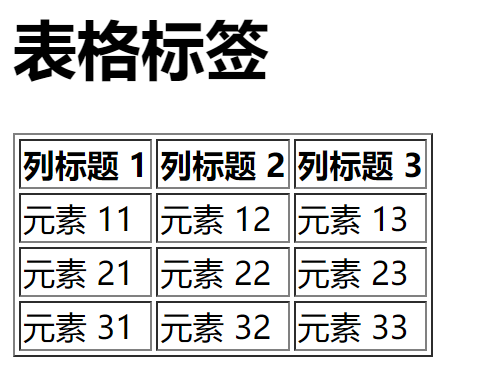
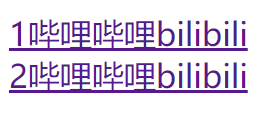
**HTML+CSS+JavaScript学习笔记**



1.3.6 “a标签”：超链接标签

<a href="https://www.bilibili.com/" target="\_blank">1哔哩哔哩bilibili</a>

<a href="https://www.bilibili.com/" target="\_parent">2哔哩哔哩bilibili</a>



target属性：target="\_blank" 在新的页面显示 target="\_parent" 在当前页面显示

1.3.7 “img标签”：图片标签

<!-- src属性可为相对路径和绝对路径 -->

<img src="素材\图片\QQ图片20231125145916.jpg" alt="">

<br>

<!-- alt属性可留白，错误路径alt有值则不显示图片 -->

<img src="素材\图片\QQ图片.jpg" alt="图片无法显示">

<br>

<!-- width height 属性可设置图片宽度和高度 -->

<img src="素材\图片\QQ图片20231125145916.jpg" alt="" width="300" height="200">



1.3.8 “div标签”：块标签

1.3.8 “span标签”：行标签

1. 表单
   1. “form”

<!-- form表单 -->

<form action="">

<!-- 提示 -->

<input type="text" placeholder="请输入内容："> <br>

<!-- 赋值 -->

<input type="text" value="请输入内容："> <br><br>

<label for="">用户名：</label>

<input type="text" placeholder="请输入用户名："> <br><br>

<label for="">密码：</label>

<input type="password" placeholder="请输入密码："> <br><br>

<!-- 选项 -->

<!-- 单选radio -->

<label for="">性别：</label>

<input type="radio" name="gender"> 男

<input type="radio" name="gender"> 女

<input type="radio" name="gender"> 其他

<br><br>

<!-- 多选checkbox -->

<label for="">爱好：</label>

<input type="checkbox" name="hobby"> 唱

<input type="checkbox" name="hobby"> 跳

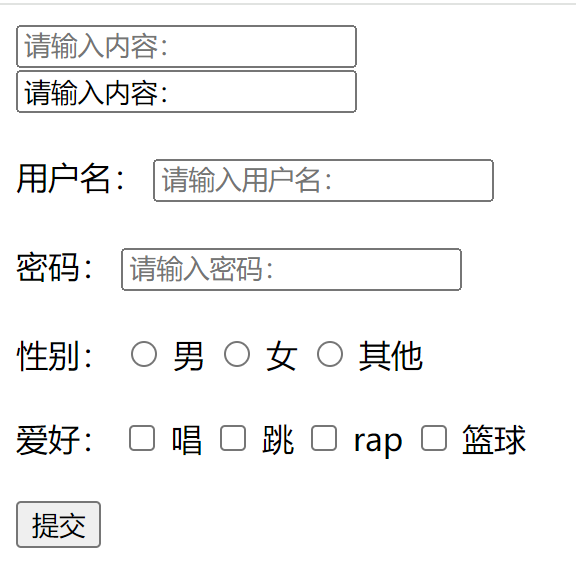
<input type="checkbox" name="hobby"> rap

<input type="checkbox" name="hobby"> 篮球

<br><br>

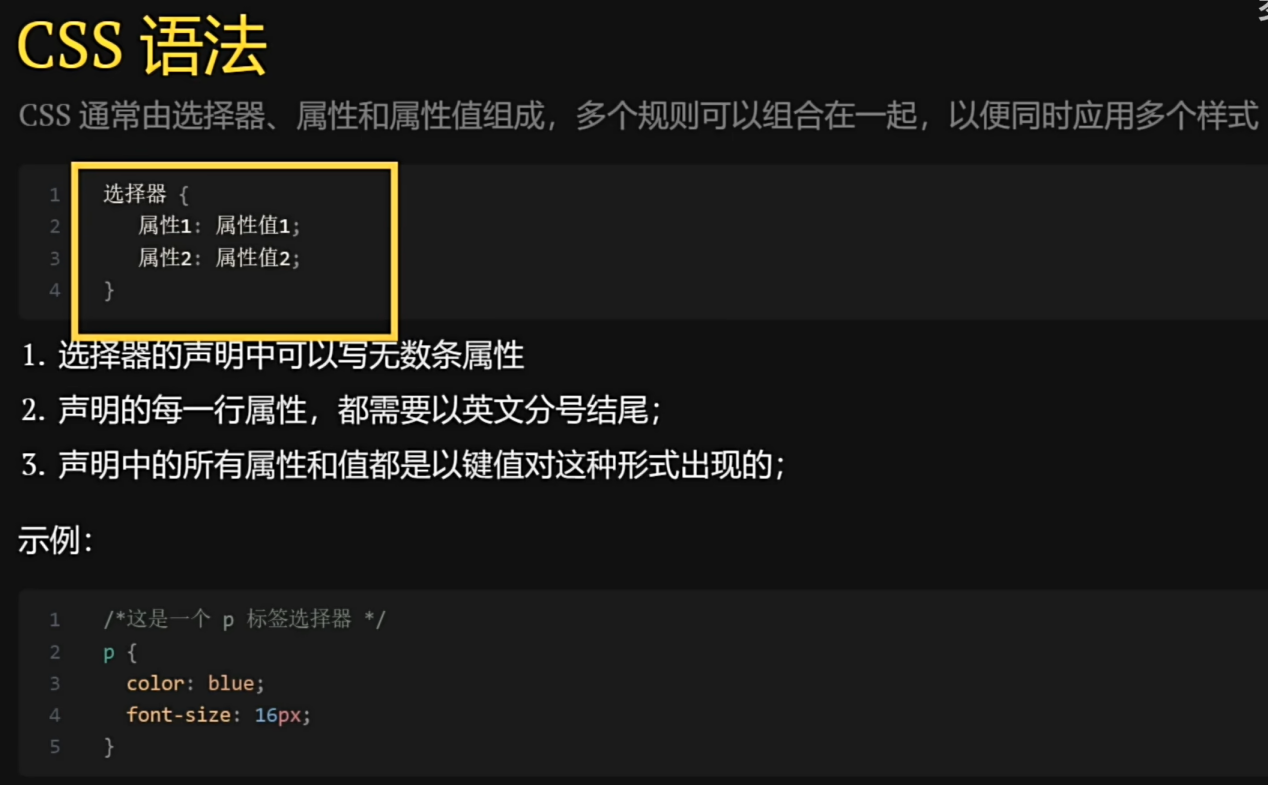
<!-- 提交submit (action赋值确定提交服务器位置) -->

<input type="submit" name="提交" id="">



**二、CSS**

1. 基本语法



1. CSS导入

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>CSS导入</title>

    <link rel="stylesheet" href="CSS样式/style.css">

    <style>

        h2{

            color: blue;

        }

    </style>

</head>

<body>

    <!-- 1、内联样式 (style 在标签内部)-->

    <h1 style="color: red;">这是一个一级标题,采用内联样式</h1>

    <!-- 2、内部样式 (style 在 head 内部)-->

    <h2>这是一个二级标题,采用内部样式</h2>

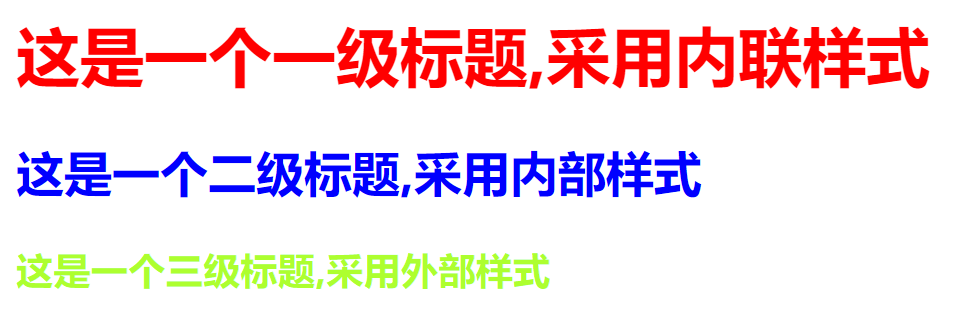
    <!-- 3、外部样式 (在 head 内部用link导入)-->

    <h3>这是一个三级标题,采用外部样式</h3>

    <!-- 不同样式优先级不同，低级会被高级覆盖 (内联样式>内部样式>外部样式) -->

</body>

</html>



1. 选择器



<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>CSS选择器</title>

    <style>

        /\* 元素选择器 \*/

        h1{

            color: aquamarine;

        }

        /\* 类选择器 \*/

        .backgr{

            color: bisque;

            background-color: yellow;

        }

        /\* ID选择器 \*/

        #IDName{

            color: red;

            /\* 更改字体大小 \*/

            font-size: 35px;

        }

        /\* 通用选择器 \*/

        \*{

            /\* 更改字体样式 \*/

            font-family: KaiTi;

        }

        /\* 子元素选择器 \*/

        .father > .son{

            color: chartreuse;

            font-family: SongTi;

            font-size: 35px;

        }

        /\* 后代选择器 \*/

        .father p{

            color: rgb(15, 40, 230);

        }

        /\* 相邻元素选择器 \*/

        h3 + p{

            color: darkgreen;

            font-size: 40px;

        }

        /\* 伪类选择器 \*/

        /\* hover 为悬停在元素上方进行操作 \*/

        #weilei:hover{

            background-color: aqua;

        }

    </style>

</head>

<body>

    <!-- 1.元素选择器 (直接写标签名)-->

    <h1>1.元素选择器实例,采用内部样式</h1>

    <!-- 2.类选择器 (.类名)-->

    <!-- 创建类快捷键：h2.类名 -->

    <h2 class="backgr">2.类选择器实例</h2>

    <!-- 3.ID选择器 (#ID名)-->

    <!-- 创建ID快捷键：h3#ID名 -->

    <h3 id="IDName">3.ID选择器实例</h3>

    <!-- 4.通用选择器 (\*)-->

    <!-- 5.子元素选择器 (.父元素名 > .子元素名)-->

    <div class="father">

        <p class="son">5.子元素选择器实例</p>

    </div>

    <!-- 6.后代选择器 (.祖先元素名 标签名)-->

    <div class="father">

        <p class="son1">6.后代选择器实例1</p>

        <div>

            <p class="grandson">6.后代选择器实例2</p>

        </div>

    </div>

    <!-- 7.相邻元素选择器 (标签名 + 相邻标签名(下一个))-->

    <p>这是一个普通 p 标签</p>

    <h3>7.相邻元素选择器实例</h3>

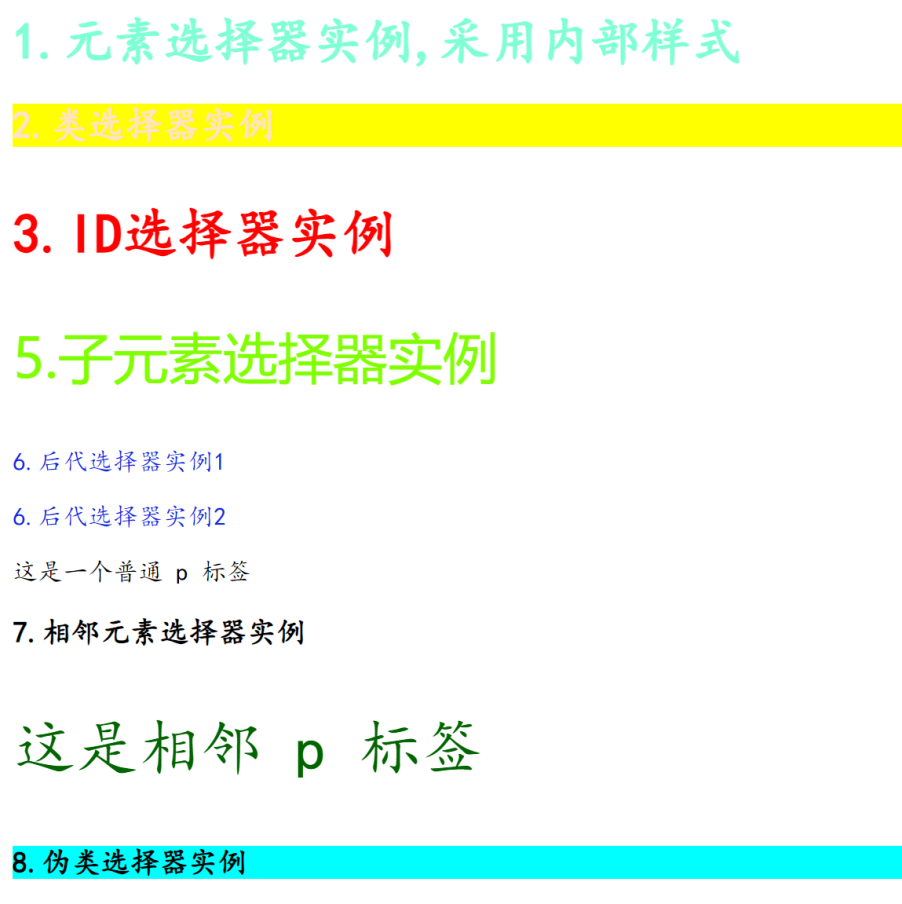
    <p>这是相邻 p 标签</p>

    <!-- 8.伪类选择器 (标签名/.类名/#ID名:操作名)(冒号两边不能有空格！！) -->

    <h3 id="weilei">8.伪类选择器实例</h3>

</body>

</html>



1. CSS常用属性

<!-- 1.复合属性 font -->

<h1 style="font: bolder 50px 'KaiTi';">1.复合属性实例</h1>

<!-- 2.行高/行距 -->

<p style="line-height: 40px;">这是一段文本行高40px,这是一段文本行高40px,

这是一段文本行高40px,这是一段文本行高40px,这是一段文本行高40px,

这是一段文本行高40px,这是一段文本行高40px,这是一段文本行高40px,

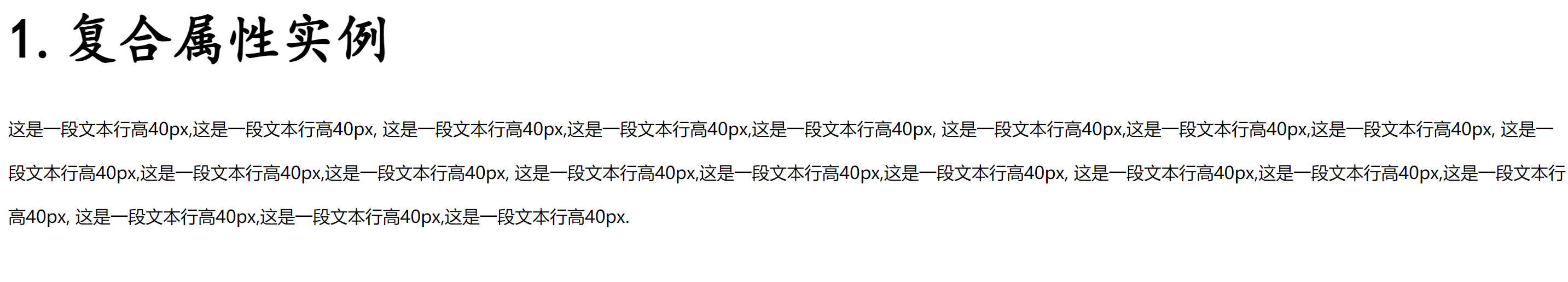
这是一段文本行高40px,这是一段文本行高40px,这是一段文本行高40px,

这是一段文本行高40px,这是一段文本行高40px,这是一段文本行高40px,

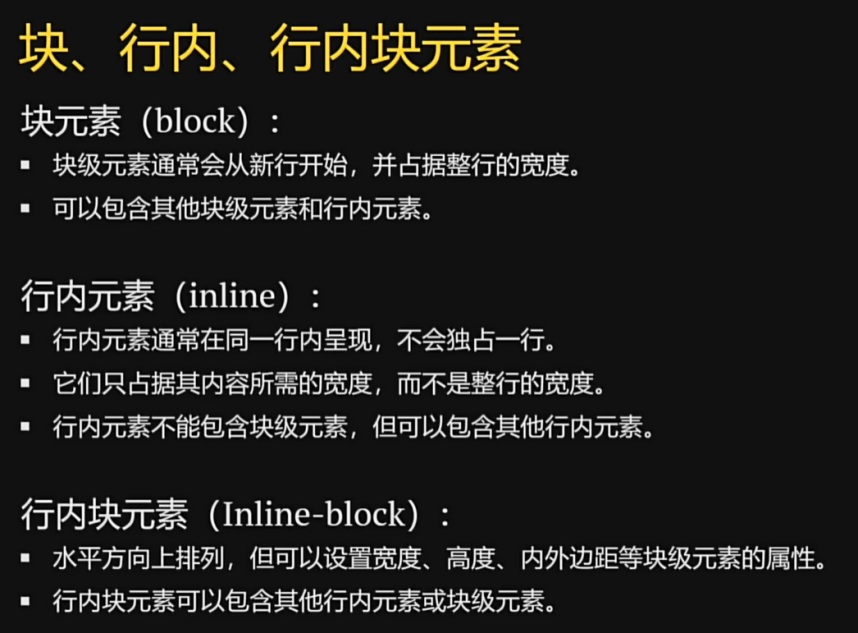
这是一段文本行高40px,这是一段文本行高40px,这是一段文本行高40px,

这是一段文本行高40px,这是一段文本行高40px,这是一段文本行高40px.

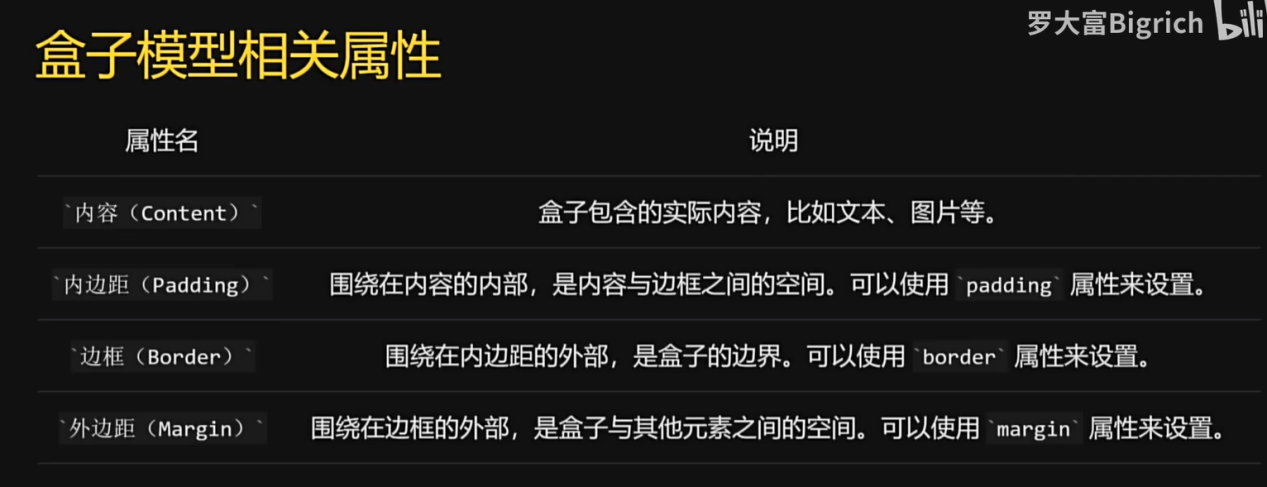
</p>



块、行内、行内块元素的应用与相互转换 (display:)



1. 盒子模型



<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

    <style>

        .exa{

            background-color: aqua;

            display: inline-block;

            border: 20px solid green;

            /\* padding margin 与 border 一样可单独设置属性 \*/

            padding: 20px;

            margin: 100px

        }

        .border-exa{

            background-color: yellow;

            width: 100px;

            height: 80px;

            /\* border 是一个复合属性，与 font 类似 \*/

            /\* border-style 四个值分别表示上、右、下、左边框样式(实线、虚线、点线、双实线) \*/

            border-style: solid dashed dotted double;

            border-color: aqua;

            /\* border-width 四个值分别表示上、右、下、左边框宽度 \*/

            border-width: 40px 20px 10px 0;

            /\* 可单独设置属性

            border-left: 40px solid red;

            border-right: ;

            border-top: ;

            border-bottom: ; \*/

        }

    </style>

</head>

<body>

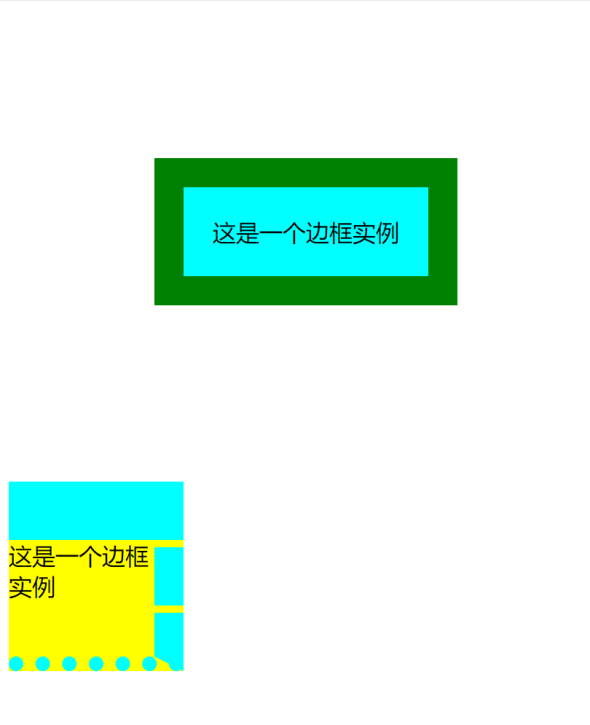
    <!-- 1.边框 (border) -->

    <div class="exa">这是一个边框实例</div><br><br>

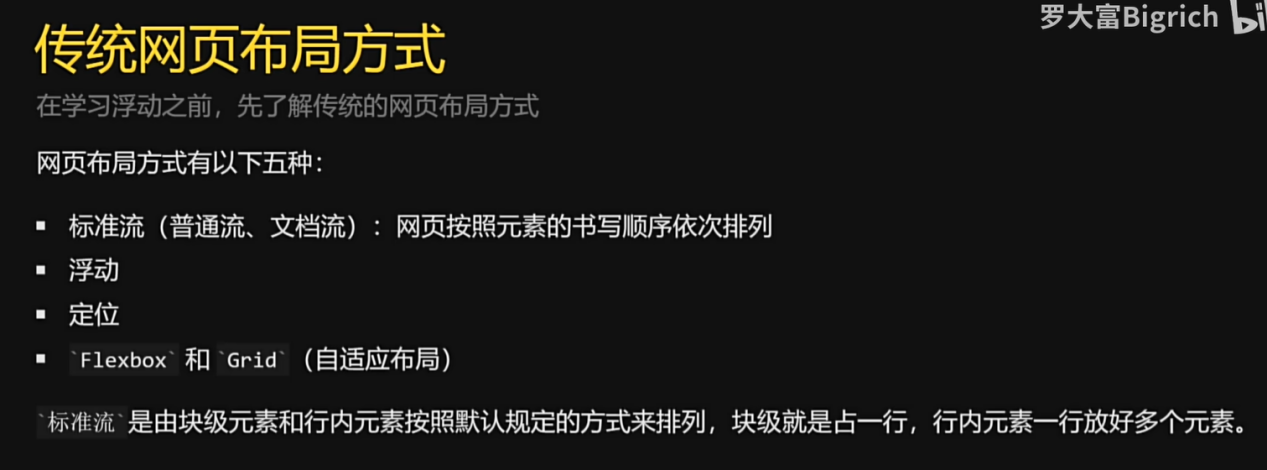
    <div class="border-exa">这是一个边框实例</div>

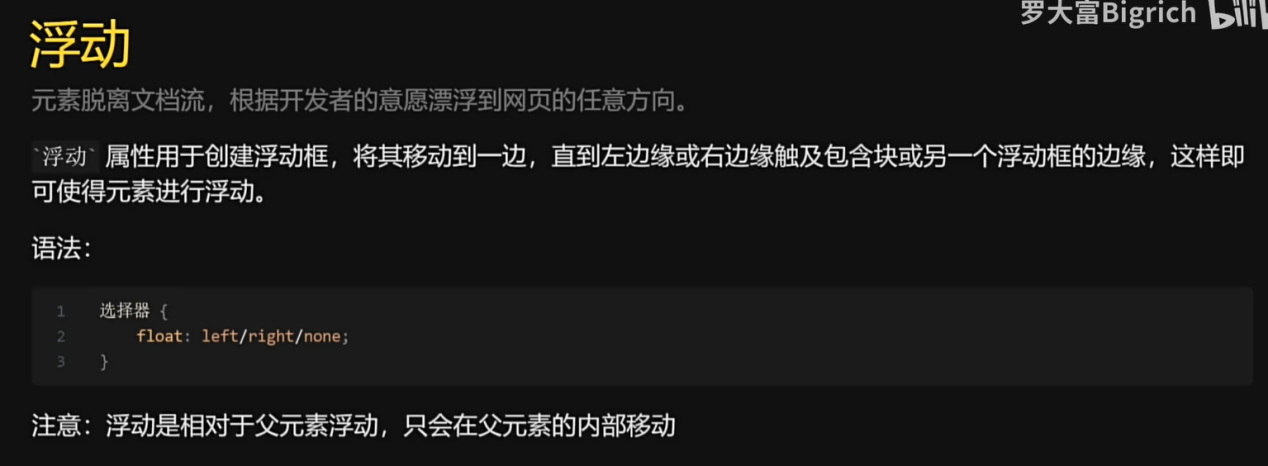
</body>

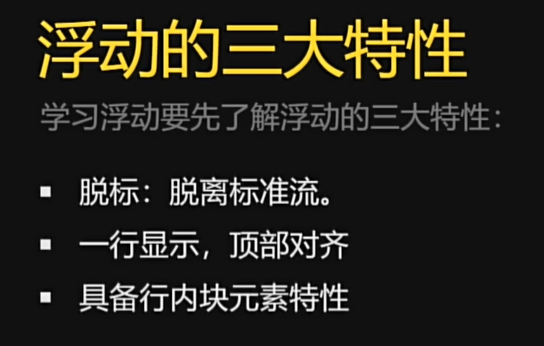
</html>



1. 浮动







<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>浮动</title>

    <style>

        .father{

            background-color: aqua;

            /\* 不设置高度，浮动会导致高度坍塌，格式错乱 \*/

            height: 150px;

            /\* 若未设置高度，则需清除浮动 \*/

            overflow: hidden;  /\* 清除浮动 \*/

        }

        .left-son{

            width: 100px;

            height: 100px;

            background-color: yellow;

            float: left;  /\* 左浮动 \*/

        }

        .right-son{

            width: 100px;

            height: 100px;

            background-color: yellowgreen;

            float: right;  /\* 右浮动 \*/

        }

    </style>

</head>

<body>

    <div class="father">

        <div class="left-son">左浮动</div>

        <div class="right-son">右浮动</div>

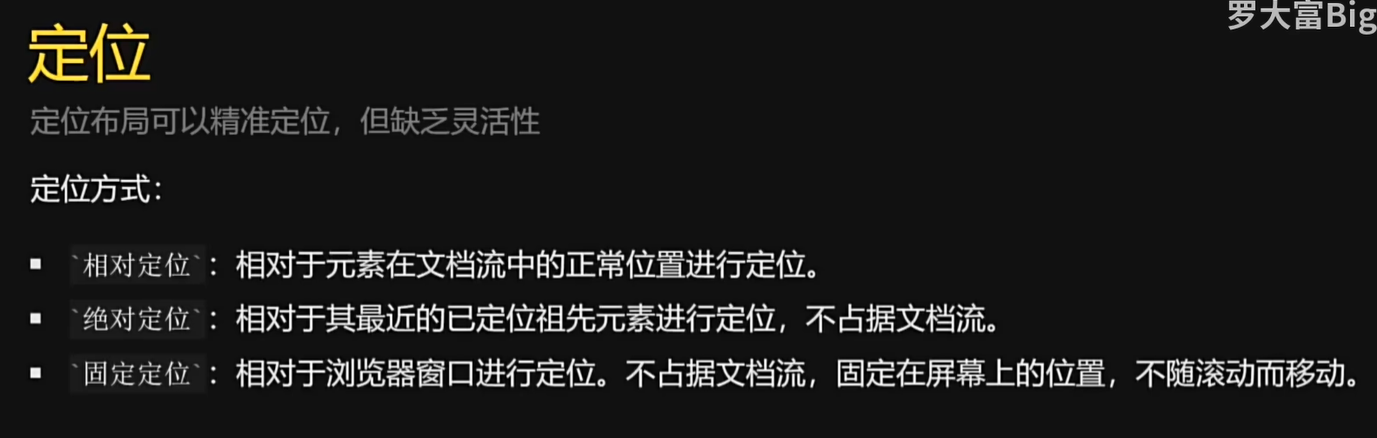
    </div>

</body>

</html>



1. 定位



<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>定位</title>

    <style>

        .box1{

            height: 400px;

            background-color: aqua;

        }

        .box-normal{

            width: 100px;

            height: 100px;

            background-color: blueviolet;

        }

        .box-relative{

            width: 100px;

            height: 100px;

            background-color: yellow;

            position: relative;

            /\* 相对定位，相对于自己原来的位置进行移动,数值代表移动距离 \*/

            /\* 不脱离原本文档流 \*/

            top: 50px;

            left: 130px;

        }

        .box2{

            height: 400px;

            background-color: aqua;

        }

        .box-absolute{

            width: 100px;

            height: 100px;

            background-color: green;

            position: absolute;

            /\* 绝对定位，相对于离他最近的定位元素进行移动,数值代表移动距离 \*/

            /\* 脱离原本文档流 \*/

            left: 150px;

        }

        .box-fixed{

            width: 100px;

            height: 100px;

            background-color: red;

            position: fixed;

            /\* 固定定位，相对于浏览器窗口进行移动,数值代表移动距离 \*/

            /\* 脱离原本文档流 \*/

            right: 0;

            top: 200px;

        }

    </style>

</head>

<body>

    <h1>相对定位</h1>

    <div class="box1">

        <div class="box-normal"></div>

        <div class="box-relative"></div>

        <div class="box-normal"></div>

    </div>

    <br><br>

    <h1>绝对定位</h1>

    <div class="box2">

        <div class="box-normal"></div>

        <div class="box-absolute"></div>

        <div class="box-normal"></div>

    </div>

    <br><br>

    <h1>固定定位</h1>

    <div class="box-fixed"></div>

</body>

</html>





**三、JavaScript**

1. JavaSript的导入

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>JS导入</title>

    <script>

        console.log("JavaScript 内联样式实例2");

    </script>

    <script src="./JS/JS样式.js"></script>

</head>

<body>

    <h1>JavaScript 导入</h1>

    <script>

        /\*  console.log 日志 (快捷键：fn + F12)  \*/

        console.log("JavaScript 内联样式实例1");

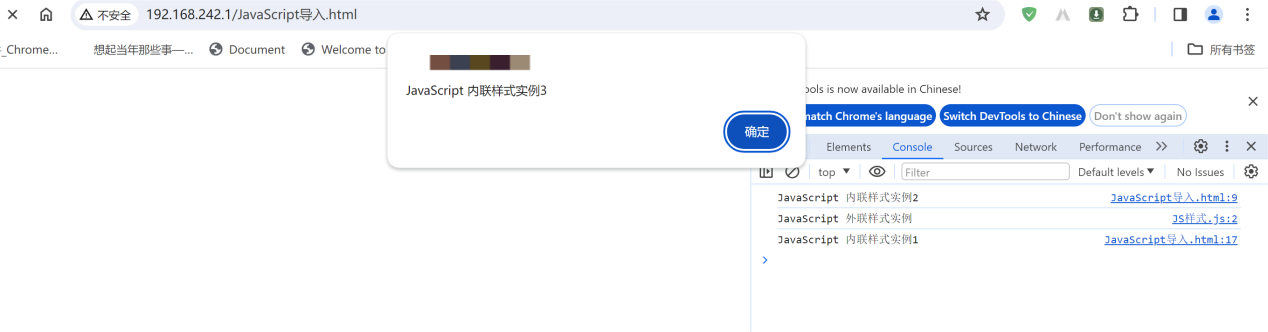
        /\*  alert 弹窗  \*/

        alert("JavaScript 内联样式实例3");

    </script>

</body>

</html>



1. JavaScript 语法

2.1变量与数据类型

<!-- 1.变量与数据类型 -->

    <script>

    //变量var(作用于函数)(不常用)

    var a;

    //变量let(作用于块级元素)

    let b = 5;

    //常量const(作用于块级元素)

    const c = 100;

    //字符串

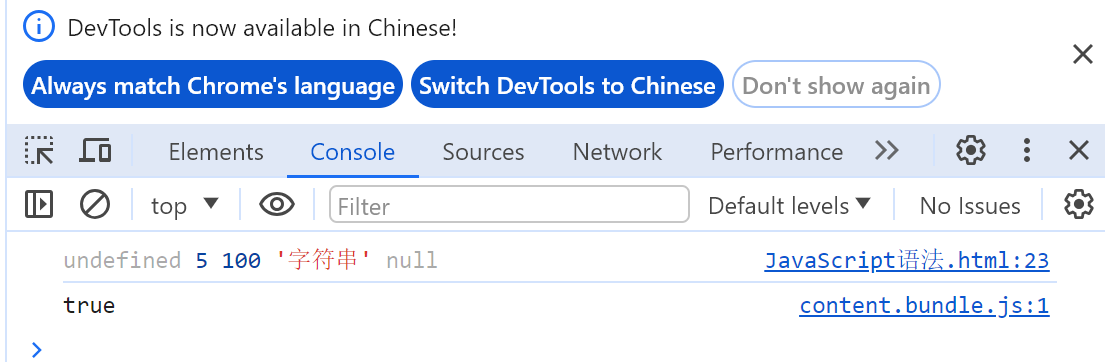
    let s = "字符串"

    //空值

    let empty = null;

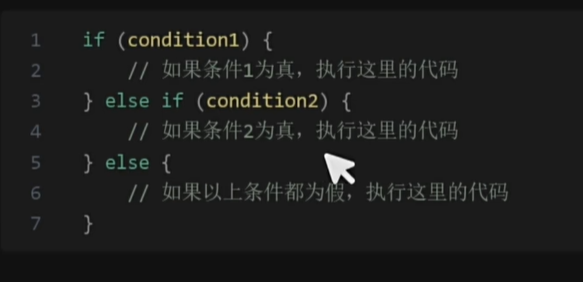
    console.log(a,b,c,s,empty);

</script>

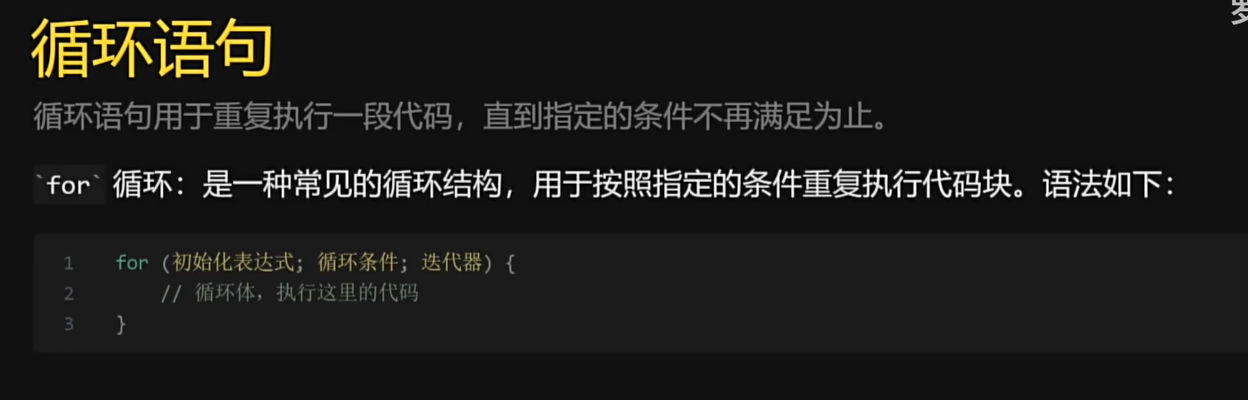


2.2控制语句

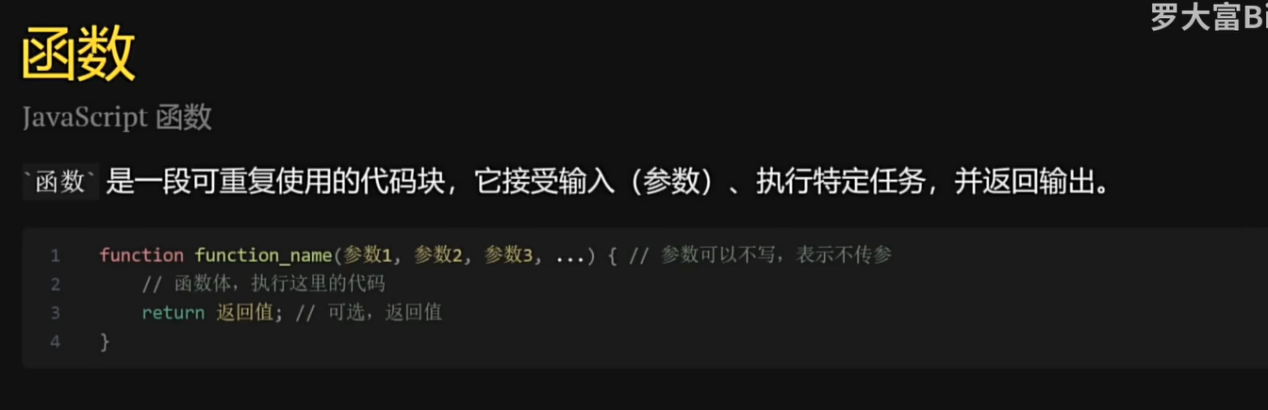
2.2.1条件语句



2.2.2循环语句

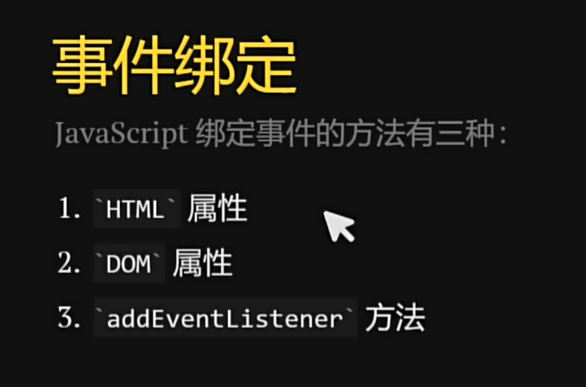


2.3函数



1. JavaScript 事件





* 1. HTML属性

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>事件</title>

</head>

<body>

    <button onclick="clickEvent()">这是一个点击事件按钮</button><br>

    <input type="text" onfocus="focusEvent()" onblur="blurEvent()"><br>

    <input type="text" onmouseover="mouseoverEvent()" onmouseout="mouseoutEvent()">

    <script>

        // 点击事件

        function clickEvent() {

            alert("你点击了按钮");

        }

        // 聚焦事件

        function focusEvent() {

            console.log("你获取了焦点");

        }

        // 失去焦点事件

        function blurEvent() {

            console.log("你失去了焦点");

        }

        // 鼠标移入事件

        function mouseoverEvent() {

            console.log("鼠标移入了");

        }

        // 鼠标移出事件

        function mouseoutEvent() {

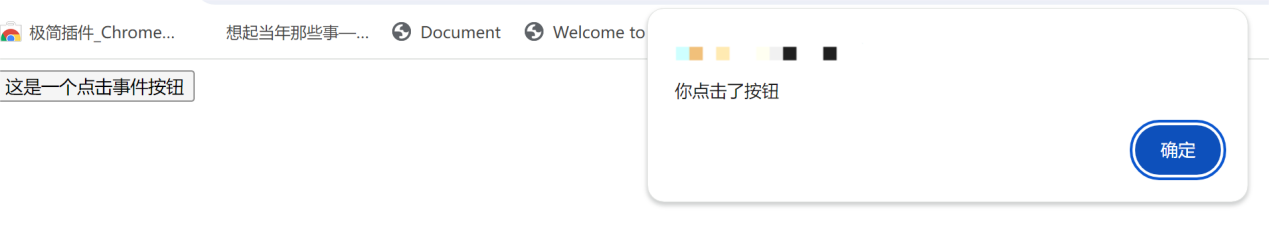
            console.log("鼠标移出了");

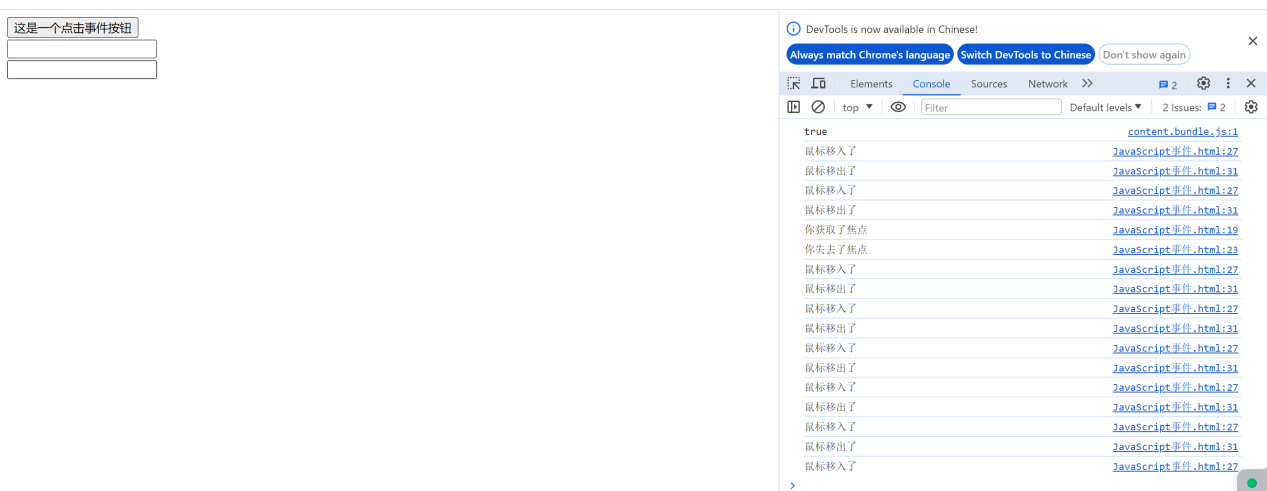
        }

    </script>

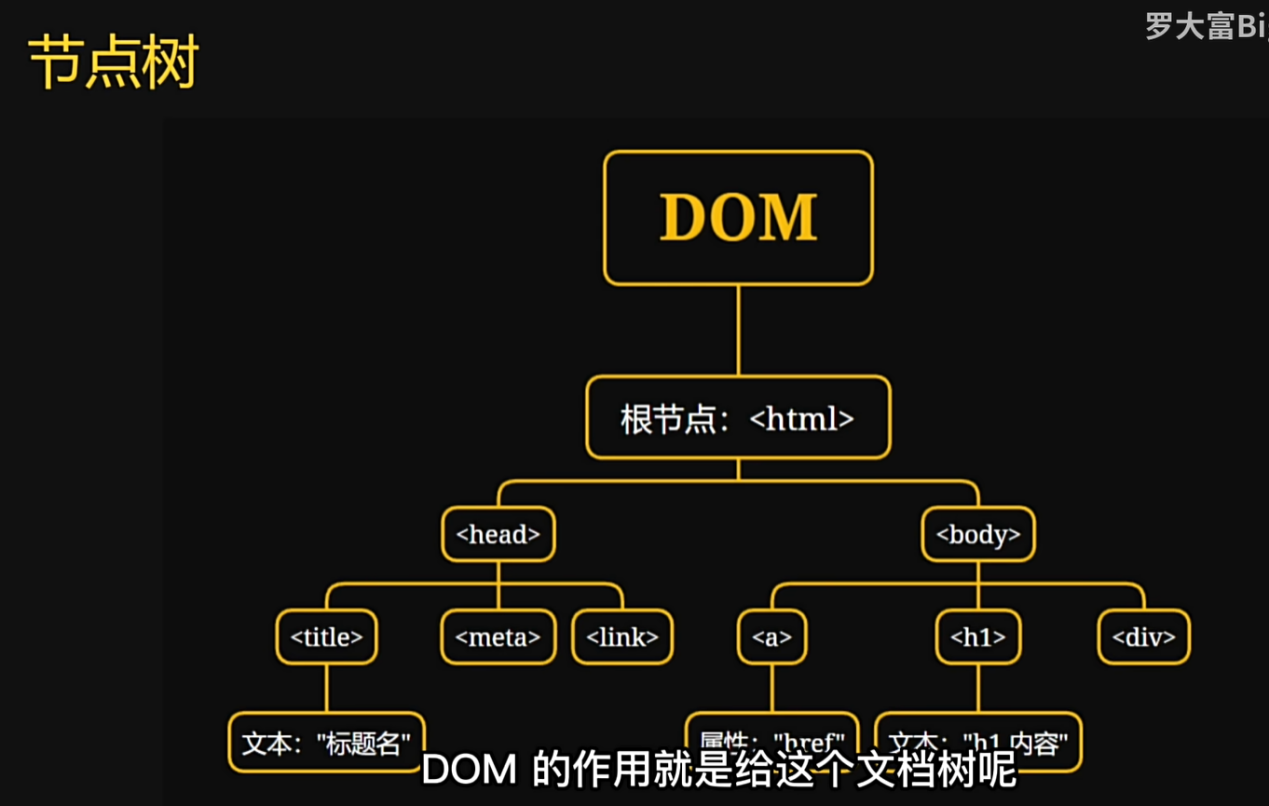
</body>

</html>





1. DOM





<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>DOM</title>

</head>

<body>

    <div id="box1">这是一个ID标签</div>

    <div class="box2">这是一个类标签</div>

    <div>这是一个普通标签</div>

    <button>点击按钮</button>

    <script>

        var element\_id = document.getElementById('box1');

        console.log(element\_id);

        var element\_class = document.getElementsByClassName('box2')[0];

        console.log(element\_class);

        var element\_tag = document.getElementsByTagName('div')[2];

        console.log(element\_tag);

        // 修改ID标签文本内容

        element\_id.innerText = '修改后的ID标签文本内容';

        // 修改类标签文本内容

        element\_class.innerText = '修改后的类标签文本内容';

        // 修改普通标签文本内容

        element\_tag.innerText = '修改后的普通标签文本内容';

        // 修改普通标签属性

        element\_tag.style.color = 'red';

        element\_tag.style.fontSize = '20px';

        // DOM 属性绑定事件

        var button\_element = document.getElementsByTagName('button')[0];

        console.log(button\_element);

        // button\_element.onclick = function() {

        //     alert('DOM 属性按钮被点击了');

        // }

        button\_element.addEventListener('click', function() {

            alert('DOM 属性按钮被点击了');

        })

    </script>

</body>

</html>

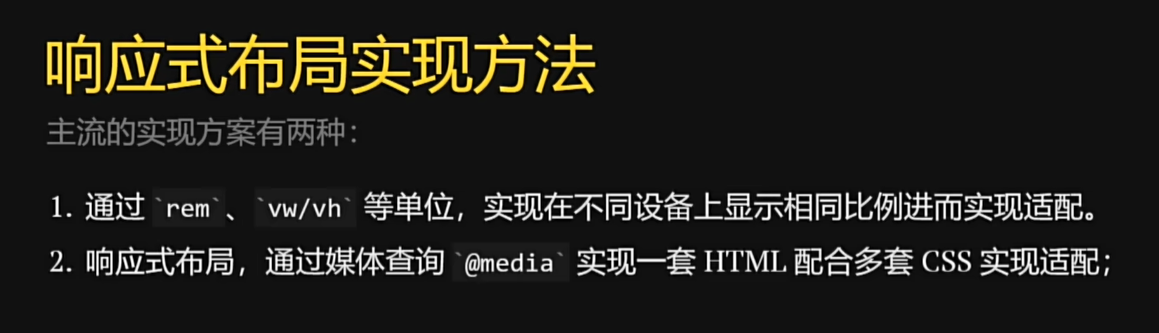


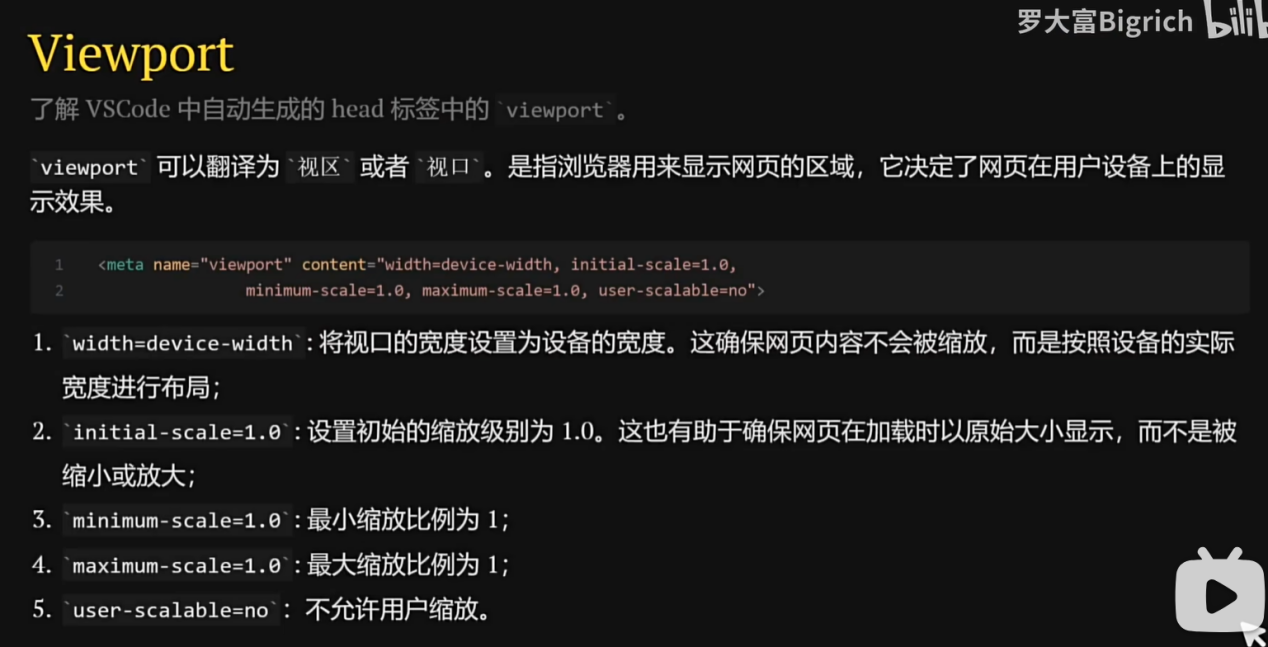
1. 练习——表格的增删改查

F:\nginx-1.25.4\html\练习\_表格的增删改查.html



1. **移动端适配及响应式布局**
2. 响应式布局







<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

    <style>

        /\* 1. px 像素（固定单位） 一般用于pc端网页开发 \*/

        /\* 2. rem 相对单位（相对html的font-size） 一般用于移动端网页开发 \*/

        /\* 3. em 相对单位（相对父元素的字体大小） \*/

        .box-px{

            width: 200px;

            height: 200px;

            background-color: aqua;

            margin-bottom: 30px;

        }

        .box-rem{

            width: 5rem;

            height: 3rem;

            background-color: blueviolet;

        }

    </style>

</head>

<body>

    <div class="box-px"></div>

    <div class="box-rem"></div>

    <script>

        // 根据设备宽度计算 font-size 的属性值，设备宽度/10

        function refreshRem(){

            // var deviceWidth = document.documentElement.clientWidth || window.innerWidth;

            // console.log(deviceWidth);

            document.documentElement.style.fontSize = (screen.width/10) + 'px';

        }

        refreshRem();

        // 绑定事件

        window.onresize = refreshRem;

    </script>

</body>

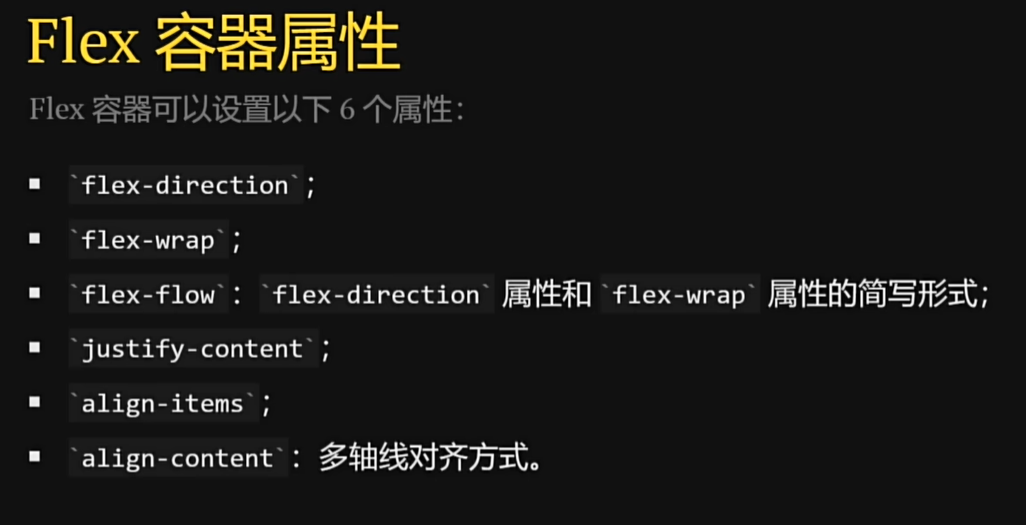
</html>





1. Flex弹性盒子















<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Flex弹性盒子</title>

    <style>

        html {

            font-size: 10px;

        }

        .container {

            background-color: aqua;

            height: 40rem;

            display: flex;

            flex-direction: column;

            flex-wrap: wrap;

            justify-content: space-around;

            align-items: center;

            align-content: space-between;

        }

        .item {

            width: 20rem;

            font-size: 8rem;

        }

    </style>

</head>

<body>

    <div class="container">

        <div class="item" style="background-color: yellow;">1</div>

        <div class="item" style="background-color: red;">2</div>

        <div class="item" style="background-color: blue;">3</div>

        <div class="item" style="background-color: chartreuse;">4</div>

        <div class="item" style="background-color: darkorange;">5</div>

        <div class="item" style="background-color: brown;">6</div>

    </div>

</body>

</html>



1. **后续学习**

