

html5 的新特性

1. 新的标签

1.1. 语义化标签

1.2. 多媒体标签

1.3. 新增的input类型

1.4. 新增的表单属性

2. css3的新特性

2.1. 新增选择器

2.1.1. 属性选择器

2.1.2. 结构伪类选择器

2.1.3. 伪元素选择器（重点）

2.2. 盒子模型

2.3. 其他特性

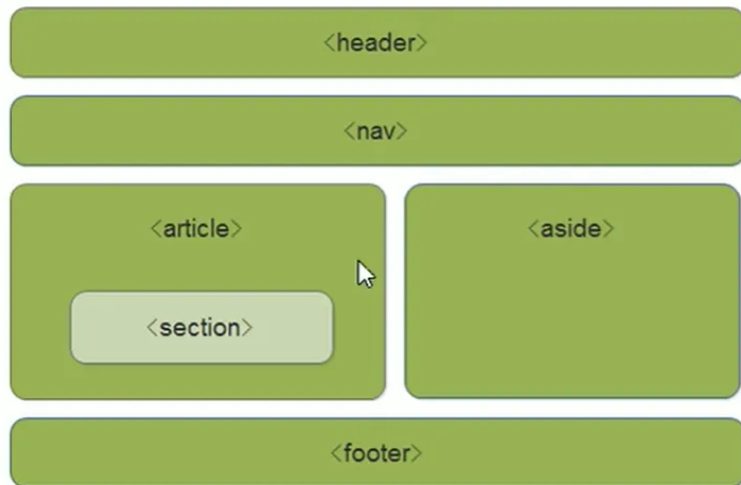
2.4. css3 过渡

1. 新的标签

1.1. 语义化标签

在IE9中需要把这些标签转换为块元素

- `<header>`：头部标签
- `<nav>`：导航标签
- `<article>`：内容标签
- `<section>`：定义文档某个区域
- `<aside>`：侧边栏标签
- `<footer>`：尾部标签



1.2. 多媒体标签

video 尽量使用MP4格式 audio尽量使用mp3格式

1. 音频：`<audio>`
2. 视频：`<video>`

loop循环播放 source是

谷歌浏览器把音频和视频自动播放禁止了

3. 多媒体标签总结

- 音频标签和视频标签使用方式基本一致
- 浏览器支持情况不同
- 谷歌浏览器把音频和视频自动播放禁止了
- 我们可以给视频标签添加 `muted` 属性来静音播放视频，音频不可以（可以通过JavaScript解决）
- 视频标签是重点，我们经常设置自动播放，不使用 `controls` 控件，循环和设置大小属性

1.3. 新增的input类型

type="email"

type="number"	限制用户输入必须为数字类型
type="tel"	手机号码
type="search"	搜索框

1.4. 新增的表单属性

required 表单不能为空

placeholder	提示文本	表单的提示信息，存在默认值将不显示
-------------	------	-------------------

可以通过以下设置修改placeholder的颜色

```
<style>
  input::placeholder {
    color: pink;
  }
</style>
</head>
<body>
  <form action="">
    <input type="search" name="" id="" required="required" placeholder="pink老师">
    <input type="submit" value="提交">
  </form>
</body>
```

multiple	multiple	可以多选文件提交
----------	----------	----------

2. css3的新特性

2.1. 新增选择器

类选择器、属性选择器、伪类选择器 权重为10

2.1.1. 属性选择器

```
/* 必须是input 但是同时具有 value这个属性 选择这个元素 */  
input[value] {  
    color: pink;  
}
```

选择属性=某值的 一些元素

```
input[type=text] {  
    color: pink;  
}
```

选择 属性值开头的某些元素

```
/* 选择首先是div 然后 具有class属性 并且属性值 必须是 icon开头的这些元素 */  
div[class^=icon] {  
    color: red;  
}
```

选择以xx结尾的

```
section[class$=data] {  
    color: blue;  
}
```

E[att]	选择具有 att 属性的 E 元素
E[att="val"]	选择具有 att 属性且属性值等于 val 的 E 元素
E[att^="val"]	匹配具有 att 属性且值以 val 开头的 E 元素
E[att\$="val"]	匹配具有 att 属性且值以 val 结尾的 E 元素
E[att*="val"]	匹配具有 att 属性且值中含有 val 的 E 元素

2.1.2. 结构伪类选择器

一般用于选择父级里面的第几个孩子

E:first-child	匹配父元素中的第一个子元素 E
E:last-child	匹配父元素中最后一个 E 元素
E:nth-child(n)	匹配父元素中的第 n 个子元素 E
E:first-of-type	指定类型 E 的第一个
E:last-of-type	指定类型 E 的最后一个
E:nth-of-type(n)	指定类型 E 的第 n 个

nth-child (n) 选择某个父元素的一个或多个特定的子元素

- n 可以是数字，关键字和公式
- n 如果是数字，就是选择第 n 个子元素，里面数字从1开始...
- n 可以是关键字：even 偶数，odd 奇数

选择了所有的

```
/* 3.nth-child(n) 从0开始 每次加1 往后面计算 这里面必须是n 不能是其他的字母 选择了所有的孩子*/
ol li:nth-child(n) {
    background-color: pink;
}
```

2n偶数 2n+1奇数 5n n+5 从第五个开始（包括第五个）往后 -n+5 前五个 （包括第五个）

```
/* nth-of-type 会把指定元素的盒子排列序号 */
/* 执行的时候首先看 :nth-child(1) 之后回去看 前面 div */
section div:nth-of-type(1) {
  background-color: blue;
}
```

区别：

1. **nth-child** 对父元素里面所有孩子排序选择（序号是固定的）先找到第n个孩子，然后看看是否和E匹配
2. **nth-of-type** 对父元素里面指定子元素进行排序选择。先去匹配E，然后再根据E找第n个孩子

2.1.3. 伪元素选择器（重点）

什么是伪元素 css伪元素代表了某个元素的子元素，这个子元素虽然在逻辑上存在，但却并不实际存在于文档树中。

::before	在元素内部的前面插入内容
::after	在元素内部的后面插入内容

- **before** 和 **after** 创建一个元素，但是属于行内元素
- 新创建的这个元素在文档树中是找不到的，所以我们称为**伪元素**
- 语法：**element::before {}**
- before 和 after 必须有 **content** 属性
- before 在父元素内容的前面创建元素，after 在父元素内容的后面插入元素
- **伪元素选择器**和**标签选择器**一样，权重为 1

使用场景 1：伪元素字体图标

```

, 里面放链接, 然后用定位
: content: '\e6a6'; border: 1px solid red; font-family: 'iconfont';
}

div::after {
    position: absolute;
    top: 10px;
    right: 10px;
    font-family: 'icomoon';
    content: '□';
    color: red;
    font-size: 18px;
}
</style>

```

fonts文件要在代码同级下, 然后style中要引入 font-style div中要加入font-family

使用场景2: 遮罩层

```

.tudou::before {
    content: '';
    /* 隐藏遮罩层 */
    display: none;
    position: absolute;
    top: 0;
    left: 0;
    width: 100%;
    height: 100%;
    background: rgba(0, 0, 0, .4) url(images/arr.png) no-repeat center;
}

/* 当我们鼠标经过了 土豆这个盒子, 就让里面before遮罩层显示出来 */
.tudou:hover::before {
    /* 而是显示元素 */
    display: block;
}

```


后面两种伪元素清除浮动算是第一种额外标签法的一个升级和优化。

```
.clearfix:after {  
  content: "";  
  display: block;  
  height: 0;  
  clear: both;  
  visibility: hidden;  
}
```

伪元素必须写的属性

插入的元素必须是块级

不要看见这个元素

核心代码清除浮动

不要看见这个元素

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

转换为块级元素并且一行显示

2.2. 盒子模型

CSS3 中可以通过 **box-sizing** 来指定盒模型，有2个值：即可指定为 **content-box**、**border-box**，这样我们计算盒子大小的方式就发生了改变。

可以分成两种情况：

1. box-sizing: content-box 盒子大小为 width + padding + border （以前默认的）
2. box-sizing: border-box 盒子大小为 width

如果盒子模型我们改为了box-sizing: border-box，那padding和border就不会撑大盒子了（前提padding和border不会超过width宽度）

```
<style>
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}
```

2.3. 其他特性

filter

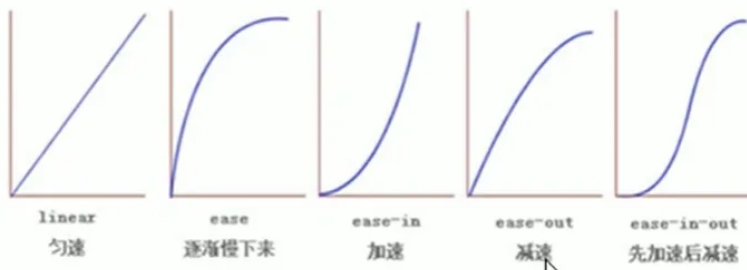
calc

```
width: calc(100% + 30px);
```

2.4. css3 过渡

transition: 要过渡的属性 花费时间 运动曲线 何时开始;

1. 属性：想要变化的 css 属性，宽度高度 背景颜色 内外边距都可以。如果想要所有的属性都变化过渡，写一个all 就可以。
2. 花费时间：单位是 秒（必须写单位）比如 0.5s
3. 运动曲线：默认是 ease（可以省略）
4. 何时开始：单位是 秒（必须写单位）可以设置延迟触发时间 默认是 0s（可以省略）



记住过渡的使用口诀：谁做过渡给谁加

```
div {
  width: 200px;
  height: 100px;
  background-color: pink;
  transition: all 0.5s;
  /* transition: 变化的属性 花费时间 运动曲线 何时开始; */
  /* transition: width .5s ease 0s, height .5s ease 1s; */
  /* 如果想要写多个属性，利用都好进行分割 */
  /* transition: width .5s, height .5s; */
  /* 如果想要多个属性都变化，属性写all就可以了 */
  /* transition: height .5s ease 1s; */
}
div:hover {
  width: 400px;
  height: 200px;
}
```

进度条案例：

```
.bar_in {  
  width: 50%;  
  height: 100%;  
  background-color: red;  
  /* 谁做过渡给谁加 */  
  transition: all .7s;  
}  
  
.bar:hover .bar_in {  
  width: 100%;  
}  
  
</style>
```