* 1. HTML结构标记+说明  
    
  • <html>：HTML文档的根元素，整个HTML页面都包含在这个标签内。  
    
  • <head>：包含了文档的元数据，如<title>（定义文档的标题，显示在浏览器的标题栏）、<meta>（用于提供关于HTML文档的元信息，如字符集、页面描述等）、<link>（用于链接外部的样式表等资源）、<script>（用于嵌入或引用JavaScript代码）等。  
    
  • <body>：包含了文档的所有内容，如文本、图像、链接、表单、表格等可视元素。  
    
  2. 表单标记  
    
  • <form>：用于创建HTML表单。它有一些重要的属性，如action（指定表单数据提交的URL）、method（指定提交数据的方式，常见的有get和post）。  
    
  • <input>：是最常用的表单元素，根据type属性的不同可以创建多种类型的输入字段，如text（文本输入框）、password（密码输入框，输入内容显示为黑点等加密形式）、submit（提交按钮）、reset（重置按钮）、radio（单选按钮）、checkbox（复选框）等。  
    
  • <textarea>：用于创建多行文本输入区域。  
    
  • <select>和<option>：<select>创建下拉列表，<option>则是<select>中的选项。  
    
  3. 表格标记  
    
  • <table>：定义一个表格。  
    
  • <tr>：表示表格中的行（table row）。  
    
  • <td>：表示表格中的单元格（table data），用于存放表格内容。如果是表头单元格，可以使用<th>，它默认会有一些样式上的加粗、居中效果。  
    
  二、  
    
  1. css文本属性+说明  
    
  • color：用于设置文本的颜色，可以使用颜色名称（如red）、十六进制值（如#FF0000）、RGB值（如rgb(255, 0, 0)）等。  
    
  • font - family：指定文本的字体系列，如"Arial", sans - serif，浏览器会按照顺序尝试使用指定的字体，如果找不到则使用默认的sans - serif字体。  
    
  • font - size：设置文本的大小，可以使用绝对单位（如px）或相对单位（如em、rem）。  
    
  • font - weight：控制文本的粗细，如normal（正常）、bold（加粗），也可以使用数字值（100 - 900）。  
    
  • text - align：设置文本的水平对齐方式，有left（左对齐）、right（右对齐）、center（居中对齐）、justify（两端对齐）。  
    
  • text - decoration：用于添加文本修饰，如underline（下划线）、overline（上划线）、line - through（删除线）等。  
    
  2. css选择器+说明  
    
  • 元素选择器：直接使用HTML元素名称作为选择器，如p选择所有的<p>段落元素。  
    
  • 类选择器：以.开头，用于选择具有特定类名的元素，如.my - class选择所有class="my - class"的元素。类名可以在多个元素上使用，实现样式的复用。  
    
  • ID选择器：以#开头，用于选择具有特定ID的元素，如#my - id选择id="my - id"的元素。ID在一个页面中应该是唯一的。  
    
  • 后代选择器：用空格分隔，如ul li选择<ul>元素内部的所有<li>元素。  
    
  • 子选择器：用>分隔，如div > p选择<div>元素的直接子元素<p>。  
    
  • 相邻兄弟选择器：用+分隔，如h2 + p选择紧跟在<h2>元素后面的<p>元素。  
    
  • 通用兄弟选择器：用~分隔，如h2~p选择跟在<h2>元素后面的所有<p>元素。  
    
  3. css样式引用方式+示例  
    
  • 内联样式：直接在HTML元素的style属性中设置样式，例如<p style="color: red; font - size: 16px;">这是一个段落</p >。  
    
  • 内部样式表：在HTML文档的<head>部分使用<style>标签定义样式，例如：  
  <head>  
  <style>  
  p {  
  color: blue;  
  font - size: 14px;  
  }  
  </style>  
  </head>  
  • 外部样式表：创建一个独立的.css文件，然后在HTML文档中使用<link>标签引用，例如：  
    
  • 在styles.css文件中有：  
  p {  
  color: green;  
  font - size: 12px;  
  }  
  • 在HTML文件中：  
  <head>  
  <link rel="stylesheet" href="styles.css">  
  </head>  
  三、  
    
  1. 盒子模型  
    
  • 盒子模型描述了HTML元素在页面布局中所占的空间。一个HTML元素的盒子模型由内容（content）、内边距（padding）、边框（border）和外边距（margin）组成。  
    
  • 内容（content）：是元素实际包含的内容，如文本、图像等。  
    
  • 内边距（padding）：是内容与边框之间的空白区域，可以分别设置上、下、左、右内边距（padding - top、padding - bottom、padding - left、padding - right）。  
    
  • 边框（border）：围绕元素内容和内边距的边界，可以设置边框的宽度、样式（如实线、虚线等）和颜色。  
    
  • 外边距（margin）：是元素与相邻元素之间的空白区域，同样可以分别设置上、下、左、右外边距（margin - top、margin - bottom、margin - left、margin - right）。  
    
  2. Css背景属性（综合背景设置）+说明  
    
  • background - color：设置元素的背景颜色，用法与color属性类似。  
    
  • background - image：用于设置元素的背景图像，可以是本地图像文件（如url('image.jpg')）或渐变等特殊图像。  
    
  • background - repeat：控制背景图像的重复方式，有repeat（默认，在水平和垂直方向重复）、repeat - x（仅在水平方向重复）、repeat - y（仅在垂直方向重复）、no - repeat（不重复）。  
    
  • background - position：指定背景图像的起始位置，可以使用关键字（如top left）、百分比或具体的像素值。  
    
  • background - size：设置背景图像的大小，可以使用关键字（如cover，缩放背景图像以完全覆盖容器，可能会裁剪图像；contain，缩放背景图像以完全包含在容器内，可能会有空白区域）或具体的尺寸值。  
    
  • 综合设置可以使用background属性，例如background: url('image.jpg') no - repeat center center / cover;，它同时设置了背景图像、不重复、居中显示且以覆盖方式缩放。  
    
  四、  
    
  1. 块元素有哪些、行内元素有哪些？  
    
  • 块元素：常见的块元素有<div>、<p>、<h1> - <h6>、<ul>、<ol>、<li>、<table>、<form>等。块元素在页面中会独占一行，默认宽度为父元素的100%（除非设置了具体宽度），并且可以设置高度、内边距、外边距等属性。  
    
  • 行内元素：常见的行内元素有<span>、<a>、<img>、<input>、<label>、<strong>、<em>等。行内元素不会独占一行，它们按照文本流的顺序排列，宽度和高度由内容自动撑开，并且垂直方向的外边距和内边距设置可能不会按照预期显示（在某些浏览器中）。  
    
  2. 如何切换行内元素和块元素  
    
  • 使用display属性。如果要将行内元素转换为块元素，可以设置display: block;。例如，将<span>转换为块元素：  
  span {  
  display: block;  
  }  
  • 要将块元素转换为行内元素，可以设置display: inline;。例如，将<div>转换为行内元素：  
  div {  
  display: inline;  
  }  
  • 还有display: inline - block;，它使元素具有行内元素的排列方式，但又可以像块元素一样设置宽度、高度等属性。  
    
  五、  
    
  1. 元素浮动有哪些  
    
  • 在CSS中，元素的浮动有left（向左浮动）和right（向右浮动）。当一个元素设置为浮动时，它会脱离文档流的正常布局，向左或向右移动，直到它的外边缘碰到包含框或另一个浮动元素的外边缘。  
    
  2. 如何清除浮动  
    
  • 使用clear属性。clear属性有left（清除左浮动的影响）、right（清除右浮动的影响）、both（清除左右浮动的影响）。例如，如果有一个<div>元素受到前面浮动元素的影响，可以在这个<div>上设置clear: both;来使其不受浮动元素的影响，正常布局在浮动元素下方。  
    
  • 也可以使用伪元素来清除浮动，例如：  
  .clearfix::after {  
  content: "";  
  display: table;  
  clear: both;  
  }  
  • 然后将class="clearfix"添加到需要清除浮动的父元素上。  
    
  3. 元素定位方式  
    
  • 静态定位（position: static）：这是HTML元素的默认定位方式，元素按照正常的文档流进行布局，不能通过top、bottom、left、right属性进行定位。  
    
  • 相对定位（position: relative）：元素相对于它在正常文档流中的位置进行定位。可以使用top、bottom、left、right属性来移动元素，但元素原来所占的空间仍然保留在文档流中。  
    
  • 绝对定位（position: absolute）：元素相对于最近的已定位祖先元素（如果没有已定位的祖先元素，则相对于<body>元素）进行定位。元素脱离文档流，原来所占的空间不再保留，其他元素会填补它原来的位置。  
    
  • 固定定位（position: fixed）：元素相对于浏览器窗口进行定位，不随页面滚动而移动，脱离文档流。  
    
  • 粘性定位（position: sticky）：元素在正常文档流中，当页面滚动到特定位置时，元素会固定在某个位置，它是相对定位和固定定位的混合体。