

HTML-CSS

chengbenchao



目 录

第一章

- 1-1.开发环境的配置
- 1-2 账号注册
- 1-3 入门
- 1-4 接口地址
- 1-5 工具集
- 1-6 设计资源

第二章

- HTML02-am
- HTML02-pm
- 命名规范

第三章

第四章

- HTML04-am
- HTML04-pm

第五章

- HTML05-am
- HTML05-pm
- css3动画
- vue-animation

第六章

- 6-1 补充教程
- 6-2 好看的渐变色

第七章 学生作品展示

第八章 node.js

- node-ejs
- node-json接口

第九章 React入门

- 9-1 豆瓣电影

第十章 周训课件

第十一章 python

第十二章 前端技术栈

第一章

课件地址

<http://www.chengbenchao.top/>

课件下载

软件下载地址

- 开发环境的配置
- HTML,CSS基础的讲解
- 盒子模型

学生需要注册的账号

<https://gitee.com>

<https://www.yuque.com>

1-1.开发环境的配置

开发环境配置

1.安装vscode

安装插件

```
open in browser
Path Intellisense
Auto Close Tag
Live Server
css peek
HTML CSS Support
Auto Rename Tag
Color Highlight
Beautify
//文件图标主题
vscode-icons
//颜色主题
Atom One Dark Theme
One Dark Pro
```

2.npm的配置

```
npm config set registry https://registry.npm.taobao.org --global
npm config set disturl https://npm.taobao.org/dist --global
```

3.安装极速pdf ,Typora , 看图王

4.死宅的插件

```
Random Anime Wallpaper
```

1-2 账号注册

HTML-01am

1.翻墙,注册google账号安装插件

Momentum, Fireshot, pageRuler, colorZilla, 草料, adblock, JSONView

2.版本控制git,githun,码云

<https://gitee.com/>
<https://github.com/>

3.每日笔记——简书

<https://www.jianshu.com/>

4.课件地址

<https://docs.ibase.work>

好的google插件

<https://www.jianshu.com/p/3288f71bbbb6>

<https://www.jianshu.com/p/b35afc6f16ba>



1-3 入门

HTML-CSS01-PM

- 为什么要学习HTML
- HTML和CSS是什么
- 常用的HTML标签
- 常用的CSS样式
- 盒子模型

1.为什么要学习HTML和CSS



网页就是由HTML和CSS书写而成

2. HTML和CSS是什么

把网页比作一个房子



网页就是由一个个HTML标签组成



CSS样式负责修饰HTML标签

3.了解HTML标签

```
//开始标签  
<body>  
</body>  
//结束标签
```

4.HTML标签的特点:语义化

<https://blog.csdn.net/njcgosling/article/details/47130723>

5.常用的HTML标签

```
h1~h6, p, img, div, a, input, button, ul, li, dl, dt, dd
```

6.常用的css样式

- 6.1 css的语法



```
selector {property: value}
```

selector选择器通常是您需要改变样式的 HTML 元素

属性 (property) 是您希望设置的样式属性 (style attribute)

eg:

```
//HTML
```

```
<h1>hello world</h1>
```

```
//css
```

```
h1 {color:red; font-size:14px;}
```

• 6.2css常用选择器

```
<p class="one" id="two">hello world</p>
```

```
p{} //元素选择器
```

```
p.one{} //class选择器
```

```
p#two{} //id选择器
```

• 6.3盒子模型

```
//元素居中仅仅针对块元素有效
```

```
margin-left:auto;
```

```
margin-right:auto;
```

• 6.4样式重置

```
*{margin:0;padding:0}
```

• 6.5常用的css样式

color:设置文字的颜色
width:设置一个元素的宽度
height:设置一个元素的高度
background-color:设置背景颜色
background-image:设置一个元素的背景图片
line-height:设置文字的行高
text-align:设置文字对其的方式
border-width:边框的宽度
border-style:边框的样式
border-color:边框的颜色
p:hover{color:blue}当鼠标移动到元素上时可以改变元素的css样式

1-4 接口地址

1.top250

```
https://api.douban.com/v2/movie/top250
```

2.查询接口

```
https://api.douban.com/v2/movie/search?q=你的名字&count=1
```

3.subject详情接口

```
https://api.douban.com/v2/movie/subject/26683290
```

4.小程序接口地址

YAip

```
http://yapi.demo.qunar.com/mock/59961/small/music  
http://yapi.demo.qunar.com/mock/59961/small/movie
```

5.音乐播放接口

```
https://api.imjad.cn/cloudmusic.md  
//搜索接口  
https://api.imjad.cn/cloudmusic/?type=search&s=你的名字  
&limit=1  
//依赖id播放音乐  
https://api.imjad.cn/cloudmusic/?type=song&id=516358164
```

网易云音乐api

备用地址

```
//搜索接口  
https://music.aityp.com/search?keywords=${keyword}&limit=1  
//依赖id播放音乐  
https://music.aityp.com/song/url?id=347230  
http://music.163.com/song/media/outer/url?id=${id}
```

1-5 工具集

1.接口管理平台

easy-mock

2.图片管理

```
https://sm.ms/
```

1-6 设计资源

<https://pixabay.com>
<https://1x.com>
<http://pic.netbian.com>
<https://visualhunt.com/>
<https://icons8.cn/>
<https://web.500px.com>
<https://www.pexels.com>
<https://unsplash.com/>
<https://wallhaven.cc/>
花瓣

第二章

- HTML标签的分类
- CSS选择器
- CSS基本样式
- CSS的继承

HTML02-am

1. 盒子模型

1.1 盒子模型的传参

```
margin:0; //四个方向都改变
margin:0 10px; //top,bottom为0px;left,right为10px
margin:0 10px 20px; //top 0;left,right 10px;bottom 20px;
```

padding如上

1.2 元素在盒子中的起始位置

从自身宽度的地方开始

2. HTML 标签的分类

2.1 标签分类

- 块标签

```
//特点
1.独占一行
2.能够设置width,height
//常用的块标签
div,h1~h6,p,ul,li,dl,dt,dd
```

- 内联标签

```
//特点
1.并排显示
2.不能设置width,height
3.不能设置margin-top,margin-bottom
a,span,em,strong
```

- 内联块

```
//特点
1.并排显示
2.可以设置宽高
```

```
//常用的内联块标签
button, img, input
```

2.2 原理

块标签：独占一行,能够设置宽高

```
div, h1~h6, p, ul, li, dl, dt, dd
```

```
display: block;
```

内联标签:并排显示,不能设置宽高,margin-top,margin-bottom

```
a, span, em, strong
```

```
display: inline-block
```

内联块:并排显示,可以设置宽高

```
button, img, input
```

```
display: inline-block
```

2.3 skill

a.如何让内联元素和内联块元素水平居中

```
display: block;
margin-left: auto;
margin-right: auto;
```

b.如何让内联元素和内联块元素水平居中

```
//给父级加
text-align: center
```

3. css 选择器

定义选定你所要改变的元素的一种方式。

3.1分类

```
<p class="test" id="first">hello world</p>
<h4>标题</h4>
```

(1)css元素选择器

```
p{color:pink}
```

(2)class选择器

```
.test{color:yellow}
```

(3)id选择器

```
#first{color:blue}
```

(4)分组选择器

```
p,h4{background:gray}
```

(5)后代选择器

```
div>span{} //选取div所有子元素为span的标签
```

```
div span{} //选中div之后的所有span元素
```

(6)兄弟选择器

```
div+p{} 选取紧邻div之后的第一个兄弟元素
```

```
div~p{} 选取紧邻div之后的所有兄弟元素
```

(7)伪类选择器

```
div:hover{}
```

```
input:focus{}
```

(8)伪元素 -> 用css自定义生成的元素

":before" 伪元素可以在元素的内容前面插入新内容

```
p:before{
  content:''
}
```

":after" 伪元素可以在元素的内容之后插入新内容。

```
p:after{
  content:''
}
```

(9). 属性选择

```
div[class='test']{}
```

http://www.w3school.com.cn/cssref/css_selectors.asp

3.2选择器的优先级别排序

```
<div class='test' id='first'>hello world</div>
```

元素选择器<class选择器<ID选择器<!important

```
div{color:pink}<div.test{color:blue}<div#first{color:yellow}<div{color:red !important}
```

3.3选择器的权重

```
<div class="parent">  
    <div class="child">child</div>  
</div>
```

```
/* 选择器嵌套的层次越深, 那么权重越高 */  
.child{  
    color:red;  
}  
.parent>.child{  
    color:green;  
}
```


HTML02-pm

1. css 基本样式

1.1背景

背景颜色:background-color

背景图片:backgorund-image

背景重复:background-repeat

背景位置:background-position: x y

//第一个参数表示离x轴的距离,y表示离y轴的距离

//简写

background: color image repeat position

背景吸附:background-attachment:fixed | scroll

background-size

//指定背景图片大小。

background-size: x y;

//x表示宽度, y表示高度

background-size:cover;

此时会保持图像的纵横比并将图像缩放成将完全覆盖背景定位区域的最小大小。

相当于background-size:100% 100%;

1.2文本

text-align文本对齐方式

text-align: right|left|center

text-decoration文本修饰

text-decoration: underline|overline|line-through

text-indent文本缩进

text-transform文本转换(了解)

text-transform:uppercase|lowercase|capitalize

1.3字体

color:设置字体的颜色

颜色是通过CSS最经常的指定：

十六进制值 - 如: #FF0000
 一个RGB值 - 如: RGB(255,0,0)
 颜色的名称 - 如: red

```
body {color:red;}
h1 {color:#00ff00;}
h2 {color:rgb(255,0,0);}
```

//font-family 属性应该设置几个字体名称作为一种"后备"机制, 如果浏览器不支持第一种字体, 他将尝试下一种字体
 p{font-family:Helvetica,Microsoft YaHei}

推荐使用 简书的

font-family: -apple-system,SF UI Text,Arial,PingFang SC,Hiragino Sans GB,Microsoft YaHei,WenQuanYi Micro Hei,sans-serif;

font-size
 font-style:normal | italic
 font-weight:normal | bold | lighter

行高
 line-height

1.4链接

a:link - 正常, 未访问过的链接
 a:visited - 用户已访问过的链接
 a:hover - 当用户鼠标放在链接上时
 a:active - 链接被点击的那一刻

/**若单独设置几个链接, 必须遵守如下规则:

a:hover 必须跟在 a:link 和 a:visited后面
 a:active 必须跟在 a:hover后面

1.5列表(针对 ul)

```
list-style:none;
list-style-type:circle|square
list-style-image:url("xxx")
```

1.6边框

```
border-width  边框的宽度
border-style  边框的样式
border-color  边框的颜色

//可以简写成
border: width style color
p{border:1px solid #333}
```

```
//边框-单独设置各边
p
{
    border-top:1px solid #ccc;
}
```

1.7 表格

```
border-collapse
//设置表格的边框被折叠成一个单一的边框
table{border-collapse:collapse}
```

```
//可以在td,th设置这个两个属性
colspan:value    //跨越的列
rowspan:value    //跨越的行
```

1.8 轮廓 (了解)

```
//轮廓 (outline) 是绘制于元素周围的一条线，位于边框边缘的外围，可起到突出元素的作用
p{outline:1px solid pink}
```

1.9 透明

opacity

```
visibility:hidden|visible和display:none的区别
```

2. css 样式的继承

继承:是子元素对父元素的继承

2.1 width 和 height

- width

如果子元素不设置宽度，默认情况下继承父元素的宽度

- Height(特殊)

如果父元素不设置高度，默认情况下父元素继承子元素的高度

2.2 css 可以继承的属性

```
//文本相关属性
text-align, text-decoration, text-transform, text-indent(内联标签不能设置此属性)
//字体相关属性
color, font-family, font-style, font-size, font-weight, line-height
//列表相关属性
list-style
//表格相关属性
border-collapse
//其他属性
cursor, visibility
```

命名规范

1.布局相关

语义	命名	简写
盒容器	wrap	wrap
盒子	box	box
容器	container	container
主栏	main	main
侧栏	sidebar / aside	sidebar / aside
文档	doc	doc
头部	head	hd
主体	body	bd
底部	foot	ft

2.模块相关

语义	命名	简写
导航	navigation	nav
面包屑	crumb	crumb
菜单	menu	menu
选项卡	tab	tab
标题	title	title
内容	content	content
列表	list	list
表格	table	table
排行榜	rank	rank
登陆	login	login
表单	form	form
品牌标识	logo	logo
广告	advertise	ad
搜索	search	search
切换	slide	slide
提示	tips	tips

版权	copyright	copyright
按钮	button	btn
输入	input	ipt
结果	result	result
下载	download	dowload
新闻	news	news
注册	register	reg
热点	hot	hot
展开	collapse	collapse
警告	alert	alert
提示	tooltip	tooltip
下拉	dropdown	dropdown

3.状态相关

语义	命名	简写
语义	命名	简写
选中	selected	selected
当前	current	current
激活	active	active
显示	show	show
隐藏	hidden	hide
打开	open	open
关闭	close	close
出错	error	err
禁用	disabled	disabled
扩展	extend	extend

4.命名规则

4.1驼峰命名

```
<p class="navWrap"></p>
```

4.2 划线命名

```
<p class="nav-wrap"></p>
```

或

```
<p class="nav_wrap"></p>
```

第三章

1.盒子模型



- `box-sizing: border-box;`

设置padding,和border,它的宽度还是会保持不变

- `box-sizing: content-box;` (默认设置)

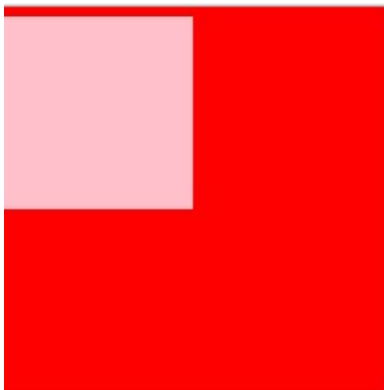
当设置padding和border时宽度会发生改变

2.float

目的:为了让元素并排显示

2.1给多个元素float

子元素float父元素的高度坍塌





2.2如何清除float

(1)给下面的兄弟元素给`clear:both;`

(2)给父级加`overflow:hidden;`

(3)用伪元素, 给父级内容生成

```
.row:before{
  display:table;
  content:""
}
.row:after{
  display:table;
  content:""
  clear:both;
}
```

demo

实现导航

3.定位position

`position:absolute` | `relative`

Relative 定位

相对定位:元素在页面上正常的位置。

Absolute定位

绝对定位的元素的位置相对于最近的相对定位的父元素, 如果没有已定位的父元素, 那么它的位置相对于<html>:

都通过left, top, right, bottom移动

z-index:设置元素的堆叠顺序 给position:absolute绝对定位的元素

position:fixed

//例子简书

demo:

1.搜索框

4.布局方式的总结

常用的布局方法

- 1.table表格布局 07年之前使用广泛,可以说是唯一的布局方式
- 2.float浮动+margin(经典)
- 3.inline-block布局 — (有小问题设置父元素的font-size:0)
- 4.flexbox布局(正统的布局方式)

5.实现元素的垂直水平居中

- 第一种

```
父元素设置parent{position:relative;}
子元素设置
child{
position:absolute;
left:50%;
top:50%;
margin-left:-50%*child*width;
margin-top:-50%*child*height;
}
```


- 第二种

```
parent{
  position:relative;
}
child{
  position:absolute;
  left:0;
  top:0;
  right:0;
  bottom:0;
  margin:auto;
}
```

6. css 样式的几种引入方式

- 外部样式表

```
<link rel="stylesheet" type="text/css" href="/c5.css">
```

- 内部样式表（位于 标签内部）

```
<style>

p{color:pink;font-size:16px}

</style>
```

- 内联样式（在 HTML 元素内部）

```
<p style="color:pink;font-size:16px">hello world</p>
```

给同一选择器设置同一样式,离元素近的样式设置方式优先级高

7.绝对路径和相对路径

绝对路径：从盘符开始的路径

```

```

相对路径：相对当前文件所在的路径

同级目录 `src='down.jpg'`

下一级目录 `src='images/down.jpg'`

上一级目录 `src='../down.jpg'`

8.定位的宽高继承问题

父元素给`position:relative`

子元素`position:absolute`

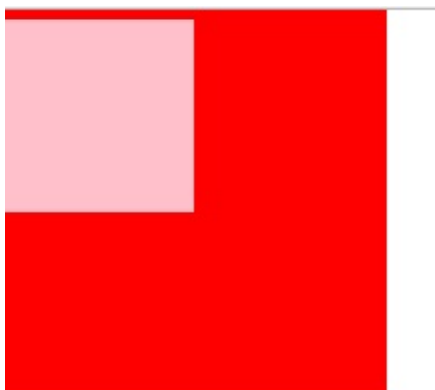
子元素绝对定位, 不会继承父元素的width

第四章

- 重说margin
- 表单
- 雪碧图
- iframe

HTML04-am

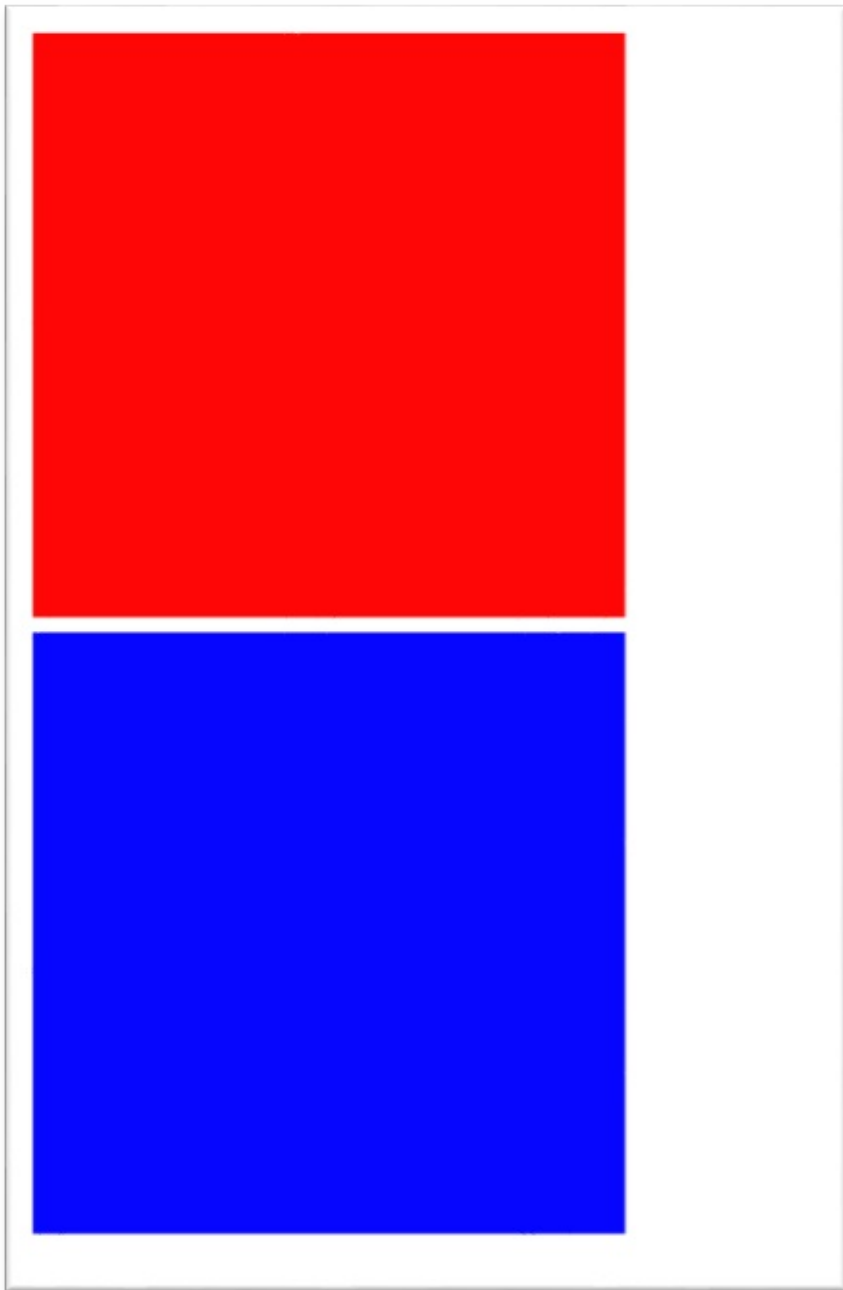
1.margin的一些问题



子元素作为父元素的[第一个元素](#),给它margin-top会存在问题

如何解决:

```
//1.给父元素
overflow: hidden;
//2.给父元素设置伪元素
parent{
    content: "";
    display: table;
}
```



两个兄弟元素之间,margin重合的问题

2. 表单

```
//input输入框type不同  
1.text 2.password 3.submit 4.radio 5.checkbox  
<input type="text">
```

2.1 一个简单的登录界面

```
<form>  
  <div>  
    <label for="text">文本</label>  
    <input id="text" type="text" />
```

```

    </div>
    <div>
      <label for="password">密码</label>
      <input id="password" type="password" />
    </div>

    <div>
      <input type="submit" value="登录" />
    </div>
  </form>

```

//定义和用法

标签为 input 元素定义 标注 （ 标记 ）。

label 元素不会向用户呈现任何特殊效果。不过，它为鼠标用户改进了可用性。如果您在 label 元素内点击文本，就会触发此控件。

标签的 for 属性 应当与相关元素的 id 属性相同 。

2.2单选框

技术要点: name名相同

```

<div>
  <label for="male">男</label>
  <input id="male" type="radio" name="sex" value="男">
  <label for="female">女</label>
  <input id="female" type="radio" name="sex" value="女">
</div>

```

2.3复合选框

```

<div>
  <label>爱好</label>
  <input type="checkbox" name="爱好" value="游泳">游泳
  <input type="checkbox" name="爱好" value="开车">开车
</div>

```

2.4下拉选框

```

<select>
  <option>洪山区</option>
  <option>青山区</option>
  <option>汉阳区</option>
</select>

```

2.5 预选下拉

```
<form >
  <select>
    <option>洪山区</option>
    <option selected>青山区</option>
    <option>汉阳区</option>
  </select>
</form>
```

2.6 文本域

```
<textarea placeholder="看点槽点，不吐不快！别憋着，马上大声说出来吧！">
</textarea>
```

2.7 特殊字符

```
&nbsp;空格 &lt; &gt;
```

再说input输入框

input type=" text"

type=" submit" 之间的区别

面试题

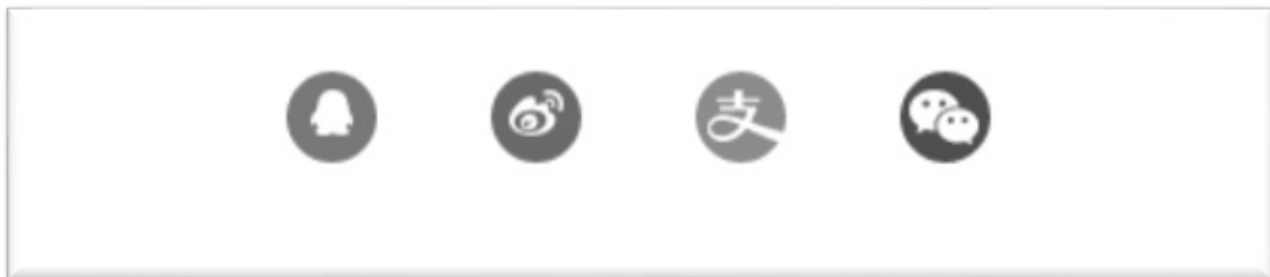
2.8 display和visibility的区别

display:none;

visibility:hidden;

2.9

讲解登录框下面的几个icon



2.10

调试代码

实现一个美化的checkbox

<https://docs.ibase.work/cssTutorial/2chan-pin/210mei-hua-checkbox.html>

HTML04-pm

1.iframe

iframe标签规定一个内联框架。

一个内联框架用来在当前 HTML 文档中嵌入另一个HTML文档

```
<iframe src="我的.html" frameborder="0" name="frame"></iframe>
<a href="test01.html" target="frame">test</a>
```

2.skill

1. 当遇到一个非文本的元素，想让其垂直居中用定位

2. 内联元素一些奇怪的现象

给button设置margin-top, span也跟着移动

```
<div>
  <button>btn</button>
  <span>深圳</span>
  <span>广州</span>
</div>
```

3. overflow:hidden;

第五章

- 下拉框
- border-radius,box-shadow
- 提取公共样式
- css2d转换
- 过渡,动画
- iconfont
- animate.css

HTML05-am

1.实现一个下拉菜单

运用知识点float, position

2.css3常用样式的讲解

2.1.边框border-radius 可以将边框设置为圆角

```
border-radius:value 四个边都会改变
可以单独改变一边
border-top-left-radius: value;
border-top-right-radius:value;
border-bottom-left-radius:value;
border-bottom-right-radius:value;
```

2.2.box-shadow可以给元素添加阴影

```
box-shadow: h-shadow v-shadow blur spread color inset;


#### *h-shadow*


必需 水平阴影的位置。允许负值。


#### *v-shadow*


必需。垂直阴影的位置。允许负值。


#### *blur*


可选。模糊距离。


#### *spread*


可选。阴影的尺寸。


#### *color*


可选。阴影的颜色。请参阅 CSS 颜色值。
inset
可选。将外部阴影（outset）改为内部阴影。
```

2.3.文字效果

```
//设置文字的阴影
2.3.1 text-shadow: h-shadow v-shadow blur color;
//用的不多，但面试的时候碰到过
2.3.2 text-overflow 文本溢出属性指定如何显示溢出内容
p{
    overflow: hidden;
    text-overflow: ellipsis;
    white-space: nowrap;
    //white-space指定文字是否换行
```

```
}
```

HTML05-pm

1. 公共样式的提取

2. css2d 转换

```
transform:translate(x,y) rotate(30deg)
//位移
translate(x,y)
//旋转
rotate()
//缩放
scale(x,y)
//倾斜
skew(x,y)
配合transform属性使用
```

2.1translate位移

该元素移动的位置，取决于宽度（X轴）和高度（Y）

```
translate(x,y)  x横坐标方向移动的距离,y纵坐标方向移动的距离div#div2

{
  transform:translate(50px,100px);
  -ms-transform:translate(50px,100px); /* IE 9 */
  -webkit-transform:translate(50px,100px); /* Safari and Chrome */
}
```

再说垂直水平居中

```
<div class="one">
  <div class="two">
  </div>
</div>

.one{
  width:400px;
  height:400px;
  background-color: red;
  position: relative;
```

```

    }
    .two{

        width:100px;
        height:100px;
        background-color: pink;
        position: absolute;
        margin:auto;
        left:0;
        top:0;
        bottom:0;
        right:0;

    }

```

2.2旋转rotate

```

div
{
transform: rotate(30deg);
-ms-transform: rotate(30deg); /* IE 9 */
-webkit-transform: rotate(30deg); /* Safari and Chrome */
}

```

2.3缩放scale()方法，该元素增加或减少的大小，取决于宽度（X轴）和高度（Y轴）的参数

```

//scale (2,3) 转变宽度为原来的大小的2倍，和其原始大小3倍的高度。

-ms-transform:scale(2,3); /* IE 9 */

-webkit-transform: scale(2,3); /* Safari */

transform: scale(2,3); /* 标准语法 */

```

2.4倾斜skew(x,y) 方法

x表示水平方向,y表示垂直方向

```

div
{
transform: skew(30deg,20deg);
-ms-transform: skew(30deg,20deg); /* IE 9 */
-webkit-transform: skew(30deg,20deg); /* Safari and Chrome */
}

```

3.transition过渡

CSS3 过渡 (transition) 配合hover使用

//改变宽度时长为2秒

```
div
{
    transition: width 2s;
    -webkit-transition: width 2s; /* Safari */
}
```

```
div:hover{
    width:100px;
}
```

多项改变

```
div
{
    transition: width 2s, height 2s, transform 2s;
    // transition: all 2s;
    -webkit-transition: width 2s, height 2s, -webkit-transform 2s;
}
div:hover{
    width:200px;
    height:200px;
    transform:rotate(30deg)
}
```

4.animation动画

动画animation

1. 定义@keyframes

A.@keyframes myfirst

```
{
    from {background: red;}
    to {background: yellow;}
}
```

B.@keyframes myfirst

```
{
    0%    {background: red;}
    25%   {background: yellow;}
```

```
50% {background: blue;}  
100% {background: green;}  
}  
  
div{  
  animation:myfirst 2s;  
}  
  
div{  
  animation:myfirst 2s infinite; //无限循环  
}
```

5. iconfont 的使用

<http://www.iconfont.cn/>

6. animate.css 的使用

<https://github.com/daneden/animate.css>

css3动画

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <style>
    .parent {
      width: 300px;
      height: 300px;
      background: url('images/timg.jpg') no-repeat center center;
      background-size: cover;
      border: 1px solid #333;
      position: relative;
      overflow: hidden;
    }

    .cover {
      position: absolute;
      width: 100%;
      height: 100%;
      right: -100%;
      bottom: 0;
      background: rgba(209, 123, 123, .5);
      transition: all 1s;
      transform-origin: left bottom;
    }

    .parent:hover .cover{
      transform: rotate(-90deg)
    }
  </style>
</head>

<body>
  <div class="parent">
    <div class="cover">

  </div>
</div>
</body>
```

```
</html>
```

vue-animation

使用vue实现一个没有卡顿的动画

Tip:animation-play-state技术要点

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <script src="https://cdn.jsdelivr.net/npm/vue@2.6.10/dist/vue.js"></script>
  <style>
    .demo{
      width:100px;
      height: 100px;
      background: pink;
      animation: circle 5s linear infinite;
    }
    .running{
      animation-play-state: running;
    }
    .paused{
      animation-play-state: paused;
    }
    @keyframes circle{
      0%{
        transform: rotate(0deg)
      }
      50%{
        transform: rotate(180deg)
      }
      100%{
        transform: rotate(360deg);
      }
    }

  </style>
</head>
<body>
  <div id="app">
    <div :class="['demo', isPlay?'running':'paused']" >

    </div>
    <button @click="handleClick" style="margin-top: 20px;">切换</button>
```

```
</div>
<script>
  new Vue({
    el: "#app",
    data: {
      isPlay: false
    },
    methods: {
      handleClick() {
        if (this.isPlay) {
          this.isPlay = false;
        } else {
          this.isPlay = true;
        }
      }
    }
  })
</script>
</body>
</html>
```

第六章

媒体查询

根据不同的设备去加载不同的样式

```
@media (max-width:768px){  
}  
@media (min-width:768px) and (max-width:992px){  
}  
@media (min-width:992px){  
}
```

6-1 补充教程

1. 标签之间的嵌套规则

块嵌套内联或内联块,a例外

eg

```
<li><a href="#">手机</a></li>
```

//错误

```
<span><div></div></span>
```

2. 简单的引入百度地图

```
<iframe src="https://map.baidu.com/"></iframe>
```

6-2 好看的渐变色

<https://uigradients.com/#VelvetSun>

```
background: #e1eec3; /* fallback for old browsers */
background: -webkit-linear-gradient(to top, #f05053, #e1eec3); /* Chrome 10-25
, Safari 5.1-6 */
background: linear-gradient(to top, #f05053, #e1eec3); /* W3C, IE 10+/ Edge, Fi
refox 16+, Chrome 26+, Opera 12+, Safari 7+ */
```

第七章 学生作品展示

2018/06-无锡冯威武组

```
https://liyao88.gitee.io/zuyuan/
```

2018/06-无锡徐保山组

```
https://mrxuxu.gitee.io/myteam/
```

2019/03-镇江武汉工程-邓舒天组

```
https://team.dengshutian.top/
```

2019/03-镇江武汉工程-刘一麟组

```
https://h5.liuyilin.top/
```


第八章 node.js

[node-ejs](#)

[node-json接口](#)

node-ejs

```
npm i express mysql
```

```
const express = require("express");
const consolidate = require("consolidate");
const mysql = require("mysql");
const server = express();
server.listen(8080);
var db = mysql.createPool({
  host: "localhost",
  user: "root",
  password: "123456",
  database: "blog"
});

server.set("view engine", "html");
server.set("views", "./template");
server.set("html", consolidate.ejs);

server.get("/", function(req, res, next) {
  //查询banner
  db.query("SELECT * FROM banner_table", (err, data) => {
    if (err) {
      res.status(500).send('database erro').end()
    } else {
      res.render('index.ejs', {banners:data});
    }
  });
});

/* static数据 */
server.use(express.static("./www"));
```

node-json接口

```
const express = require("express");
const server = express();
const mysql = require("mysql");
var db = mysql.createPool({
  host: "localhost",
  user: "root",
  password: "123456",
  database: "blog"
});
server.listen(8080);
//设置允许跨域请求
server.all("*", function(req, res, next) {
  res.header("Access-Control-Allow-Origin", "*"); //访问控制允许来源：所有
  res.header(
    "Access-Control-Allow-Headers",
    "Origin, X-Requested-With, Content-Type, Accept"
  ); //访问控制允许报头 X-Requested-With: xhr请求
  res.header("Access-Control-Allow-Methods", "PUT, POST, GET, DELETE, OPTIONS");
  //访问控制允许方法
  res.header("X-Powered-By", "nodejs"); //自定义头信息，表示服务端用nodejs
  res.header("Content-Type", "application/json;charset=utf-8");
  next();
});
server.get("/", function(req, res) {
  db.query("SELECT * FROM banner_table", (err, data) => {
    if (err) {
      res
        .status(500)
        .send("database erro")
        .end();
    } else {
      res.json(data);
    }
  });
});
```

第九章 React入门

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8" />
<title>Hello React!</title>
<script src="https://cdn.staticfile.org/react/16.4.0/umd/react.development.js"></script>
<script src="https://cdn.staticfile.org/react-dom/16.4.0/umd/react-dom.development.js"></script>
<script src="https://cdn.staticfile.org/babel-standalone/6.26.0/babel.min.js"></script>
</head>
<body>

<div id="example"></div>
<script type="text/babel">
class Hello extends React.Component{
  constructor(props){
    super(props);
    this.state = {
      msg:"change"
    }
  }
  render(){
    return (<div onClick={this.handleClick}>{this.state.msg}</div>)
  }
  handleClick=()=>{
    this.setState({
      msg:"nihao"
    })
  }
  componentDidMount(){
    console.log(1)
  }
}
ReactDOM.render(
  <Hello/>,
  document.getElementById('example')
);
</script>

</body>
</html>
```

9-1 豆瓣电影

```

<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="referrer" content="never">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <script src="https://cdn.staticfile.org/react/16.4.0/umd/react.development.
js"></script>
  <script src="https://cdn.staticfile.org/react-dom/16.4.0/umd/react-dom.deve
lopment.js"></script>
  <script src="https://unpkg.com/@babel/standalone/babel.min.js"></script>
  <script src="https://cdn.bootcss.com/jquery/3.4.1/jquery.min.js"></script>
  <title>Document</title>
  <style>
    *{margin:0;padding:0}
    .app{
      display: flex;
      max-width: 740px;
      flex-wrap: wrap;
      margin-left: auto;
      margin-right: auto;
      justify-content:space-evenly;
    }
    .app img{
      width:150px;
      height:220px;
    }
  </style>
</head>

<body>
  <div id="app"></div>
  <script type="text/babel">
    class Hello extends React.Component{
      constructor(props){
        super(props);
        this.state = {
          msg:"hello world",
          movies:[]
        }
      }
      render(){
        return (
          <div className="app">

```

```

        {this.state.movies.map((item, index)=>{
            return (
                <div className="item" key={item.id}>
                    <img src={item.images.small}/>
                    <p>{item.title.slice(0,6)+"..."}</p>
                </div>
            )
        })}
    </div>
)
}
handleClick=()=>{
    this.setState({
        msg:"change",
        movies:[]
    })
}
componentDidMount(){
    $.ajax({
        url:"https://douban.uieee.com/v2/movie/in_theaters",
        dataType:"jsonp",
        type:"get",
        success:res=>{
            this.setState({
                movies:res.subjects
            })
        }
    })
}
}
ReactDOM.render(
    <Hello/>,
    document.getElementById("app")
)
</script>
</body>

</html>

```

第十章 周训课件

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <script src="https://cdn.bootcss.com/vue/2.6.10/vue.js"></script>
  <script src="https://cdn.bootcss.com/vue-router/3.0.4/vue-router.js"></script>
  <script src="https://cdn.bootcss.com/jquery/3.4.0/jquery.js"></script>
  <style>
    * {
      margin: 0;
      padding: 0;
    }

    .app {
      text-align: center;
      padding: 20px;
      max-width: 748px;
      margin-left: auto;
      margin-right: auto;
      display: -webkit-box;
      display: -ms-flexbox;
      display: flex;
      -ms-flex-wrap: wrap;
      flex-wrap: wrap;
      -webkit-box-pack: justify;
      -ms-flex-pack: justify;
      justify-content: space-between;
    }

    .container {
      width: 150px;
      height: 240px;
      padding: 10px;
      font-size: 14px;
      -webkit-box-shadow: 0 0 5px 3px rgba(51, 51, 51, .26);
      box-shadow: 0 0 5px 3px rgba(51, 51, 51, .26);
      margin-top: 20px;
    }

    img {
      width: 150px;
```

```

    }

    button {
        margin: 0 auto;
        width: 748px;
        height: 50px;
        border-radius: 2px;
        display: block;
        height: 40px;
        line-height: 40px;
        text-align: center;
        font-size: 14px;
        background: #f7f7f7;
        text-decoration: none;
        color: #258dcd;
        border: none;
    }
</style>
</head>

<body>
    <div id="app">
        <div class="app">
            <div class="container" v-for="item of movies" ref="wrap">
                
                <p>{{item.title}}</p>

            </div>
        </div>

        <button @click="add" v-if="addMore">加载更多</button>
    </div>
<script>
    var url = "https://douban.uieee.com/v2/movie/top250"
    new Vue({
        el: "#app",
        data: {
            msg: "hello world",
            movies: [],
            addMore: false
        },
        created() {
            $.ajax({
                url,
                dataType: "jsonp",
                success: res => {
                    this.movies = res.subjects;
                    this.addMore = true
                }
            })
        }
    })

```



```
    },
    methods: {
      add() {
        this.addMore = false;
        var length = this.movies.length;
        var url = `https://api.douban.com/v2/movie/top250?start=${length}&count=20`;
        $.ajax({
          url,
          dataType: "jsonp",
          success: res => {
            this.movies = [...this.movies, ...res.subjects]
            this.addMore = true;
          }
        })
      },
    },
  })
</script>
</body>

</html>
```

第十一章 python

```
from flask import Flask, jsonify
app = Flask(__name__)
@app.route('/')
def index():
    headers = {
        'Access-Control-Allow-Origin': '*'
    }
    return jsonify({"name": "chengchao"}), 200, headers
if __name__ == "__main__":
    app.run(debug=True)
```

第十二章 前端技术栈

```
Javascript  
Typescript  
Dart  
Java  
Python  
//数据库  
MongoDB  
Mysql  
//框架  
Vue  
React  
wechat
```