

HTML+CSS学习笔记

HTML笔记

1、初识HTML

- HyperTextMarkupLanguage（超文本标记语言）
- < body >、< /body>等成对的标签，分别叫做开放标签和闭合标签，
- 单独呈现的标签（空元素），如< hr />;意为用/来关闭空元素。
- html注释：<!--注释内容-->

```
<!--DOCTYPE: 告诉浏览器使用什么规范（默认是html）-->
<!DOCTYPE html>
<!-- 语言 zh中文 en英文-->
<html lang="zh">
<!--head标签代表网页头部-->
<head>
<!-- meta 描述性标签，表示用来描述网站的一些信息-->
<!-- 一般用来做SEO-->
    <meta charset="UTF-8">
    <meta name="keywords" content="hyx的java学习">
    <meta name="description" content="一起来学习java">
    <!-- 网站标题-->
    <title>Title</title>
</head>
<!--body代表主体-->
<body>
Hello World!
</body>
</html>
```

2、网页基本标签

- 标题标签
- 段落标签
- 换行标签
- 水平线标签
- 字体样式标签
- 注释和特殊符号

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>基本标签</title>
</head>
<body>
<h1>一级标签</h1><!-- 标题标签-->
<h2>二级标签</h2>
<h3>三级标签</h3>
<h4>四级标签</h4>
<h5>五级标签</h5>
<h6>六级标签</h6>

<!-- 段落标签-->
<p>p换行1</p>
<p>p换行2</p>

<hr/><!-- 水平线标签-->
```

```

换行1 <br/><!--换行标签-->
换行2 <br/>
<!--换行标签比较紧凑，段落标签有明显段间距-->
<!--粗体 斜体-->
<h1>字体样式标签</h1>
粗体: <strong>I love you </strong><br/>
斜体: <em>I love you </em><br/>
<!--特殊符号-->
空格: 1&nbsp;2&nbsp;3&nbsp;4<br/>
空格: 1 2 3 4<br/>
大于号&gt;<br/>
小于号&lt;<br/>
版权符号&copy;<br/>
<!--特殊符号记忆: &开头;结尾，只要在idea中&敲出后就有提示-->
</body>
</html>

```

- 图像标签

- 链接标签

href: 必填，表示要跳转到那个页面

target: 表示窗口在那里打开

_blank: 在新标签中打开

_self: 在自己的网页中打开

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>图像和链接标签</title>
</head>
<body>
<!--src:资源地址
  相对地址，绝对地址
  ../上级地址
  alt: 在图片加载失败的时候，就会用文字代替
  title: 鼠标悬停在图片上时，所显示的名字
  width height 图片的高和宽-->

<br/>
<!--href: 跳转页面的地址
  a标签内文字: 即显示的文字
  可以把图片放在a标签内，点击图片跳转网页
  target: 表示在哪打开新网页_self: 当前标签打开 _blank: 新的页面中打开-->
<a href="https://www.baidu.com" target="_blank" title="123">你xxx不会百度吗</a>
<br/>
<a href="https://www.baidu.com"></a>

<!-- 锚链接
  1. 需要一个标记锚
  2. 跳转到标记
  #页面内跳转-->
<a name="top"></a>
<a href="#top">回到顶部</a>
<br/>
<!-- 也可以在网址后添加#号跳到对应网站的对应位置-->
<a href="https://www.baidu.com#down">百度底部</a> <br/>

<!-- 功能性链接
邮箱链接: mailto
qq链接
-->
<a href="mailto:xxxxxxqq.com">点击联系我</a>
<a target="_blank"
  href="http://wpa.qq.com/msgrd?v=xxx&uin=&site=qq&menu=yes"/>
  <a target="_blank"

```

```
href="http://wpa.qq.com/msg?rd?v=xxx&uin=&site=qq&menu=yes">
  
</body>
</html>
```

行内元素和块元素

- 块元素
 - 无论内容多少，该元素独占一行

例如：<p></p><hr/> <h1>...<h6>

- 行内元素：内容撑开宽度，左右都是行内元素的可以排在一行

例如：<a>

3、列表标签

什么是列表

列表就是信息资源的一种展示形式。它可以使信息结构化和条理化，并以列表的样式显示出来，以便浏览者能更快捷地获得相应的信息。

列表的分类：

- 无序列表

```
<ol>
  <li>1</li>
  <li>2</li>
  <li>3</li>
</ol>
```

- 有序列表

```
<ul><!-- 无序列表 -->
  <li>123
    <ul>
      <li>1</li>
      <li>2</li>
      <li>3</li>
    </ul>
  </li>
  <li>2</li>
  <li>3</li>
</ul>
```

- 定义列表

```
<!-- 自定义列表
dl: 标签
dt: 列表名称
dd: 列表内容 -->
<dl>
  <dt>学科</dt>
  <dd>语文</dd>
  <dd>数学</dd>
  <dd>英语</dd>
  <dt>语言</dt>
```

```
<dd>中文</dd>
<dd>英语</dd>
<dd>日语</dd>
</dl>
```

4、表格

表格的基本结构：

- 单元格
- 行
- 列
- 跨行
- 跨列

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>表格</title>
</head>
<body>
  <!-- 表格table
  行 tr
  列 td
  -->
  <table border="1px">
    <tr><!-- 跨列 -->
      <td colspan="3">1-1</td>
    </tr>
    <tr><!-- 跨行 -->
      <td rowspan="2">2-1</td>
      <td>2-2</td>
      <td>2-3</td>
    </tr>
    <tr>
      <td>3-2</td>
      <td>3-3</td>
    </tr>
  </table>
</body>
</html>
```

5、视频和音频

- video
- audio

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>媒体元素</title>
</head>
<body>
  <!-- 视频
      src 资源路径
      controls 控制面板
      autoplay 自动播放
  -->
  <video src="xxx/xxx/xxx" controls autoplay></video>
  <!-- 音频 -->
  <audio src="xxx/xxx/xxx" controls autoplay></audio>
</body>
</html>
```

6、页面结构

元素名	描述
header	标题头部区域的内容（用于页面或者页面中的一块区域）
footer	标记脚部区域的内容（用于整个页面或页面的一块区域）
section	Web页面中的一块独立区域
article	独立的文章内容
aside	相关内容或应用
nav	导航类辅助内容

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>页面结构</title>
</head>
<body>
<!-- 页面头部 -->
<header>
  <h2>网页头部</h2>
</header>
<section>
  <h2>网页主体</h2>
</section>
<footer>
  <h2>网页脚部</h2>
</footer>
</body>
</html>
```

7、iframe内联框架

```
<iframe src="path" name="mainFrame" ></iframe>
```

- ifram标签，必须要有src属性即引用页面的地址
- 给标签加上name属性后，可以做a标签的target属性，即在内联窗口中打开链接

8、表单语法(重点)

from标签，action属性为所提交的目的地址，method选择提交方式
可以选择使用post或者get方式提交

- get效率高，但在url中可以看到提交的内容，不安全，不能提交大文件
- post比较安全且可以提交大文件

标签	说明
input标签	大部分表单元素对应的标签有text、password、checkbox、radio、submit、reset、file、hidden、image和button，默认为text，可以提交用户名、密码
select标签	下拉选择框
textarea标签	文本域

属性	说明
type	指定元素的类型。text、password、checkbox、radio、submit、reset、file、hidden、image和button，默认为text
name	指定表单元素的名称（提交时所对应的key）


```

<p><!-- 文本域 -->
    反馈:
    <textarea name="text" id="10" cols="30" rows="10" >文本内容</textarea>
</p>
<p><!-- 文件域 -->
    <input type="file" name="files">
    <input type="button" name="upload" value="上传">
</p>
<!-- 邮件: 会简单验证是否是邮箱地址
    url: 会简单验证是否是网络地址
    number: 数字验证 -->
<p>邮箱: <input type="email" name="email">
    url: <input type="url"></p>

<!-- 数字验证
    max 最大数量
    min 最小数量
    step 每次点击增加或减少的数量 -->
<p>商品数量<input type="number" name="num" max="100" min="1" step="1"></p>
<!-- 滑块 -->
<p>音量: <input type="range" min="0" max="100" name="voice" step="2"></p>
<!-- 搜索框 -->
<p>搜索: <input type="search"></p>

<p><!-- 增强鼠标可用性 -->
    <label for="mark">你点我试试</label>
    <input type="text" id="mark">
</p>
</form>
</body>
</html>

```

CSS3学习

1.css是什么

2.CSS怎么用（快速入门）

3.CSS选择器（重点 + 难点）

4.美化页面（文字、阴影、超链接、列表、渐变...）

5.盒子模型

6.浮动

7.定位

8.网页动画（特效）

1.什么是CSS

1.1、什么是CSS

Cascading Style Sheet 层叠样式表

CSS：表现（美化网页）

字体，颜色，边距，高度，宽度，背景图片，网页定位，网页浮动

1.2、发展史

CSS1.0

CSS2.0: DIV（块）+CSS，HTML与CSS结构分离的思想，网页变得简单，SEO

CSS2.1: 浮动，定位

CSS3.0: 圆角、阴影、动画...浏览器兼容性~

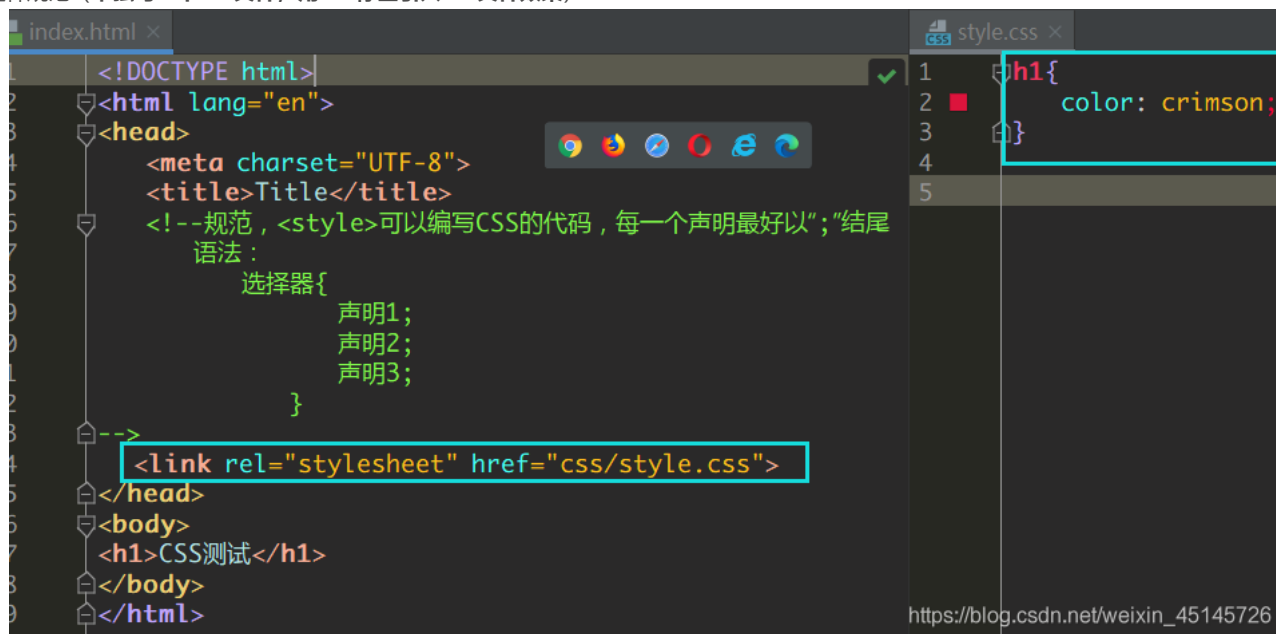
2、快速入门

2.1.1、练习格式：



```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
  <!-- 规范，<style>可以编写CSS的代码，每一个声明最好以";"结尾
  语法：
    选择器{
      声明1;
      声明2;
      声明3;
    }
```

建议使用这种规范（单独写一个css文件，用link标签引入css文件效果）：



2.1.2、CSS的优势：

- 1、内容和表现分离；
- 2、网页结构表现统一，可以实现复用
- 3、样式十分的丰富
- 4、建议使用独立于html的css文件
- 5、利用SEO，容易被搜索引擎收录！

2.1.3、CSS的3种常用导入方式

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>

  <style><!-- 内部样式 -->
    h1{
      color: green;
    }
  </style>
```




拓展：外部样式两种方法

- 链接式

html

```
<!-- 外部样式 -->
<link rel="stylesheet" href="css/style.css" />
```

- 导入式

@import是CSS2.1特有的!

```
<!-- 导入式 -->
<style>
  @import url("css/style.css");
</style>
```

2.2选择器

作用：选择页面上的某一个或者某一类元素

2.2.1、基本选择器

1、标签选择器

选择一类标签

格式： 标签 {}

```
<head>
  <meta charset="UTF-8">
  <title>Title</title>
  <style>
    h1{
      color: orange;
      background: blue;
      border-radius: 10px;
    }
    h3{
      color: orange;
```



2、类选择器class

选择所有class一致的标签，跨标签

格式： .类名 {}

```
<head>
  <meta charset="UTF-8">
  <title>Title</title>
  <style>
    /*类选择器的格式 .class的名称{}
    好处：可以多个标签归类，是同一个class，可以复用*/
    .demo1{
      color: blue;
    }
    .demo2{
      color: red;
```

3、id 选择器

全局唯一

格式：#id名{}
↓

```
<head>
  <meta charset="UTF-8">
  <title>Title</title>
  <style>
    /*id选择器: id必须保证全局唯一
    #id名称{}
    不遵循就近原则，优先级是固定的
    id选择器 > class类选择器 > 标签选择器
    */
    #demo1{
      color: red;
    }
  </style>
</head>
```

优先级: id > class > 标签
↓

2.2.2、高级选择器

1. 层次选择器

- 后代选择器: 在某个元素的后面

```
/*后代选择器*/
<style>
body p{
  background:red;
}
</style>
```

- 子选择器, 一代

```
/*子选择器*/
<style>
body>p{
  background:orange;
}
</style>
```

- 相邻的兄弟选择器 同辈

```
/*相邻兄弟选择器: 只选择一个, 相邻(向下)*/
<style>
.active+p{
background: red
}
</style>
<body>
  <p class="active">p1<p>
  <p>p2</p>
</body>
```

- 通用选择器

```

<style>
/*通用兄弟选择器，当前选中元素的向下的所有兄弟元素*/
.active~p{
    background:red;
}
</style>
<body>
    <p class="active">p1<p>
    <p>p2</p>
</body>

```

2.结构伪类选择器

伪类

```

<style>
    ul li:first-child{/*ul的第一个子元素*/
        background: aqua;
    }
    ul li:last-child{/*ul的最后一个子元素*/
        background: blue;
    }
/*选中p1：定位到父元素，选择当前的第一个元素
    选择当前p元素 的父级元素，选中父级元素的第一个，并且是当前元素才生效! */
p:nth-child(1){
    background: orange;
}

```



3.属性选择器（常用）

id + class结合

```

<head>
<meta charset="UTF-8">
<title>Title</title>
<style>
    .demo a{
        float: left;
        display: block;
        height: 50px;
        width: 50px;
        border-radius: 10px;
        background: aquamarine;
    }

```



效果：



3、美化网页元素

3.1、为什么要美化网页

1. 有效的传递页面信息
2. 美化网页，页面漂亮才能吸引客户
3. 凸显页面的主题
4. 提高用户的体验

span标签：重点要突出的字，使用span标签套起来

```
<head>
  <meta charset="UTF-8">
  <title>Title</title>
  <style>
    #title1{
      font-size: 50px;
    }
  </style>
</head>
<body>
  学习语言<span id="title1">JAVA</span>
</body>
```

3.2、字体样式

- font-family: 字体
- font-size: 字体大小
- font-weight: 字体粗细
- color: 字体颜色

```
<head>
  <meta charset="UTF-8">
  <title>Title</title>
  <style>
    body{
      font-family:"Arial Black", 楷体;
      color: red;
    }
    h1{
      font-size: 50px;
    }
  </style>
</head>
<body>
  学习语言
</body>
```



常用写法：

```
font-weight:bold; /*也可以填px，但不能超过900，相当于bolder*/
/*常用写法：*/
font:oblique bolder 12px "楷体"
```

3.3、文本样式

1. 颜色→color:rgb / rgba()
2. 文本对齐方式→text-align: center
3. 首行缩进→text-indent: 2em
4. 行高→line-height: 300px;
5. 下划线→text-decoration

```
text-decoration:underline/*下划线*/
text-decoration:line-through/*中划线*/
text-decoration:overline/*上划线*/
text-decoration:none/*超链接去下划线*/
```

1. 图片、文字水平对齐

```
img,span{vertical-align:middle}
```

3.4、文本，阴影和超链接伪类

```
<style>
  a{/*超链接有默认的颜色*/
    text-decoration:none;
    color:#000000;
  }
  a:hover{/*鼠标悬浮的状态*/
    color:orange;
  }
  a:active{/*鼠标按住未释放的状态*/
    color:green
  }
}
```

阴影：

```
/* 第一个参数：表示水平偏移
   第二个参数：表示垂直偏移
   第三个参数：表示模糊半径
   第四个参数：表示颜色
*/
text-shadow:5px 5px 5px 颜色
```

3.6、列表ul li

主页index.html代码：

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
  <link href="css/style.css" rel="stylesheet" type="text/css">
</head>
<body>
<div id="nav">
  <h2 class="title">全部商品分类</h2>
  ...
</div>
```

css代码：

```
#nav{
  width: 300px;
  background: antiquewhite;
}
.title{
  font-size: 18px;
  font-weight: bold;
  text-indent: 1em;/*缩进*/
  line-height: 35px;
  background: red;
}
```

3.7、背景

1. 背景颜色: background
2. 背景图片

```
background-image:url("");/*默认是全部平铺的*/
background-repeat:repeat-x/*水平平铺*/
background-repeat:repeat-y/*垂直平铺*/
background-repeat:no-repeat/*不平铺*/
```

3.8、渐变

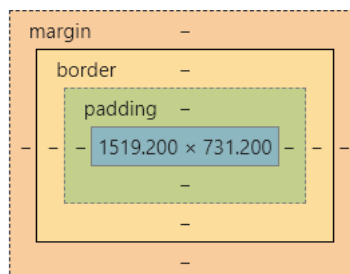
渐变背景网址: <https://www.grabient.com>

径向渐变、圆形渐变

```
body{
    background-color: #4158D0;
    background-image: linear-gradient(43deg, #4158D0 0%, #C850C0 46%, #FFCC70 100%);
}
```

4、盒子模型

4.1什么是盒子模型



1. margin: 外边距
2. padding: 内边距
3. border: 边框

4.2、边框

border: 粗细 样式 颜色

1. 边框的粗细
2. 边框的样式
3. 边框的颜色

4.3、外边距----妙用: 居中

margin-left/right/top/bottom→表示四边, 可分别设置, 也可以同时设置如下

```
margin:0 0 0 0/*分别表示上、右、下、左; 从上开始顺时针*/
/*例1: 居中*/
margin:0 auto /*auto表示左右自动*/
/*例2: */
margin:4px/*表示上、右、下、左都为4px*/
/*例3*/
margin:10px 20px 30px/*表示上为10px, 左右为20px, 下为30px*/
```

盒子的计算方式:

margin+border+padding+内容的大小

总结:

body总有一个默认的外边距 margin:0

常见操作: 初始化

4.4、圆角边框----border-radius

```
<style>
    div{
        width: 100px;
        height: 100px;
        border: 10px solid red;
        /*一个border-radius只管一个圆的1/4*/
        border-radius: 50px 20px 20px 30px; /*左上 右上 右下 左下 ,顺时针方向*/
    }
</style>
```

4.5、盒子阴影

```
box-shadow: 10px 10px 1px black;
```

5、浮动

5.1标准文档流

块级元素：独占一行 h1~h6、p、div、列表...

行内元素：不独占一行 span、a、img、strong

注：行内元素可以包含在块级元素中，反之则不可以

5.2、display（重要）

1. block：块元素
2. inline：行内元素
3. inline-block：是块元素，但是可以内联，在一行

这也是一种实现行内元素排列的方式，但是我们很多情况用float

1. none：消失

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
    <!--block 块元素
        inline 行内元素
        inline-block 是块元素，但是可以内联，在一行
    -->
    <style>
        div{
```



5.3、float: left/right左右浮动

clear: both

5.4、overflow及父级边框塌陷问题

clear:

right: 右侧不允许有浮动元素

left: 左侧不允许有浮动元素

both: 两侧不允许有浮动元素

none:

解决塌陷问题方案：

方案一：增加父级元素的高度；

方案二：增加一个空的div标签，清除浮动

```
<div class = "clear"></div>
<style>
    .clear{
        clear:both;
        margin:0;
        padding:0;
    }
</style>
```

方案三：在父级元素中增加一个overflow属性

```
overflow:hidden/*隐藏超出部分*/
overflow: scoll/*滚动*/
```

方案四：父类添加一个伪类:after

```
#father:after{
    content:'';
    display:block;
    clear:both;
}
```

小结：

1. 浮动元素增加空div----> 简单、代码尽量避免空div
2. 设置父元素的高度-----> 简单，但是元素假设有了固定的高度，可能就会超出范围
3. overflow-----> 简单，下拉的一些场景避免使用
4. 父类添加一个伪类:after（推荐）----> 写法稍微复杂，但是没有副作用，**推荐使用**

display与float对比

1. display：方向不可以控制
2. float：浮动起来的话会脱离标准文档流，所以要解决父级边框塌陷的问题。

6、定位

6.1、相对定位

相对定位：**positon: relstive;**

相对于原来的位置，进行指定的偏移，相对定位的话，它仍然在标准文档流中！原来的位置会被保留

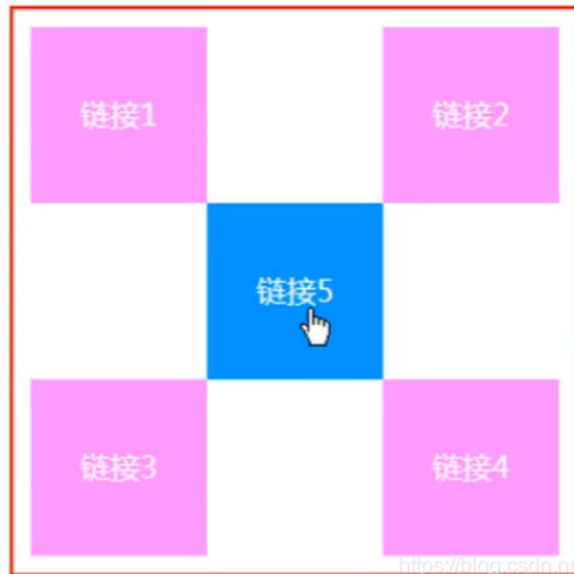
```
top:-20px;/*向上偏移20px*/
left:20px;/*向右偏移20px*/
bottom:10px;/*向上偏移10px*/
right:20px;/*向左偏移20px*/
```

代码：

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>相对定位</title>
    <!-- 相对定位
        相对于自己原来的位置进行偏移
    -->
<style>
    body{
        padding: 20px;
```


练习:

- 使用<div>和超链接<a>布局页面
- 每个超链接宽度和高度都是100px，背景颜色是粉色，鼠标指针移上去时变为蓝色
- 使用相对定位改变每个超链接的位置



https://blog.csdn.net/weixin_45145726

实现代码:

```
<style>
    #box{
        height: 300px;
        width: 300px;
        border: 2px red solid;
        padding: 10px;
    }
    a{
        height: 100px;
        width: 100px;
        background-color: #e673b7;
    }
```

6.2、绝对定位-absolute和固定定位-fixed

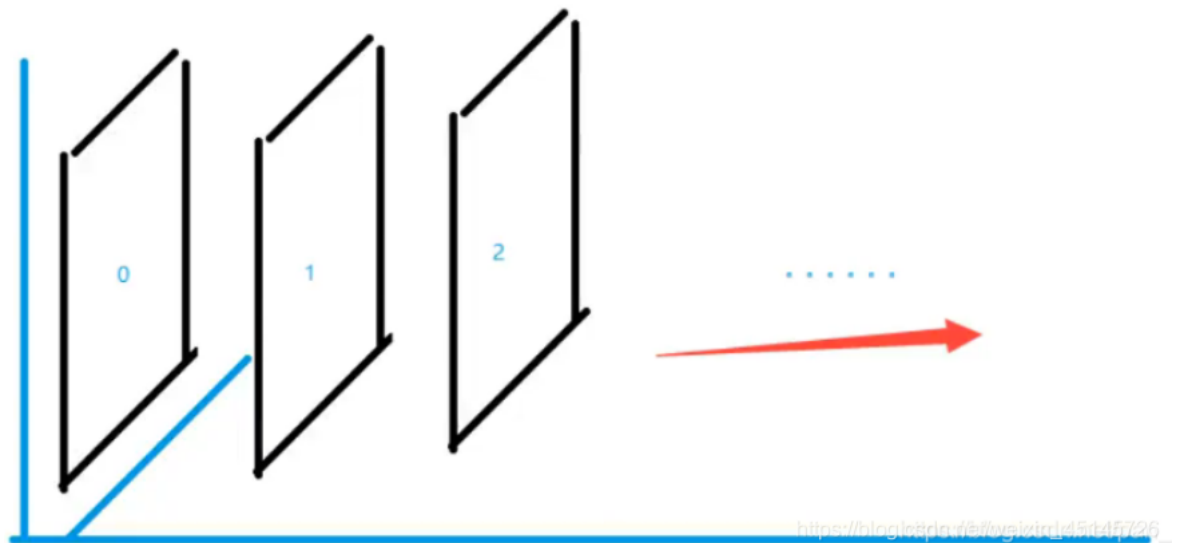
定位: 基于xxx定位, 上下左右~

- 1、没有父级元素定位的前提下, 相对于浏览器定位
- 2、假设父级元素存在定位, 我们通常会相对于父级元素进行偏移
- 3、在父级元素范围内移动

总结: 相对一父级或浏览器的位置, 进行指定的偏移, 绝对定位的话, 它不在标准文档流中, 原来的位置不会被保留

```
<head>
    <meta charset="UTF-8">
    <title>Title</title>
    <style>
        body{
            height: 1000px;
        }
        div:nth-of-type(1){
            width: 100px;
            height: 100px;
        }
```

6.3、z-index



图层-z-index：默认是0，最高无限~999

index.html代码：

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
  <link rel="stylesheet" href="css/style.css" type="text/css">
  <style></style>
</head>
<body>
<div id="content">
  ...
</div>
```

css代码：

```
#content{
  width: 380px;
  padding: 0px;
  margin: 0px;
  overflow: hidden;
  font-size: 12px;
  line-height: 25px;
  border: 1px solid yellow;
}
ul,li{
  padding: 0px;
  margin: 0px;
  list-style: none;
}
/*父级元素相对定位*/
#content ul{
  position: relative;
}
.tipText,.tipBg{
  position: absolute;
  width: 380px;
```

```
        height: 25px;
        top:216px
    }
    .tipText{
        color: white;
        z-index: 999;
    }
    .tipBg{
        background: orange;
        opacity: 0.5;/*背景透明度*/
        filter: alpha(opacity=50);
    }
```

7、动画及视野拓展

css做动画过于繁琐，已有很多工具间接性做出

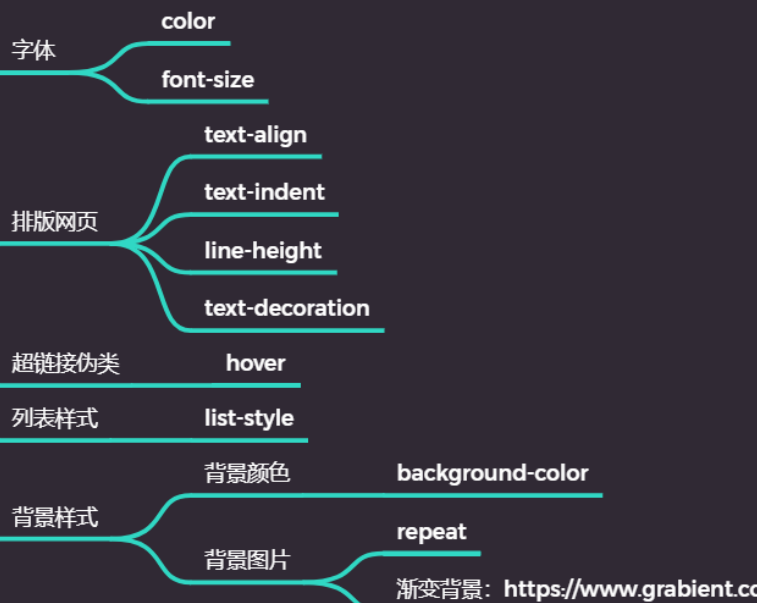
百度搜索canvas动画、[卡斯基基监控站](#)（仅作了解）

CSS3

初识及入门



美化网页



盒子模型



