一般喜欢给单选框和多选框绑定文字：就是可以点击相关内容，也选择选框。提升用户体验。

方法：将文字内容卸载一个label标签内部，让lable标签的for属性值和input标签的id属性值一致。

html杂项：

1. 注释：里面的内容是给程序员看的。在浏览器上不会渲染
2. 作用：清晰标注，有一部分代码不想显示，希望保留。
3. 书写：<!—注释内容-->
   1. Sublime ：ctrl+/
4. 字符实体：
   1. 我们的一些特殊符号，并不能直接在文本内容里书写。
   2. Html提供了一套书写特殊符号的替代符号，就是字符实体，或转义字符

<小于号：&lt; less than

>大于号：&gt; greater than

无换行空格：&nbsp; non-breaking space

版权©：&copy; copy right

Css概念：

1. css:cascading style sheets 层叠式样式表。规定了html标签在网页上的显示样式。
   1. html4的时候，w3c组织将html的结构和样式做了分离
   2. 前端三层：
      1. Html 结构层，打架网页的整体结构。
      2. Css 样式层，装饰页面。
      3. Javascript 行为层，一些页面交互效果。
   3. 这三层标准都是w3c组织来维护。
   4. Css发展到现在，最新版本是css3，最常用的是css2.1
2. Css作用
   1. Css两个重要的概念：层叠式，样式。
   2. 样式：html标签在页面的显示效果。
   3. 某一个标签又什么样式，直接将对应的属性及属性值罗列出来。
   4. Css样式设置的时候，有两个关键：选择器，样式表。
   5. Css作用细化：
      1. 给文本添加文字显示样式
      2. 给盒子添加属性进行结构布局
3. Css几个小属性
   1. 文本样式
      1. 字体、颜色、大小
         1. 大小：font-size。本身是一个符合属性font；里面的单一属性，需要用font单一属性名。属性值，以像素为单位。
         2. 颜色：color。属性值有几种选择方案：十六进制，RGB，rgba，英文颜色名。
         3. 字体：font-family。复合属性的一个单一属性。中文字体和英文字体。
         4. 书写的时候将英文字体写在前面，中文字体在后面。
         5. 英文字体常用：Arial，consolas。
         6. 中文字体：默认字体为宋体，常用字体有宋体、微软雅黑，一般写字体时，标注好备选字体。
         7. 字体用引号包裹，字体之间用逗号给开，表示或。
         8. 为了追求加载速度，将一些中文字体名字写成英文表示法。
            1. 微软雅黑：Microsoft Yahei
            2. 宋体：SimSun
         9. 盒子实体化基本属性
         10. 实体化：将盒子给宽、高、背景色、边框。
         11. 宽度：width。属性值是像素为单位。
         12. 高度：height。属性值也是像素为单位。
         13. 背景色：background-color。属性值就是颜色色值。
         14. 边框：border。复合属性，有多个属性值，属性值之间用空格隔开。边框的宽度、颜色、线的类型。
         15. 实现边框：solid。
4. Css的书写位置
   1. 根据书写的css属性位置不同，将我们的样式表分为三种：行内式，内嵌式，外链式。
   2. 行内式：
      1. Css属性写在标签的一个style属性里，可以书写他的所有css样式。
   3. 内嵌式：
      1. 写在html文件里的，head标签内有一对style标签，是一个双标签，我们所有的css样式属性都写在双标签内部。
      2. Style标签有个属性type，属性值“text/css”，表示我们内部写的是纯文本样式代码。
5. Css的注意事项：
   1. Html的属性和属性值，键值对：k=‘v’
   2. Css的属性和属性值，键值对：k：v；
   3. 多个属性之间必须用分号分隔。不写出错。
   4. 内嵌式。所有的样式必须写在一对大括号里。
   5. Css样式，对于空格，换行，缩进不敏感
   6. 合理换行，提高可读性。
   7. 为了提高加载速度，需要将所有的空格，缩进，换行删除，即压缩代码。

基础选择器

1. 选择器：指的是我们选择需要添加样式的标签元素的模式
2. 标签选择器：通过标签名直接选择相应的标签。标签是什么我们就用什么选择。
3. 用途，默认样式的清除
4. Id选择器
   1. 通过标签的id属性值来选择相应的标签。
   2. 选择器书写：#开头，后面紧跟id名。中间没有任何空格。
   3. 由于我们的id页面唯一，id选择器只能选中一个元素。
   4. 标签都有id属性，属性值也就是id名，命名需要遵循一个规则：
      1. 必须以字母开头（严格区分大小写，A和a表示的不一样），后面可以是数字、下划线或者横线。
      2. 一个页面不允许出现两个相同的id名，即使是不同类型的标签。
5. Id选择器劣势：只能选中一个元素。如果有相同的样式，只能分开命id名，多次书写相同属性。
6. 标签选择器劣势：会选中所有的元素，不能进行单独或者部分控制。
7. Id常用场景：并不是用来添加样式，留给js添加行为。

类选择器

1. 通过标签的clss属性来选择这个标签。
2. 选择器写法：开头，后面class属性值。中间不能有空格。
3. Class属性，值可以不唯一。类选择器选中的是相同的class 名的元素。
4. 标签的class属性，属性值命名规则与id一样。
5. 优点：可以选择一部分标签，添加相同的样式。
6. 根据类的特性，我们要习惯使用原子类。原子类设置的原则，里面只设置单一的css属性，某一个标签需要添加这个属性，可以直接添加这个原子类的类名
7. 一个标签的class可以有多个属性值，有多个类名。多个类名之间要用空格隔开
8. 类选择器好用，多用于css样式书写。
9. 类上样式，id上行为。

通配符\*

可以选中包括body在内的所有标签，现在不常用，效率不高。

高级选择器

1. 基础选择器并不能满足我们所有的需求，在基础选择器上进行了一些延申。
2. 三种：后代选择器、交集选择器、并集选择器。
   1. 后代选择器：又叫做包含选择器。通过标签之间的嵌套关系、层级关系，限定我们大体范围，在范围内具体查找相关元素。
      1. 后代选择器：根据嵌套关系，空格左侧是右侧的祖先元素。
      2. 后代选择器空格两边只要是后代关系都可以，不一定是父子关系
      3. 一般的开始的祖先元素都不使用标签选择器。
      4. 可以根据嵌套关系来确定最终元素
   2. 交集选择器：
      1. 既满足条件1也满足条件2
      2. 两种或者以上选择器同时存在一个标签上
      3. 写法：选择器之间直接连接，没有任何符号。
      4. 交集使用的基础选择器，可以都是类选择器，可以标签和类的混写。
      5. 交集的连续书写。
      6. 兼容问题：IE6不支持类的连续交集。
   3. 并集选择器
      1. 有很多元素可能都有相同css样式，我们可以通过并集选择器将这些标签写在一起，统一设置css样式，前面所有的选择器都可以作为并集里的一项，在选择器中间用逗号隔开。

继承性和层叠性

1. 继承性
   1. 有一些属性给祖先元素设置了，后代元素会继承
   2. Css继承性：会继承文字的样式，不继承盒子的样式
2. 层叠性：
   1. 我们学了多种选择器，同一个标签可以用多个选择器选中。
   2. 问题：多个选择器选中一个元素，相同的样式听？
   3. 涉及到一个选择器权重问题：
      1. 标签、类、id权重逐渐增高
      2. 通配符权重最低
   4. 到底一个标签的样式听谁的：
      1. 选择器都选中了元素。
         1. 首先比较权重，权重大的层叠权重小的
         2. 计算权重的方法：数基础选择器的个数，依次比较id个数🡪类的个数🡪标签的个数。
         3. 如果权重一样，比较是书写顺序，谁写在后面，听谁的
      2. 如果没有选中元素，样式靠继承。
         1. 第一种：祖先元素距离目标元素距离不同，继承的样式，谁近，听谁的
         2. 第二种情况：如果距离相同，去比较权重。
         3. 第三种情况：如果距离相同，权重相同，谁在后，听谁的
3. 关键字 ！important
   1. 给单一属性添加，会给这个属性的权重提升到最大。前提：不适用于就近原则。