项目课知识点手册

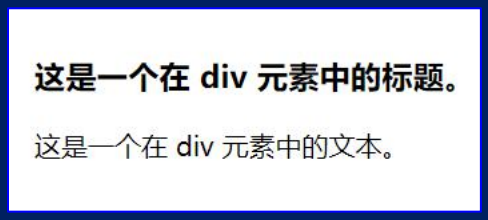
标签： ……

作用：定义文档中的一个区域块(容器)

示例：

```
<div>
    <h3>这是一个在 div 元素中的标题。</h3>
    <p>这是一个在 div 元素中的文本。</p>
</div>
```

示例效果：



标签：
……

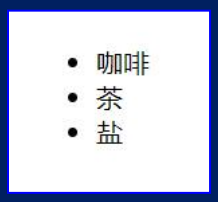
……

作用： 标签定义无序列表， 标签定义列表项目。将 <ul> 标签与 <li> 标签一起使用，创建无序列表。

示例：

```
<ul>
  <li>咖啡</li>
  <li>茶</li>
  <li>盐</li>
</ul>
```

示例效果：



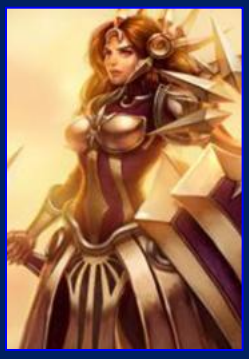
标签：<img>

作用：定义图像

示例：

```
<img src="img/shu.jpg" alt="曙光女神">
```

示例效果：



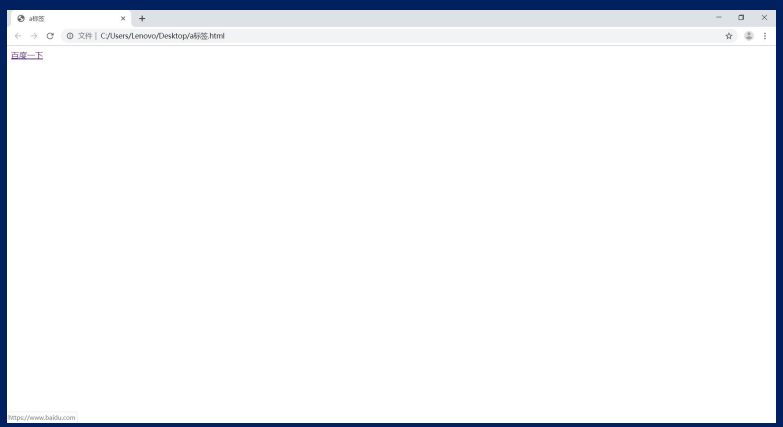
标签：……

作用：定义超链接，链接地址 url，在新窗口中打开

示例：

```
<a href="https://www.baidu.com/" target="_blank">百度一下</a>
```

示例效果：点击”百度一下”，会跳转到百度页面，并在新窗口打开。



---

css 的引入方法：行间样式、内部样式表、外部样式表

行间样式： 直接在标签中设置样式

内部样式表：

```
<head>
        <meta charset="utf-8"/>
        <style type="text/css">
            p{
                color:red;
            }
        </style>
</head>
```

外部样式表：

```
<head>
    <meta charset="utf-8" />
    <link href="style.css" rel="stylesheet" />
</head>
```

---

css 选择器：元素选择器、class 选择器、id 选择器

示例：

```
<div class="c1" id="d1">
        我是一个容器
</div>
```

```
div{ background-color:red;}          /* 元素选择器 */
.c1{background-color:yellow;}        /* class 选择器 */
#d1{background-color:pink;}          /* id 选择器 */
```

属性：width 和 height

作用：设置宽度和高度

示例：

```
<p>这个段落的高和宽都是 100px</p>
p{
    height:100px;
    width:100px;
    background:red;
}
```

示例效果：

属性：background

作用：设置元素背景

示例：

```
<body>
    <h1>背景图片不平铺</h1>
</body>
body{ background:url('img.png')   no-repeat ; }
```

示例效果：

背景图片不平铺

属性：background-size

作用：指定背景图片大小

示例：

```
body{
        background:url('img.png');
        background-size:100% 100%;
        background-repeat:no-repeat;
 }
```

示例效果：

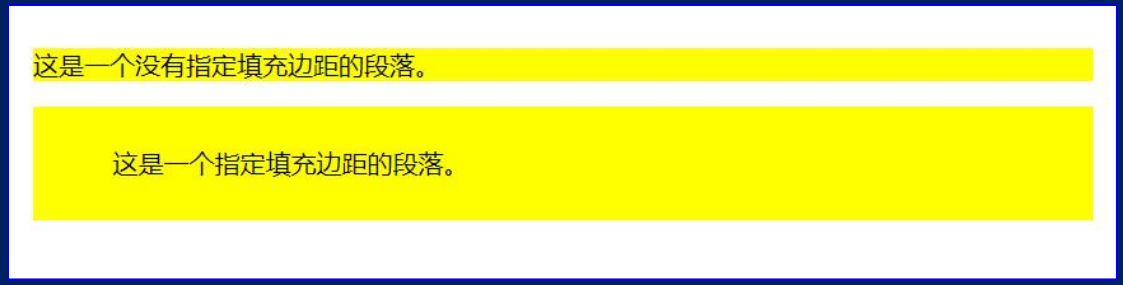
属性：margin

作用：设置外边距(margin-top 上外边距;margin-bottom 下外边距;margin-left 左外边距;margin-right 右外边距)

示例：

```
<p>这是一个没有指定外边距的段落。</p>
<p class="margin">这是一个指定外边距的段落。</p>
p{  background-color:yellow; }
p.margin
{
    margin-top:100px;
    margin-bottom:100px;
    margin-right:50px;
    margin-left:50px;
}
```

示例效果：



常用： margin : 25px 50px ; 表示上下外边距为 25px，左右外边距为

50px。

属性：padding

作用：设置内边距(padding-top 上内边距;padding-bottom 下内边距;padding-left 左内边距;padding-right 右内边距)

示例：

```
<p>这是一个没有指定填充边距的段落。</p>
<p class="padding">这是一个指定填充边距的段落。</p>
p{  background-color:yellow; }
.padding
{
    padding-top:25px;
    padding-bottom:25px;
    padding-right:50px;
    padding-left:50px;
}
```

示例效果：

常用： padding : 25px 50px ; 表示上下内边距为 25px，左右内边距为 50px。

属性：color

作用：设置文字颜色

示例：

```
<p style="color:red;">这段文本是红色</p>
```

示例效果：

这段文本是红色

属性：border

作用：定义元素边框

(border-left 为左边框、border-right 为右边框、border-top 为上边框、border-bottom 为下边框)

示例：

```
<body>
    <p>边框宽度为 5px，边框样式为实线，边框颜色为红色</p>
</body>
p{ border:5px solid red; }
```

示例效果：

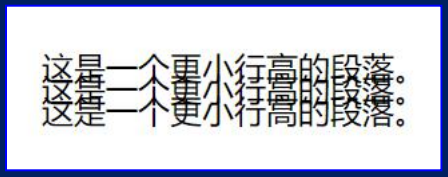
属性：line-height

作用：设置文本行高

示例：

```
<p style="line-height:70%;">
这是一个更小行高的段落。<br>
这是一个更小行高的段落。<br>
这是一个更小行高的段落。<br>
</p>
```

示例效果：



属性：position

作用：指定一个元素定位的类型

值	描述
absolute	生成绝对定位的元素，相对于 static 定位以外的第一个父元素进行定位。 元素的位置通过 "left", "top", "right" 以及 "bottom" 属性进行规定。
fixed	生成固定定位的元素，相对于浏览器窗口进行定位。 元素的位置通过 "left", "top", "right" 以及 "bottom" 属性进行规定。
relative	生成相对定位的元素，相对于其正常位置进行定位。 因此，"left:20" 会向元素的 LEFT 位置添加 20 像素。

示例：

```
<body style="background:red;">
    <h2>这是一个绝对定位了的标题</h2>
</body>
h2
{
    position:absolute;
    left:50px;
    top:50px;
}
```

示例效果：

这是一个绝对定位了的标题

属性：display

作用：规定元素应该生成的框的类型

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）

属性：float

作用：指定一个盒子（元素）是否应该浮动

值	描述
left	元素向左浮动。
right	元素向右浮动。
none	默认值。元素不浮动，并会显示在其在文本中出现的位置。

示例：

```
<body>
    <p>一个段落</p>
    <p class="pra">这是一个段落</p>
</body>
.pra{ float:right; }
```

示例效果：

一个段落

这是一个段落

属性：font-size

作用：设置字体大小

示例：

```
<body>
    <p>这是一个段落</p>
    <p>这是一个段落</p>
</body>
p { font-size:30px; }
```

示例效果：

这是一个段落

这是一个段落

属性：z-index

作用：指定一个元素的堆叠顺序

(拥有更高堆叠顺序的元素总是会处于堆叠顺序较低的元素的前面)

示例：

```
<body>
    <h1>标题</h1>
    <img src="w3.jpg" />
</body>
img{ z-index:-1; }
```

示例效果：

属性：opacity

作用：设置元素的不透明级别，默认值为 1。

示例：

```
<div>本元素的不透明度是 0.5。文本和背景色都受到不透明级别的影响。</div>

div{
    background-color:red;
    opacity:0.5;
}
```

示例效果：

引入 js 文件：<script src="文件名"></script>

说明：可以将脚本放置于 <head> 或者 <body>中。

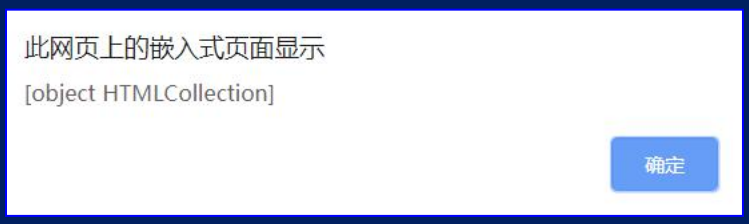
方法：getElementsByClassName( )

作用：返回文档中所有指定类名的元素集合。

示例：

```
<div class="oDiv">JavaScript</div>
<script>
    var x = document.getElementsByClassName("oDiv");
    alert(x);
</script>
```

示例效果：



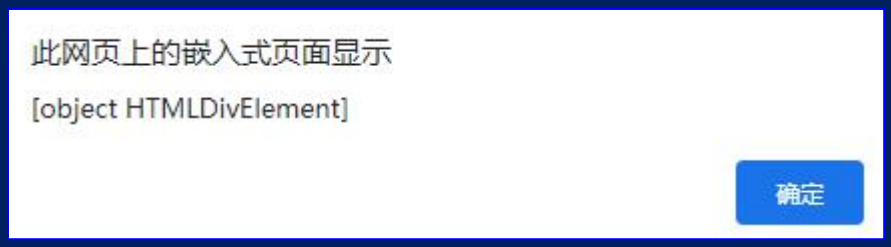
方法：getElementById( )

作用：可返回对拥有指定 ID 的第一个对象的引用。

示例：

```
<div id="oDiv">JavaScript</div>
<script>
    var x = document.getElementById("oDiv");
    alert(x);
</script>
```

示例效果：



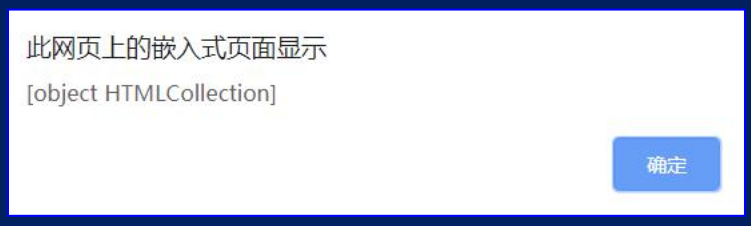
方法：getElementsByTagName( )

作用：返回带有指定标签名的对象的集合

示例：

```
<p>JavaScript</p>
<script>
    var para = document.getElementsByTagName("P");
    alert(para);
</script>
```

示例效果：



函数声明和调用

语法：

声明：

```
function  函数名称(参数){
    执行代码;
}
```

调用：

```
函数名称(参数)
```

---

if……else 语句

语法：

```
if (条件)
{
    当条件为 true 时执行的代码;
}
else
{
    当条件不为 true 时执行的代码;
}
```

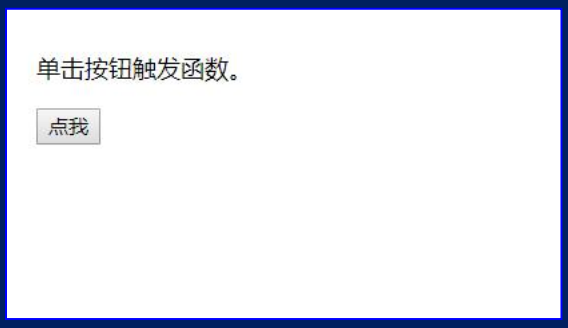
事件：onclick

作用：事件会在元素被点击时发生。

示例：

```
元素对象 .onclick = function(){
    执行代码;
}
```

示例效果：



JavaScript 更改 html 元素的样式

语法：

```
元素对象 .style . 要修改的样式 = 新的样式
```

示例：

```
<p id="p1">Hello World!</p>
<p id="p2">Hello World!</p>
<script>
    document.getElementById("p2").style.color = "blue";
    document.getElementById("p2").style.fontFamily = "Arial";
    document.getElementById("p2").style.fontSize = "larger";
</script>
```

示例效果：

Hello World!

Hello World!

声明数组：var a=[1,2,3];

作用：通过中括号声明数组。如果定义空数组则为 var a=[ ];

方法：for 循环

作用：可以将代码块执行指定的次数

语法：

```
for (语句 1; 语句 2; 语句 3)
{
    被执行的代码块
}
```

语句 1 （代码块）开始前执行

语句 2 定义运行循环（代码块）的条件

语句 3 在循环（代码块）已被执行之后执行

方法：getComputedStyle()

作用：用于获取指定元素的 CSS 样式。

示例：

```
<p>这是一个段落</p>
<style>
    p{color:red;}
</style>
<script>
     var pq = document.getElementsByTagName("p")[0];
    var a = getComputedStyle(pq).color;
    alert(a);
</script>
```

示例效果：