

# 项目课知识点手册

- 1、[<div>标签\(第 2 页\)](#)
- 2、[<img>标签\(第 2 页\)](#)
- 3、[css 引入方法\(第 3 页\)](#)
- 4、[css 选择器\(第 4 页\)](#)
- 5、[width 和 height 属性\(第 4 页\)](#)
- 6、[background 属性\(第 5 页\)](#)
- 7、[background-size 属性\(第 6 页\)](#)
- 8、[margin 属性\(第 7 页\)](#)
- 9、[z-index 属性\(第 8 页\)](#)
- 10、[position 属性\(第 8 页\)](#)
- 11、[overflow 属性\(第 10 页\)](#)
- 12、[cursor 属性\(第 12 页\)](#)
- 13、[类名获取元素\(第 12 页\)](#)
- 14、[控制台打印信息\(第 13 页\)](#)
- 15、[定时器\(第 14 页\)](#)
- 16、[函数声明和调用\(第 14 页\)](#)
- 17、[if.....else 语句\(第 15 页\)](#)
- 18、[onclick 事件\(第 16 页\)](#)
- 19、[js 更改元素样式\(第 16 页\)](#)
- 20、[offsetLeft 偏移量\(第 17 页\)](#)

---

标签: `<div>.....</div>`

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:

```
这是一个在 div 元素中的标题。

这是一个在 div 元素中的文本。
```

---

标签: `<img>`

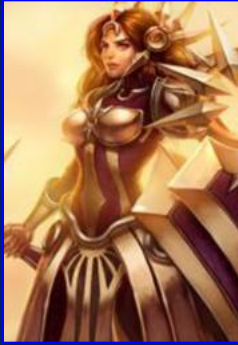
作用: 定义图像

示例:

```

```

示例效果:



---

css 的引入方法：行间样式、内部样式表、外部样式表

行间样式：<p style="color:green;">直接在标签中设置样式</p>

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

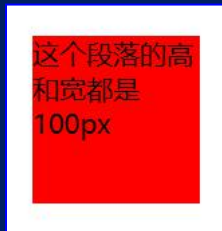
作用: 设置宽度和高度

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:

这是一个没有指定填充边距的段落。

这是一个指定填充边距的段落。

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

50px。

---

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>

</body>

img{ z-index:-1; }
```

示例效果:



属性: **position**

作用: 指定一个元素定位的类型



值	描述
<a href="#">absolute</a>	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。
<a href="#">fixed</a>	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。
<a href="#">relative</a>	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: `overflow`

作用: 规定当内容溢出元素框时发生的事情

值	描述
visible	默认值。内容不会被修剪，会呈现在元素框之外。
hidden	内容会被修剪，并且其余内容是不可见的。
scroll	内容会被修剪，但是浏览器会显示滚动条以便查看其余的内容。
auto	如果内容被修剪，则浏览器会显示滚动条以便查看其余的内容。
inherit	规定应该从父元素继承 <code>overflow</code> 属性的值。

示例:

```
<div>
```

这个属性定义溢出元素内容区的内容会如何处理。如果值为 `scroll`，不论是否需要，用户都会提供一种滚动机制。如果值为 `hidden`，则内容会被修剪，并且其余内容是不可见的。

```
</div>
```

```
div
```

```
{
```

```
    background-color:#00FFFF;
```

```
    width:150px;
```

```
    height:150px;
```

```
    overflow: hidden;
```

```
}
```

示例效果:

这个属性定义溢出元素内容区的内容会如何处理。如果值为 `scroll`，不论是否需要，用户都会提供一种滚动机制。如果值为 `hidden`，则内容会

属性: **cursor**

作用: 定义了鼠标指针放在一个元素边界范围内时所用的光标形状。

示例:

```
<body>

    <span style="cursor:pointer">pointer</span>

</body>
```

示例效果:



---

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:

此网页上的嵌入式页面显示  
[object HTMLCollection]

确定

---

方法: `console.log()`

作用: 用于在控制台输出信息

示例:

```
<p>按下 F12 查看控制台信息。</p>

<script>

    a = 0;

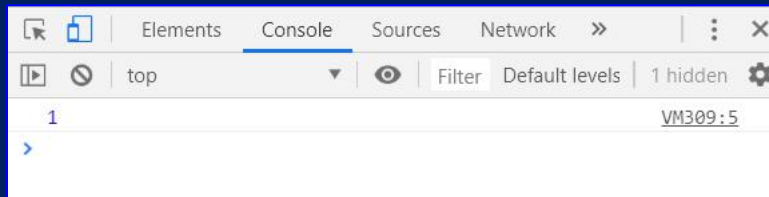
    b = 1;

    c = a + b;

    console.log(c);

</script>
```

示例效果：



---

方法： `setInterval()` 方法

作用：按照指定的周期（以毫秒计）来调用函数或计算表达式

语法：

```
setInterval(code, milliseconds);
```

第一个参数 code：要调用一个代码串，也可以是一个函数。

第二个参数 milliseconds：周期性执行或调用 code/function 之间的时间间隔，以毫秒计。

---

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:

单击按钮触发函数。

点我

---

## 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!

---

知识点： `offsetLeft` 属性可以获取元素相对于顶点的水平位置偏移量。

示例：

```
<div id="box"> </div>

<script>

    var box = document.getElementById("box");

    var l = box.offsetLeft;

    console.log(l);

</script>
```

示例效果：

