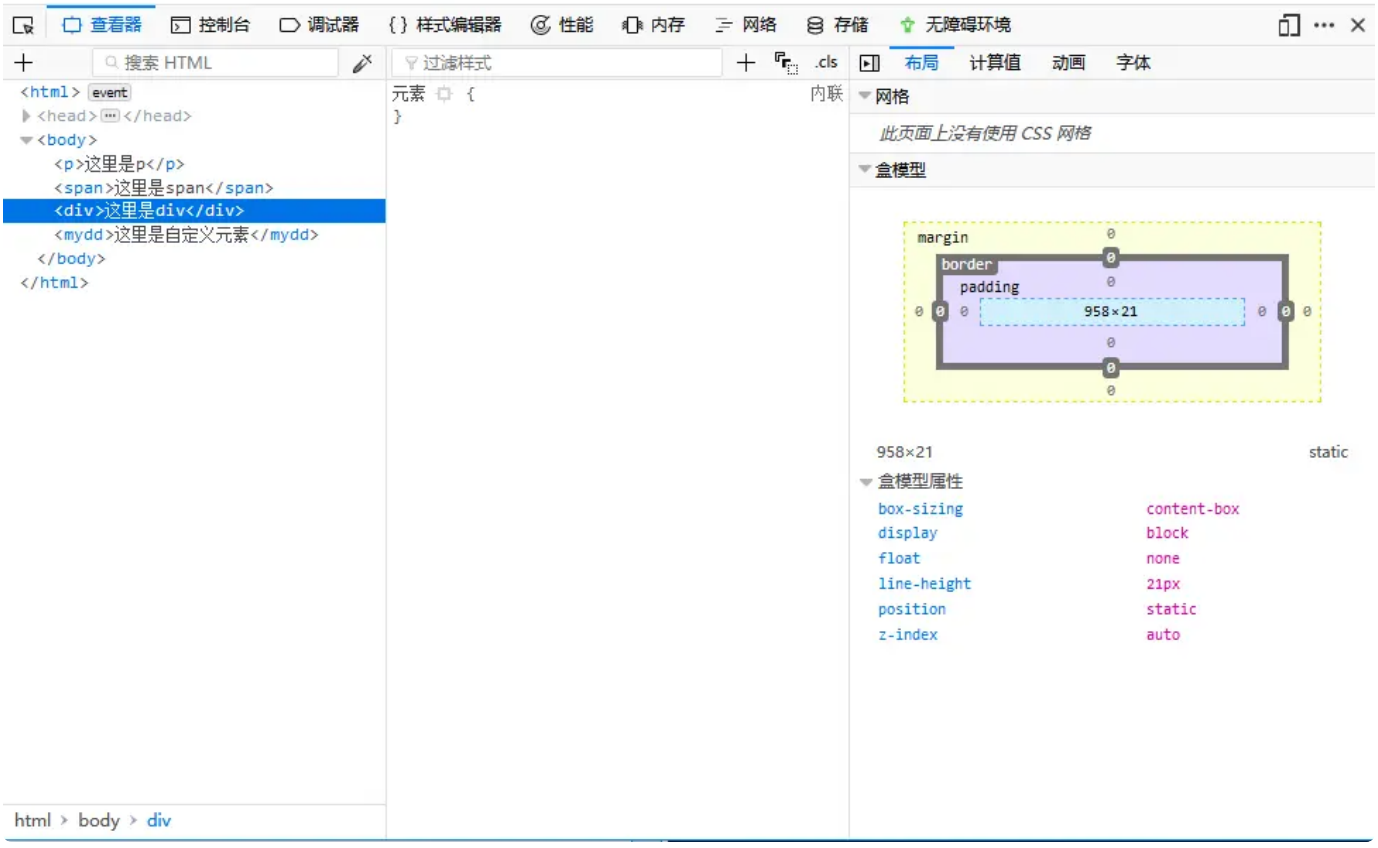
<tt>	定义打字机文本。
<var>	定义文本的变量部分。
<video>	定义视频。
<wbr>	定义可能的换行符。

注：当你不知道某一元素为什么属性时，可通过F12调用浏览器调试器查看

图例：

这里是p

这里是span
这里是div
这里是自定义元素



居中问题

我们在制作页面的时候，经常需要把元素进行水平垂直居中，下面是行内元素和块级元素水平垂直居中的多种方法，供大家选择运用。

1.行内元素水平垂直居中的方法

1.1 水平居中

text-align:center; /* 实现行内元素水平居中 */

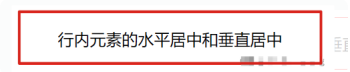
1.2 垂直居中

1.2.1单行文本的垂直居中

只需要让元素高度和行高相等即可；

HTML | 复制代码

```
<style>
    div {
        height: 50px;
        width: 300px;
        border: 1px solid red;
        line-height: 50px;    /* 行高和height相等就能使行内元素垂直居中 */
        text-align: center;   /* 实现行内元素水平居中 */
    }
</style>
<div>
  <span>行内元素的水平居中和垂直居中</span>
</div>
```

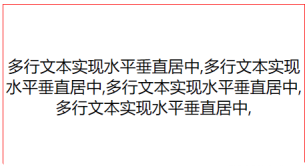


1.2.2 多行文本的垂直居中

① 不固定高度的垂直居中——通过设置上下的padding值即可实现

HTML | 复制代码

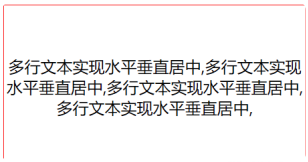
```html
<style>
.box {
    width: 300px;
    border: 1px solid red;
    text-align: center;          /* 实现水平居中 */
    /* line-height: 200px; */   /* 多行文本时，line-height不能再实现垂直居中了 */
    padding: 50px 0;    /* 在盒子div不固定高度的时候，可以使用设置上下padding值来实现垂直居中 */
}
</style>
<div class="box">
    <span>多行文本实现水平垂直居中,多行文本实现水平垂直居中,多行文本实现水平垂直居中,多行文本实现水平垂直居中,</span>
</div>
```



② 固定高度的垂直居中——通过使用表格属性实现，具体为使用display设置行内元素的父盒子为table，然后将行内元素设置为display为table-cell，然后配合样式vertical-align：middle来实现垂直居中

HTML | 复制代码

```html
<style>
.box {
   /* 在固定高度的时候，使用display设置为table，配合样式vertical-align：middle来实现垂直居中 */
   display: table;
   width: 300px;
   height: 200px;    /* 固定盒子高度为200px */
   border: 1px solid red;
   text-align: center;   /* 实现水平居中 */
        }
.box span {
   display: table-cell;   /* 子元素需要设置为display:table-cell */
   vertical-align: middle;   /* 配合使用vertical-align: middle; */
}
</style>
<div class="box">
    <span>多行文本实现水平垂直居中,多行文本实现水平垂直居中,多行文本实现水平垂直居中,多行文本实现水平垂直居中</span>
</div>
```



1.3 小结

综上，行内元素的水平居中，都是使用 text-align: center；而垂直居中就要根据情况选择不同方法了，需要注意每种方法的使用前提。

2.块级元素水平垂直居中的方法

2.1 使用flex布局实现水平垂直居中

这种方法借助flex布局中父项属性的①justify-content：center；实现主轴上子元素居中；（默认水平）　②align-items：center；实现侧轴上子元素居中；（默认垂直）

HTML | 复制代码

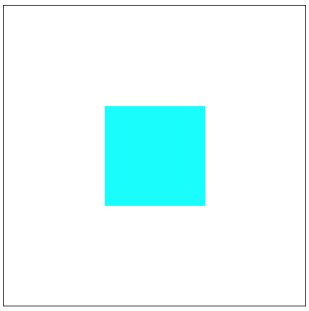
```html
<style>
.box1 {
    /* 使用弹性盒模型(flex布局) */
    display: flex;
    justify-content: center;    /* 默认主轴为x轴，实现水平居中 */
    align-items: center;       /* 实现垂直居中 */
    width: 300px;
    height: 140px;
    border: 1px solid red;
}
.item1 {
    width: 100px;
    height: 60px;
    background-color: pink;
}
</style>
<div class="box">
    <div class="item1"></div>
</div>
```

2.2 绝对定位配合margin的方式实现水平垂直居中

注意：使用这种方法的前提条件是父盒子的高度和宽度必须是固定的，只要变化了，就不再水平垂直居中，除非重新计算margin的偏移量。

HTML | 复制代码

```
<style>
.box {
    position: relative;   /* 父相 */
    width: 300px;
    height: 300px;
    border: 1px solid black;
}
.item2 {
    position: absolute;  /* 子绝 */
    left: 50%;  /* 与父盒子最左边的距离为父盒子宽度的一半 这就偏右了*/
    top: 50%;   /*与父盒子最上边的距离为父盒子高度的一半   这就偏下了*/
    margin-left: -50px;  /* margin注意要为负值   往回（向左）走宽度的一半*/
    margin-top: -50px;   /* margin注意要为负值   往回（向上）走高度的一半 */
    width: 100px;
    height: 100px;
    background-color: aqua;
}
</style>
div class="box">
        <div class="item2"></div>
</div>
```

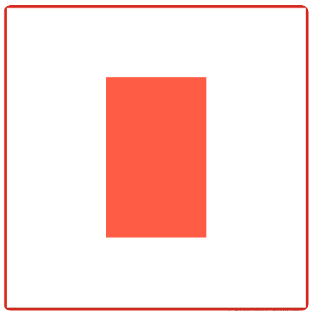


2.3 绝对定位配合margin的方式实现水平垂直居中

注意：上述使用margin的方法需要固定宽高，而现在要介绍的方法是不需要固定宽高，就算宽高改变了，依旧是水平垂直居中的。

HTML | 复制代码

```
<style>
.box {
    position: relative;   /* 父相 */
    width: 300px;   /* 可以更改宽度，但不影响水平居中的效果 */
    height: 300px;  /* 可以更改高度，但不影响垂直居中的效果 */
    border: 1px solid black;
}
.item3 {
    position: absolute;  /* 子绝 */
    left: 0;
    top: 0;
    right: 0;
    bottom: 0;
    /* 以上代码让子元素处于父盒子左上角 */
    margin: auto;    /* 使用auto就可以让子元素水平垂直居中 */
    width: 100px;
    height: 160px;
    background-color: tomato;
}
</style>
div class="box">
    <div class="item3"></div>
</div>
```



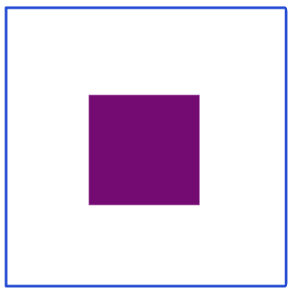
2.4 绝对定位配合translate的方式实现水平垂直居中

问：在不使用flex布局时，父盒子宽高又不固定时，如何实现水平垂直居中?

答：可以采取绝对定位配合css3中的translate的方式实现

HTML | 复制代码

```html
<style>
.box {
    position: relative;    /* 父相 */
    width: 300px;     /* 宽度随便更改，都能保证水平居中 */
    height: 300px;    /* 高度随便更改，都能保证垂直居中 */
    border: 1px solid blue;
}
.item4 {
    position: absolute;  /* 子绝 */
    left: 50%;  /* 与父盒子最左边的距离为父盒子宽度的一半 这就偏右了*/
    top: 50%;   /*与父盒子最上边的距离为父盒子高度的一半   这就偏下了*/
    transform: translate(-50%, -50%);  /* 注意要为负值，向左向上移动自身宽度和高度的一半 */
    width: 110px;
    height: 110px;
    background-color: purple;
}
</style>
div class="box">
        <div class="item2"></div>
</div>
```



translate（）括号里如果写百分比，那么移动的距离是相对自身宽度或高度的百分比。

所以这里需要向左向上移动自身宽度和高度的一半，那么就使用translate（-50%，-50%）；

一定要注意网页中，向右向下是为正的，所以向左向上即为负值。

2.5 小结

块级元素水平垂直居中的方法还是比较多的，但是每种方法都有其对应的条件，所以要根据条件进行选择最合适的方法，但是我们一般使用最多的是flex布局和绝对定位+css3中的translate这两种方式。除非需求不允许这两种方式，才会去选择绝对定位配合margin的方式。
