

HTML

HTML是一种超文本标记语言，全称HyperText Markup Language。它是用来开发网页的一种语言，主要使用标签对来给文本添加语义，或者放置文本的部件

创建网页

1. 在VSCode中，新建文件，后缀名保存为.html
2. 在桌面或者任何地方，新建.txt文本文档，将后缀名修改为.html

浏览网页

1. 找到网页文件位置，双击打开
2. 如果VSCode中下载了live server插件，使用组合键Ctrl+Shift+P，输入live，即可打开

网页的组成

1. 结构，网页的结构部分HTML
2. 表现，网页的样式CSS
3. 行为，网页要实现的交互功能javascript

HTML骨架结构

DTD，文档类型定义，全称Document Type Definition，HTML5的DTD为

html标签对，内部包括：

head标签对，存放网页配置内容

body标签对，存放网页主要内容，包括头部、尾部等

字符集

UTF-8：几乎涵盖所有语言、文字，但一个汉字字符占用3个字节

GB2312：也叫GBK，只包括了汉字、英语在内的少数语言文字，由我国制定，一个汉字只占2个字节

站点

类似文件夹，规划网站的所有内容和代码，整合资源

SEO

SEO,搜索引擎优化，全称Search Engine Optimizati~~one~~e

常见的SEO，网页的title、两个meta标签（Keywords，Discreption）

SEO应该遵守的规范：

1. h1不能使用太多次
2. Keywords中要用逗号隔开

HTTP

HTTP，超文本传输协议，全称Hypertext Transfer Protocol

网页被用户看见的过程：开发者将网页上传到服务器上，用户发出HTTP请求，服务器后端程序运行，响应HTTP请求，前端程序运行

WEB

web标准，网页的制作标准，它是根据网页的不同组成部分生成的一系列标准。结构和表现标准由W3C（万维网）起草，行为标准由ECMA（欧洲电脑网商联合会）起草。标签

又叫标记、元素。有对标签（常规标签）和单标签（空标签）两种。

常用标签

标题

h1-h6，独占一行

文本倾斜、加粗

倾斜：`<i></i>`，倾斜效果

``，本质倾斜，斜体字，具有强调作用

加粗：``，加粗效果

``，本质加粗，粗体字，强调作用

仅占内容大小

下划线

`<u></u>`，仅占内容大小

换行

`
`

水平线

`<hr>`

段落

`<p></p>`，独占一行。

只能嵌套仅占内容大小显示的标签、文字、图片。不能嵌套占据单独一行的元素。

字符

小段文本，``，仅占内容大小

列表

独占一行，通常使用 `list-style:none` 取消默认修饰

无序列表

ul常见属性为type(disc、circle、square)

```
<ul>
  <li></li>
</ul>
```

有序列表

ol常见属性为type, start, reversed

```
<ol>
  <li></li>
</ol>
```

自定义列表

一个dl中只有一个dt, 可以有多个dd

```
<dl>
  <dt></dt>
  <dd></dd>
</dl>
```

超链接

`<a>`, 仅占内容大小

属性: href——网址

target——跳转方式: `_self`、`_blank`

title——鼠标悬停信息

图片

``, 仅占内容大小

图片格式: .jpg——普通格式

.png——背景透明

.gif——动态图

属性: src——图片地址 (有绝对路径和相对路径之分)

alt——图片无法显示时的提示信息, 与搜索引擎有关

width——宽

height——高

title——鼠标悬停信息

border——边框

【注】如果只写图片的高或者宽, 图片会等比例缩放

div

网页布局的基本标签，常叫做盒子标签，独占一行

表格

rowspan实现单元格的行合并，上下合并

colspan实现单元格的列合并，左右合并，实现列合并的时候一定要分清楚每一行的元素个数

```
<table>
  <tr>           行
    <td></td>    格子（列）
  </tr>
</table>
```

表单标签

`<form></form>`，收集用户数据

属性：action——提交地址

method——提交方式

name——表单名称

表单控件

`<input>`，仅占内容大小

属性：type——类型（`text`文本框、`password`密码框、`submit`提交按钮、`reset`重置、`button`普通按钮）

name——名字

value——值，需要用户填写的时候不需要赋值

placeholder——搜索框提示文本，H5新增

单选按钮和复选框注意事项：

1. 单选按钮应该要互斥，不能让用户同时选择，只需设置相同的name即可
2. label标签包裹单选按钮及其文字可以使得点击文字就有选择的效果

常见的大纲标签

1. header
2. footer
3. section
4. nav
5. aside

CSS引入方式

1. 内部样式
`<style></style>`
2. 外部样式
`link`
`@import`
3. 内联样式
直接在标签中添加**style**属性，又叫行内式
4. 权重对比
内联>内部/外部，加载顺序影响二者排序，一般也认为内部大于外部

选择器

选中对应的元素

CSS选择器有哪些

1. 标签
2. id
3. class
4. 复合选择器（交集、并集、后代）
5. 关系选择器（+、~、>）
6. 序号选择器
7. 属性选择器
8. 伪类和伪元素

类名的使用

1. 同一个标签可以携带多个类名
2. 同一个类名可以属于多个标签
3. 原子类

伪类和伪元素

1. 一个元素用户赋予它的状态就叫伪类，符号是：
2. 用户创造出来的一个虚拟元素就是伪元素，符号是::

基本选择器

1. 类型选择器
2. 类选择器
3. id选择器
4. 通配符：*
5. 群组选择器：，
6. 后代选择器：空格

选择器权重

id选择器 > class选择器 > 类型选择器
100 10 1

内联样式：1000

群组选择器：选择器本身，不累加

后代选择器：选择器权重之和

!important: ∞

继承：0000

【注】

1. 在继承状态下，权重选择按照就近原则来，不比较大小
2. 在非继承状态下，谁的权重大就听谁的

常用CSS属性

font-size

浏览器默认文本大小**16px**，最小文本大小**12px**

计量单位：

px

em，与最近的父元素的**font-size**成倍数关系

rem，与根元素（**html**）的**font-size**成倍数关系

color

文本颜色

写法：

rgba

三原色**rgb**

英文单词

十六进制

font-family

字体样式

font-weight

字体粗细

值:

bold--加粗

bolder--粗体字

00-300--变细

600-900--加粗

font-style

文本风格

值:

italic--斜体字

oblique--倾斜效果

normal--正常

line-height

行高，行高的计算是文字加上文字到顶部的距离和到底部的距离

行高等于盒子高可以让文本垂直居中

text-align

文本水平对齐方式

值:

left--左

right--右

center--居中

justify--两端对齐

text-decoration

文本修饰

值:

none--不要修饰

underline--下划线

overline--上划线

line-through--中划线

text-indent

首行缩进

只针对第一行，可以为负值

list-style

列表风格

background-color

背景颜色，与img标签的区别是，它是内嵌的

background-image

背景图片
或背景颜色

值：url——图片地址

背景渐变效果

1. 线性渐变background-image: linear-gradient（方向，颜色，颜色）、（颜色 区间1，颜色 区间2）、（方向，颜色 区间1，颜色 区间2...）

2. 径向渐变background-image: radial-gradient（圆心百分比，颜色，颜色）

background-size

背景图片缩放，H5新增

background-repeat

背景图片平铺

值：
no-repeat——不平铺

repeat-X——水平平铺

repeat-Y——垂直平铺

background-position

背景图定位

写法：
写数字——前面是水平方向，后面是垂直方向

写单词——left center right

background-attachment

背景图关联

值：
`scroll`——随页面滚动

`fixed`——固定在页面

CSS继承性

CSS有三大特性，分别是继承性，层叠性，优先级

继承性：

不可继承的：`display`、`margin`、`border`、`padding`、`background`、`height`、`min-height`、`max-height`、`width`、`min-width`、`max-width`、`overflow`、`position`、`left`、`right`、`top`、`bottom`、`z-index`、`float`、`clear`、`table-layout`、`vertical-align`、`page-break-after`、`page-break-before`和`unicode-bidi`。

所有元素可继承：`visibility`和`cursor`。

内联元素可继承：`letter-spacing`、`word-spacing`、`white-space`、`line-height`、`color`、`font`、`font-family`、`font-size`、`font-style`、`font-variant`、`font-weight`、`text-decoration`、`text-transform`、`direction`。

终端块状元素可继承：`text-indent`和`text-align`。

列表元素可继承：`list-style`、`list-style-type`、`list-style-position`、`list-style-image`。

表格元素可继承：`border-collapse`。

CSS精灵图

使用`background-image`、`background-repeat`和`background-positon`属性实现精灵图的截取，坐标位置前要加负号

浮动

目的

让垂直排列的盒子，水平排列

影响：

让元素脱离文档流，就是脱离上下排列的这种顺序。浮动之后原本的元素位置会空出来被后面的元素占据。

使用情况

浮动流与标流使用方式：垂直方向使用标准流, 水平方向使用浮动流

贴靠现象

如果父元素的宽度能够显示所有浮动元素, 那么浮动的元素会并排显示

如果父元素的宽度不能显示所有浮动元素, 那么会从最后一个元素开始往前贴靠

如果贴靠了前面所有浮动元素之后都不能显示, 最终会贴靠到父元素的左边或者右边

浮动元素字围现象

浮动元素不会挡住没有浮动元素中的文字, 没有浮动的文字会自动给浮动的元素让位置, 这个就是浮动元素字围现象

高度塌陷问题

当父元素不设置高度的时候靠子元素撑大, 也就是说子元素有多高, 父元素就有多高, 当子元素浮动后, 父元素就会高度塌陷。父元素高度塌陷后, 父元素下面的元素就会向上移动, 这样会导致整个页面的布局混乱。

浮动元素的高度

浮动元素如果没有高度, 则高度由子盒子撑开

盒模型

1. 盒模型的组成

内容区域: `width`, `height`

内边距区域: `padding`

边框区域: `border`

外边距: `margin`

2. padding和margin的三种写法

四个数值: 上, 右, 下, 左

三个数值: 上, 左右, 下

两个数值: 上下, 左右

3. margin塌陷现象

在标准文档流中, 垂直方向的`margin`有塌陷现象, 数值小的`margin`会塌陷进数值大的`margin`里面

解决方案:

兄弟关系——只给一个方向的`margin`就行

父子关系——给父元素添加边框或者用父元素的内边距撑开

形成BFC

4. 标准盒模型

5. 怪异盒模型

边框

1. 边框的三要素

`border-width`宽度、`border-style`线形、`border-color`颜色

常见线形有`solid`实线、`dashed`虚线、`dotted`点状线

边框三要素也可合写成`border: width style color`的形式

边框的四个边也可单独设置，四个边的三要素也可单独设置

2. 圆角的设置

`border-radius`，边框半径

想要设置正圆形，就采用`50%`的描述方法，而不是`px`

3. 阴影的设置

`border-shadow`

四个数据分别代表，水平方向的位移，垂直方向的位移，阴影模糊程度，阴影颜色

盒子可以有多个阴影，只需要用逗号隔开

添加`inset`可以给盒子设置内阴影

4. 盒子制作三角形

宽度、高度设置为0，选择一条边框设定颜色，其他三条边框的颜色设置为透明色(`transparency`)

PS基本使用

1. 测距：矩形选框工具，F8查看信息

清除参考线：

单根——矩形选框工具，`Ctrl+鼠标左键拖拽到标尺 (Ctrl+R)`

全部——视图，清除参考线

2. 取色：吸管工具，前景色

3. 页面放大缩小：`Ctrl+Shift+Alt`+滚轮，也可以设置为滚轮

4. 切图：

标尺: **Ctrl+R**

信息面板: **F8**

尺寸: 编辑-首选项-单位与标尺-改单位像素

切图保存: **Ctrl+Shift+Alt+S**, 拉下阴影, 存储, 所有用户切面

文本溢出

overflow

文本溢出

值:

visible---默认值

hidden---超出隐藏

scroll---滚动查看

auto---自适应, 滚轮

white-space

文本换行方式

值: **nowrap**---强制不换行, 但是优先级没有`
`标签高

text-overflow

省略号

值: **clip**---裁剪

ellipsis---省略号

【注】单行文本才能设置文本溢出

省略号应用

```
width:;  
white-space:nowrap;  
overflow:hidden;  
text-overflow:ellipsis;
```

【注】主要针对单行文本设置

元素类型

块状元素 (block)

1. 单独占据一行
2. 能设置宽高
3. 能正常设置内外边距
4. 矩形显示
5. 通常用来搭建页面结构
6. 代表元素: `div`、`h1-h6`、`p`、`ol`、`ul`、`li`.....

行内元素 (inline)

1. 只占据自己内容大小的位置
2. 不能设置宽高
3. 不能正常设置垂直方向的内外边距
4. 非矩形显示
5. 不能用来搭建页面结构
6. 代表元素: `span`、`a`、`u`、`em`、`i`.....

行内块元素 (inline-block)

1. 只占据自己内容的大小
2. 可以设置宽高
3. 能正常设置外边距
4. 标尺效应
作用: 让一个元素在容器内垂直居中
原理: `vertical-align` 行内块元素专用垂直居中 (两个子元素互相比对)
用法: 图片与文字之间用得最多
两个兄弟元素都是`inline-block`
都设置了`vertical-align: middle`
标尺元素`width:0;height:100%;`
5. 代表元素: `img`、`input` (实际上是属于`inline`元素)

display

值:

`block`—转换成块级元素 (显示)

`inline`—转换成行内元素

`inline-block`—转换成行内块元素

`none`—隐藏

`list-item`—列表显示

伪类选择器

1. `:link`—原始状态
2. `:hover`—点击时状态
3. `:visited`—点击后状态
4. `:active`—进入时状态

【注】有顺序要求，爱恨原则love&hate

置换元素

img和input，其他都是非置换元素

定位

作用

改变元素的位置

position

值：`relative`—相对定位

【注】

以自身为参考

不会脱离文档流，依然占据自己的位置

`absolute`—绝对定位

【注】

以最近的定位的父元素为参考或者根元素（`html`）

会脱离文档流，后面的元素会占据元素位置

`fixed`—固定定位

【注】

以浏览器为参考

会脱离文档流，后面的元素会占据元素位置

`sticky`—粘性定位

【注】

吸顶效果

相对定位和绝对定位的关系

参考物不同

前者占位，后者不占位

包含块

嵌套关系

子绝父相

层级关系

改变元素的层叠关系：**z-index**

数字越大，层级越高

可以为负，默认值为**auto**

必须添加了定位的元素才可以修改，默认定位除外

并列关系：都能添加层级关系

嵌套关系：只能修改子元素的层级关系

绝对定位居中方式

方式一：

```
position: absolute;
```

```
left: 0;
```

```
right: 0;
```

```
top: 0;
```

```
bottom: 0;
```

```
margin: auto
```

方式二：

```
position: absolute;
```

```
left: 负的宽度的一半;
```

```
top: 负的高度的一半;
```

```
margin-left: 50%;
```

margin-top: 50%;

透明度

opacity: 数字; (0-1)

filter: alpha (opacity=数字); (0-100)

锚点

a标签跳转地址加id名字就可以实现锚点

H5表格新增内容

Html新标签

1. th 列标题

2. colgroup 列分组

样式名: span--数字, 几列为一组

3. thead、tbody、tfoot

CSS新属性

1. border-spacing 单元格间距

2. border-collapse 是否合并单元格边框

值: separate--默认值

collapse--合并

3. table-layout 单元格是否固定

值: auto--自动

fixed--固定

H5表单新增内容

1. input

type="radio" 单选按钮 (同组的name值必须一致, 保持互斥)

"checkbox" 多选框

"file"文件上传域

"hidden"隐藏表单

"image"图片按钮

src=图片地址

2. select >option

下拉列表

option的value必须写

3. textarea

多行文本输入框

样式: cols—宽度

rows—高度

checked—默认选中

属性: resize:none; 禁止拖拽

4. fieldset >legend

表单字段集

子集只能有一个legend，但是可以嵌套fieldset

legend

样式: align—对齐方式 (left、right、center、justify)

5. label

提示信息

样式: for—关联标签中的id值

BFC

块级格式化上下文，简单理解就是一个特殊的容器

作用

解决margin重叠问题

解决高度塌陷问题

BFC布局规则

1. 内部的Box会在垂直方向，一个接一个地放置(遵循普通文档流)
2. Box垂直方向的距离由margin决定。属于同一个BFC的两个相邻Box的margin会发生重叠(遵循普通文档流)
3. 每个元素的margin box的左边，与包含块border box的左边相接触
4. BFC的区域不会与float box重叠
5. BFC就是页面上的一个隔离的独立的容器，容器里面的子元素不会影响到外面的元素。
6. 计算BFC的高度时，浮动元素也参与计算

BFC触发方式

1. 根元素html
2. float属性不为none
3. position为absolute或fixed
4. display为inline-block, table-cell, table-caption, flex, inline-flex
5. overflow不为visible

公共样式设置的原因

1. img设置为border:0;
低版本浏览器在，使用a标签包裹图片时，图片会出现边框
2. input、fieldset设置为outline:none;
输入时无边框

PC宽高自适应

1. PC端自适应
元素宽高由子元素撑起来
2. 宽高自适应

宽度自适应

- 1.width: 100%,
 2. 不写宽度
- 【只针对块级元素】

高度自适应

- 1.height: auto
 2. 不写高度
 - 3.height: 100%
- 【前提条件：父元素有高度，或者所有的父元素都有百分比，html, body{height: 100%}】

3. 最大/最小高度

min-height: 最小高度
max-height: 最大高度

4. 兼容写法

【注】ie低版本浏览器不兼容

兼容写法:

1.

```
min-height:100px
```

```
height:auto !important
```

```
height:100px
```

2.

```
min-height:100px
```

```
max-height:300px
```

```
_height:100px
```

隐藏属性

1. `visibility:hidden`

2. 与`display:none`的区别

都是隐藏，`visibility:hidden`隐藏占位，`display:none`隐藏不占位

伪元素

1. 单冒号写法: `css`

2. 双冒号写法: `css3`，没有具体区别，不影响使用

```
::before {content:"";}
```

```
::after{content:"";}
```

```
::first-line{}第一行文本
```

```
::first-letter{}第一个字文本
```

【只针对块级元素有效】

高度塌陷

1. 子元素浮动时，如果父元素不给高度，就会产生塌陷

2. 解决办法：清除浮动

1. 给父元素添加高度——不灵活

2. 给父元素添加`overflow:hidden`；形成BFC——超出父元素的内容将被隐藏

3. 给浮动元素的最后位置添加一个兄弟块级元素，设置`clear:left/right/both`；——代码冗余

4. 万能清除法

```
.clearFix {  
  content:"";/* content:"." 兼容写法*/  
  display:block;  
  width:0;  
  height:0;  
  clear:both;  
  /*visibility:hidden;  
  overflow:hidden; 兼容写法*/  
}  
/* .clearFix {zoom:1} 兼容写法*/
```

兼容性前缀

```
-webkit- webkit内核  
-ms-      ie  
-moz-     火狐  
-o--      欧朋
```

语义化标签

```
header  
section  
footer  
nav  
article  
aside  
figure > figcaption  
main (ie不兼容)
```

音频标签

```
video 视频标签  
    width=""  
    height=""  
    controls="显示控件"  
    autoplay="自动播放"  
    loop="循环播放"  
    poster="第一帧图像"  
    muted="静音"  
  
audio 音频标签  
    controls="显示控件"  
    autoplay="自动播放"  
    loop="循环播放"  
    muted="静音"  
  
source 备份文件  
    src="文件地址"  
  
    type="文件类型"  
    视频格式 类型:  
        video/ogg video/mp4 video/webm  
  
    音频格式类型:  
        audio/ogg audio/mpeg
```

H5表单元素

html5表单:

`type="email"` 限制用户必须输入email类型

`type="url"` 限制用户必须输入url类型

`type="range"` 产生一个滑动条表单

`type="number"` 限制用户必须输入数字

`type="search"` 产生一个搜索意义的表单

`type="color"` 生成一个颜色选择的表单

`type="time"` 限制用户必须输入时间类型

`type="month"` 限制用户必须输入月类型

`type="week"` 限制用户必须输入周类型

`type="datetime-local"` 选取本地时间

`type="date"` 日期

`output` 输出标签

`datalist > option` 提示信息

表单属性:

`required` 监测是否为空。

`min` 最小

`max` 最大

`step` 步幅 确定一个法定值。 -3 0 3 6 9

`list` 必须结合`datalist`标签，绑定`datalist` id名称。

`autocomplete` 是否自动提示信息 属性值 on off

`placeholder` 文本框的提示信息

`autofocus` 自动聚焦。一个页面只能由一个。

`pattern` 后面的属性值是一个正则表达式。

`novalidate` 取消验证

`multiple` 选择（上传）多个

线性渐变

linear-gradient(方位, 颜色1 起始位置, 颜色2 结束位置...)

方位:

left right
top bottom
deg

起始位置:

% px 结束位置比起始位置的数字大

重复渐变: **repeating-linear-gradient**()

径向渐变

radial-gradient(圆心坐标, 半径, 形状, 颜色1 起始位置, 颜色2 结束位置...)

圆心坐标

px %

半径:

px %

近边 近角 远边 远角

起始位置:

% px 结束位置比起始位置的数字大

形状:

ellipse 椭圆(元素本身是正方形, 就不可能出现椭圆)

circle 正圆

重复径向渐变: **repeating-radial-gradient**()

过渡—transition

过渡属性 过渡时间 延迟时间 运动速度;

transition-property: 过渡属性;

ransition-duration: 过渡时间;

transition-delay: 延迟时间;

transition-timing-function: 过渡速度;

linear

ease

ease-in

ease-out

ease-in-out

贝塞尔曲线 **cubic-bezier**(0.02, 0.44, 0.25, 1)

【过渡可以包含变形】

变形—transform

功能函数:

位移

`translate(x,y)`

`translateX()`

`translateY()`

【注】可以为负数

缩放:

`scale(x,y)`

`scaleX()`

`scaleY()`

【注】可以为负数,默认值为1

旋转:

`rotateX()`

`rotateY()`

`rotate()`

【注】默认是z轴

变形:

`skew(x,y)`

`skewX()`

`skewY()`

【注】可以为负数

变形原点:

`transform-origin;`

`left center right`

`top center bottom`

`px %`

【拓展】

先旋转,后位移 和先位移,后旋转 效果是否一样

答: 效果不一样,因为先旋转改变了位移的方向

3D

景深

给父元素添加`perspective`或者给子元素添加`transform:perspective()`

取值一般在900-1200之间

观察3D元素的角度

`perspective-origin:`

`center center`(中心)

`left top`(左上角)

`100% 100%`(右下角)

实现3D场景

`transform-style: flat`

`preserve-3d`

3D位移

CSS3中的3D位移主要包括`translateZ()`和`translate3d()`两个功能函数：

`translate3d(tx,ty,tz)`

ty：代表纵向坐标位移向量的长度；

tx：代表横向坐标位移向量的长度；

tz：代表Z轴位移向量的长度。此值不能是一个百分比值，如果取值为百分比值，将会认为无效值。

`translateZ(t)`

指的是Z轴的向量位移长度。

3D旋转

CSS3中的3D旋转主要包括`rotateX()`、`rotateY()`、`rotateZ()`和`rotate3d()`四个功能函数：

`rotate3d(x,y,z,a)` (建议取值0或1)

x：是一个0到1之间的数值，主要用来描述元素围绕X轴旋转的矢量值；

y：是一个0到1之间的数值，主要用来描述元素围绕Y轴旋转的矢量值；

z：是一个0到1之间的数值，主要用来描述元素围绕Z轴旋转的矢量值；

a：是一个角度值，主要用来指定元素在3D空间旋转的角度，如果其值为正值，元素顺时针旋转，反之元素逆时针旋转。

3D缩放

3D缩放：CSS3中的3D缩放主要包括`scaleZ()`和`scale3d()`两个功能函数；

通过使用3D缩放函数，可以让元素在Z轴上按比例缩放。默认值为1，当值大于1时，元素放大，反之小于1大于0.01时，元素缩小

`scale3d()`

sx：横向缩放比例；

sy：纵向缩放比例；

sz：Z轴缩放比例

`scaleZ(s)`

s：指定元素每个点在Z轴的比例。

【注】`scaleZ()`和`scale3d()`函数单独使用时没有任何效果，需要配合其他的变形函数一起使用才会有效果

动画

关键帧

不同于过渡动画只能定义首尾两个状态，关键帧动画可以定义多个状态，或者用关键帧的话来说，过渡动画只能定义第一帧和最后一帧这两个关键帧，而关键帧动画则可以定义任意多的关键帧，因而能实现更复杂的动画效果。

关键帧函数

```
@keyframes mymove{
  from{初始状态属性}
  to{结束状态属性}
}
或
@keyframes mymove{
  0%{初始状态属性}
  50%（中间再可以添加关键帧）
  100%{结束状态属性}
}
```

animation

animation-name

检索或设置对象所应用的动画名称

必须与规则@keyframes配合使用，

eg:@keyframes mymove{} animation-name:mymove;

animation-duration

检索或设置对象动画的持续时间

说明: animation-duration:3s; 动画完成使用的时间为3s

animation-timing-function

检索或设置对象动画的过渡类型

属性值

linear: 线性过渡。等同于贝塞尔曲线(0.0, 0.0, 1.0, 1.0)

ease: 平滑过渡。等同于贝塞尔曲线(0.25, 0.1, 0.25, 1.0)

ease-in: 由慢到快。等同于贝塞尔曲线(0.42, 0, 1.0, 1.0)

ease-out: 由快到慢。等同于贝塞尔曲线(0, 0, 0.58, 1.0)

ease-in-out: 由慢到快再到慢。等同于贝塞尔曲线(0.42, 0, 0.58, 1.0)

step-start:马上跳到动画每一结束帧的状态

animation-delay

检索或设置对象动画延迟的时间

说明: animation-delay:0.5s; 动画开始前延迟的时间为0.5s

animation-iteration-count

检索或设置对象动画的循环次数

属性值

animation-iteration-count: infinite | number;

infinite: 无限循环

number: 循环的次数

animation-direction

检索或设置对象动画在循环中是否反向运动

属性值

normal: 正常方向

reverse: 反方向运行

alternate: 动画先正常运行再反方向运行，并持续交替运行

alternate-reverse: 动画先反运行再正方向运行，并持续交替运行

animation-play-state

检索或设置对象动画的状态

属性值

animation-play-state:running | paused;

running: 运动

paused: 暂停

animation-play-state: paused;

当鼠标经过时动画停止，鼠标移开动画继续执行

简写: 名称 时间 过渡类型(速率) 延迟 动画方式(是否无限循环)

与transition(过渡)的区别

相同点: 都是随着时间改变元素的属性值。

不同点: **transition**需要触发一个事件(hover事件或click事件等)才会随时间改变其css属性; 而**animation**在不需要触发任何事件的情况下也可以显式的随着时间变化来改变元素css的属性值, 从而达到一种动画的效果, css3的**animation**就需要明确的动画属性值

渐进增强和优雅降级

区别: 优雅降级是从复杂的现状开始, 并试图减少用户体验的供给, 而渐进增强则是从一个非常基础的, 能够起作用的版本开始, 并不断扩充, 以适应未来环境的需要。降级(功能衰减)意味着往回看; 而渐进增强则意味着朝前看, 同时保证其根基处于安全地带。

选择器

属性选择器

1. E[attr]: 只使用属性名, 但没有确定任何属性值
2. E[attr="value"]: 指定属性名, 并指定了该属性的属性值
3. E[attr~="value"]: 指定属性名, 并且具有属性值, 此属性值是一个词列表, 并且以空格隔开, 其中词列表中包含了一个value词, 而且等号前面的"~"必须的
4. E[attr^="value"]: 指定了属性名, 并且有属性值, 属性值是以value开头的
5. E[attr\$="value"]: 指定了属性名, 并且有属性值, 而且属性值是以value结束的
6. E[attr*="value"]: 指定了属性名, 并且有属性值, 而且属性值中包含了value
7. E[attr|=“value”]: 指定了属性名, 并且属性值仅是value或者以“value-”开头的值(比如说left-con)

伪类选择器

结构性伪类选择器

1. **X:first-child** 匹配子集的第一个元素。IE7就可以支持
2. **X:last-child** 匹配父元素中最后一个X元素
3. **X:nth-child(n)** 用于匹配索引值为n的子元素。索引值从0开始
4. **X:only-child** 这个伪类一般用的比较少, 比如上述代码匹配的是div下的有且仅有一个的p, 也就是说, 如果div内有多个p, 将不配。
5. **X:nth-last-child(n)** 从最后一个开始算索引。
6. **X:first-of-type** 匹配同级兄弟元素中的第一个X元素
7. **X:last-of-type** 匹配同级兄弟元素中的最后一个X元素
8. **X:nth-of-type(n)** 匹配同类型中的第n个同级兄弟元素X
9. **X:only-of-type** 匹配属于同类型中唯一兄弟元素的X
10. **X:nth-last-of-type(n)** 匹配同类型中的倒数第n个同级兄弟元素X
11. **:root** 匹配文档的根元素。在HTML(标准通用标记语言下的一个应用)中, 根元素永远是HTML

12. **:not(e)**否定选择器
13. **X:empty**匹配没有任何子元素（包括包含文本）的元素X

目标伪类选择器

E:target——选择匹配E的所有元素，且匹配元素被相关URL指向

UI 元素状态伪类选择器

1. E:**enabled**
匹配所有用户界面（form表单）中处于可用状态的E元素
2. E:**disabled**
匹配所有用户界面（form表单）中处于不可用状态的E元素
3. E:**checked**
匹配所有用户界面（form表单）中处于选中状态的元素E
4. E::**selection**
匹配E元素中被用户选中或处于高亮状态的部分

动态伪类选择器

1. E:**link**
链接伪类选择器
选择匹配的E元素，而且匹配元素被定义了超链接并未被访问过。常用于链接描点上
2. E:**visited**
链接伪类选择器
选择匹配的E元素，而且匹配元素被定义了超链接并已被访问过。常用于链接描点上
3. E:**hover**
用户行为选择器
选择匹配的E元素，且用户鼠标停留在元素E上。IE6及以下浏览器仅支持a: hover
4. E:**active**
用户行为选择器
选择匹配的E元素，且匹配元素被激活。常用于链接描点和按钮上
5. E:**focus**
用户行为选择器
选择匹配的E元素，而且匹配元素获取焦点

层级选择器

E>F

子选择器
选择匹配的F元素，且匹配的F元素所匹配的E元素的子元素

E+F

相邻兄弟选择器
选择匹配的F元素，且匹配的F元素紧位于匹配的E元素的后面

E~F

通用选择器
选择匹配的F元素，且位于匹配的E元素后的所有匹配的F元素

属性

text-shadow

文本阴影

word-wrap、word-break

文本换行

@font-face

@font-face是CSS3中的一个模块，他主要是把自己定义的Web字体嵌入到你的网页中，随着@font-face模块的出现，我们在Web的开发中使用字体不怕只能使用Web安全字体（@font-face这个功能早在IE4就支持）

@font-face的语法规则：

```
@font-face {  
    font-family: <YourWebFontName>;  
    src: <source> [<format>][, []]*;  
    [font-weight: <weight>;  
    [font-style: <style>;  
}
```

background-origin

背景原点，指定background-origin属性应该是相对位置

属性值：

padding-box	背景图像填充框的相对位置	默认值
border-box	背景图像边界框的相对位置	
content-box	背景图像的相对位置的内容框	

background-clip

背景裁切，规定背景的绘制区域。

属性值：

border-box	背景被裁剪到边框盒。	
padding-box	背景被裁剪到内边距框。	
content-box	背景被裁剪到内容框。	默认值

background-size

背景尺寸

属性值：

length (10px)	规定背景图的大小。第一个值宽度，第二个值高度。
Percentage(%)	以百分比为值设置背景图大小
cover	把背景图像扩展至足够大，以使背景图像完全覆盖背景区域
contain	把图像扩展至最大尺寸，以使其宽度或高度完全适应内容区域。

多背景图的设置

`background:url(),url()...url()` 颜色

写在前面的图片会压盖在后面的图片上

border-image

图片边框

`border-image` 属性是一个简写属性，用于设置以下属性：

```
border-image-source 用在边框的图片的路径。
border-image-slice  图片边框向内偏移(不加单位)。
border-image-repeat 图像边框是否应平铺(repeat)、铺满(round)或拉伸(stretch)
border-image-outset 边框图像区域超出边框的量(值是一个倍数)
```

border-radius

边框圆角

box-shadow

盒子阴影

属性值：

```
h-shadow    必需的。水平阴影的位置。允许负值
v-shadow    必需的。垂直阴影的位置。允许负值
blur        可选。模糊距离
spread      可选。阴影的大小
color       可选。阴影的颜色。
inset       可选。从外层的阴影（开始时）改变阴影内侧阴影
box-shadow: 10px 10px 5px 10px #888888 inset;
```

计算模式

`calc(100% - 100px)`

1. 低版本浏览器不兼容
2. 运算符前后需要添加空格

怪异盒模型

`box-sizing`

正常盒模型

属性：`content-box`

$w = \text{width} + \text{左右内边距} + \text{左右边框}$

$h = \text{height} + \text{上下内边距} + \text{上下边框}$

怪异盒模型

属性: border-box

w = width (包含边框和边距)

h = height (包含边框和边距)

Flex布局

display:

flex 弹性盒 (容器本身独占一行)

inline-flex 弹性盒 (容器本身在一行)

【注】:

1. 浮动、清楚浮动、垂直方向对齐失效
2. 默认主轴为水平方向, 侧轴为垂直方向
3. 容器 (flex-container) : 父元素; 项目 (flex-item) : 子元素
4. 任何一个容器都可以设置为flex布局, 众生平等

容器属性

flex-direction:

row 水平主轴

row-reverse 反向水平主轴

column 垂直主轴

column-reverse 反向垂直主轴

flex-wrap:

nowrap 不换行

wrap 换行

wrap-reverse 从底部开始换行

flex-flow: (缩写形式) 是否换行, 主轴方向

justify-content: 设置子元素在主轴上的对齐方式

flex-start 起始位置

center 居中

flex-end 结束位置

space-around 两端平分 (每个子元素)

space-between 两端对齐

space-evenly 平均分配 (有兼容问题)

align-items: 设置子元素在侧轴上的对齐方式 (单行)

flex-start 起始位置

center 居中

flex-end 结束位置

stretch 拉伸（沿侧轴）

baseline 文字底部对齐

align-content: 设置子元素在侧轴上的对齐方式（多行），必须和换行一起使用，否则无法生效

项目属性

align-self: 单个子元素侧轴对齐方式

auto 默认值

stretch 拉伸

center 居中

flex-start

flex-end

order: 排序，数字越大，越靠后

flex: ; 项目占比

拆分写法:

flex-grow: ; 扩大

flex-shrink: ; 缩小

flex-basis: ; 长度

多列布局

1. column-count:分列;
2. column-gap:列间距;
3. column-rule:1px solid #000; 列边框
4. column-fill:填充方式;
auto 优先铺满前面的
balance 尽可能平均分
5. column-width:300px; 列宽
6. columns: 200px 2; 列宽和列数的综合写法
7. break-inside: avoid 防止断层

响应式布局

1. 概念：对页面进行响应式的设计实现，需要对相同内容进行不同宽度的布局设计，有两种方式：pc 优先（从pc端开始向下设计）；移动优先（从移动端向上设计）；无论基于那种模式的设计，要兼容所有设备，布局响应时不可避免地需要对模块布局做一些变化（发生布局改变的临界点称之为断点）。

- 2. 特点：
 - 面对不同分辨率设备灵活性强
 - 能够快速解决多设备显示适应问题
- 3. 缺点：
 - 兼容各种设备工作量大，效率低下
 - 代码累赘，会出现隐藏无用的元素，加载时间加长
 - 其实这是一种折中性质的设计解决方案，多方面因素影响而达不到最佳效果
 - 一定程度上改变了网站原有的布局结构，会出现用户混淆的情况

媒体查询

- 1. 概念:媒体查询可以让我们根据设备显示器的特性（如视口宽度、屏幕比例、设备方向：横向或纵向）为其设定CSS样式，媒体查询由媒体类型和一个或多个检测媒体特性的条件表达式组成。媒体查询中可用于检测的媒体特性有 width、 height 和 color （等）。使用媒体查询，可以在不改变页面内容的情况下，为特定的一些输出设备定制显示效果。
- 2. 语法

```
@media all and (min-width:320px) {  
    body { background-color:blue;}  
}
```

- 3. 设备类型(默认值为all)

值	类型
all	所有设备
aural	听觉设备
braille	点字触觉设备
handle	便携设备
print	打印预览图等设备
projection	投影设备
screen	显示器、笔记本、移动端等设备
tty	如打字机或者终端等设备
tv	电视机等设备类型
embossed	盲文打字机

- 4. 主断点和次断点
 - 主断点用来区分不同大小的设备，次断点用来区分相同设备下的不同类型

移动端适配

元标签的设置

H5页面窗口自动调整到设备宽度，并禁止用户缩放页面

```
<meta name="viewport" content="width=device-width,initial-scale=1.0,minimum-scale=1.0,maximum-scale=1.0,user-scalable=no" />
```

忽略将页面中的数字识别为电话号码

```
<meta name="format-detection" content="telephone=no" />
```

忽略Android平台中对邮箱地址的识别

```
<meta name="format-detection" content="email=no" />
```

当网站添加到主屏幕快速启动方式，可隐藏地址栏，仅针对ios的safari

```
<meta name="apple-mobile-web-app-capable" content="yes" />
```

<!-- ios7.0版本以后，safari上已看不到效果 -->

将网站添加到主屏幕快速启动方式，仅针对ios的safari顶端状态条的样式

```
<meta name="apple-mobile-web-app-status-bar-style" content="black" />
```

<!-- 可选default、black、black-translucent -->

重点设置

```
<meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=no">
```

width = device-width: 宽度等于当前设备的宽度

initial-scale: 初始的缩放比例（默认设置为1.0）

minimum-scale: 允许用户缩放到的最小比例（默认设置为1.0）

maximum-scale: 允许用户缩放到的最大比例（默认设置为1.0）

user-scalable: 用户是否可以手动缩放（默认设置为no，因为我们不希望用户放大缩小页面）

尺寸名词

1、屏幕尺寸：指的是屏幕对角线的长度

2、分辨率：是指宽度上和高度上最多能显示的物理像素点个数

3、点距：像素与像素之间的距离，点距和屏幕尺寸决定了分辨率大小

4、PPI: 屏幕像素密度，即每英寸(1英寸=2.54厘米)聚集的像素点个数，这里的一英寸还是对角线长度

5、DPI: 每英寸像素点，印刷行业术语。对于电脑屏幕而言和PPI是一个意思

6、设备像素(又称为物理像素): 指设备能控制显示的最小物理单位，意指显示器上一个一个的点。从屏幕在工厂生产出的那天起，它上面设备像素点就固定不变了，和屏幕尺寸大小有关，单位 pt。

7、设备独立像素(也叫密度无关像素或逻辑像素): 可以认为是计算机坐标系统中得一个点，这个点代表一个可以由程序使用的虚拟像素(比如: css像素)，这个点是没有固定大小的，越小越清晰，然后由相关系统转换为物理像素。

8、css像素(也叫虚拟像素): 指的是 CSS 样式代码中使用的逻辑像素, 在 CSS 规范中, 长度单位可以分为两类, 绝对(absolute)单位以及相对(relative)单位。px 是一个相对单位, 相对的是设备像素(device pixel)

9、DPR(设备像素比): 设备像素比 = 设备像素(物理像素) / 设备独立像素(逻辑像素)。(在Retina屏的iphone上, DPR为2, 1个css像素相当于2个物理像素)

常用单位

1. px

px其实就是像素的意思, 全称pixel, 也就是图像的基本采样单位。对于不同的设备, 它的图像基本单位是不同的, 比如显示器和打印机。而我们通常所说的显示器分辨率是指桌面设定的分辨率, 不是显示器的物理分辨率, 但是现在我们的桌面分辨率和物理分辨率几乎是一致的, 因为这样显示效果最佳。所以总的来说px就是对应我们显示器的分辨率。这样就会有个问题就是如果使用px的话我们就要根据不同电脑的分辨率来做自适应, 有点麻烦。

2. em

em是相对长度单位。相对于当前对象内文本的字体尺寸(如果没有设置本文尺寸, 那就是相对于浏览器默认的字体尺寸, 也就是16px), 这样计算的话。如果没有设置字体尺寸就是1em = 16px。如果使用em的话, 有个好的建议, 就是将body的font-size设置成62.5%, 也就是16px * 62.5% = 10px。这样的话1em = 10px, 方便我们计算。

3. rem

rem和em一样也是相对长度单位, 但是不一样的是rem始终都是相对html根元素。这样有个很大的有点就是使用rem后不会受到对象内文本字体尺寸的影响, 而且只需要改变根元素就能改变所有的字体大小。兼容性也是不错的, IE8以上版本和其他浏览器都已经支持, 是个做响应式页面的好选择

4. vw, vh(移动端开发)

vw和vh是视口(viewport units)单位, 何谓视口, 就是根据你浏览器窗口的大小的单位, 不受显示器分辨率的影响, 是不是很神奇, 这就代表了, 我们不需要顾虑到现在那么多不同电脑有关分辨率的自适应问题。

vw是可视窗口的宽度单位, 和百分比有点一样, 1vw = 可视窗口的宽度的百分之一。比如窗口宽度大小是1800px, 那么1vw = 18px。和百分比不一样的是, vw始终相对于可视窗口的宽度, 而百分比和其父元素的宽度有关。

vh就是可视窗口的高度了。

这边顺便提一下vmin和vmax, vmin是指选择vw和vh中最小的那个, 而vmax是选择最大的那个

栅栏布局

1. display:grid

2. 划分列: grid-template-columns

属性值: repeat(4, 1fr)

四个一排平铺, 一个大小为1fr, 单位也可以是其他的

3. 划分行: grid-template-rows

属性值: `repeat(4, 1fr)`

四个一排平铺, 一个大小为1fr

4. 划分区域: 父元素`grid-template-areas`, 子元素`grid-area`

5. 填充区域: `grid-auto-flow`

属性值:

<code>column</code>	优先填满第一列
<code>row</code>	优先填满第一行

6. 项目主轴对齐: `justify-items`

7. 项目侧轴对齐: `align-items`

8. 容器主轴对齐: `justify-content: stretch`

9. 容器侧轴对齐: `align-content`

10. 划分间距

写法:

```
grid-row-gap: 10px;  
grid-column-gap: 20px;  
row-gap: 20px;  
column-gap: 10px;  
gap: 10px 20px;
```

页面优化

1. 页面主题优化, 网站名称合理

2. 页面头部优化, `keywords`、`description`合理, 简洁, 明了, 关键字设置8-10个左右, 描述用一段文字介绍网站特点、内容

3. 图片优化, 每个图片都应该添加上`alt`属性, 如果该图片没有没什么意义, 则`alt`的值设置为空

4. PageRank (pr值, 友情链接), google提出的一个网站热度参数, 类似与流量的概念, 表示网站的重要程度