

HTML5&CSS3 —

一、什么是 HTML5

1. HTML5 的概念与定义

- 定义：HTML5 定义了 HTML 标准的最新版本，是对 HTML 的第五次重大修改，号称下一代的 HTML
- 两个概念：
 - 是一个新版本的 HTML 语言，定义了新的标签、特性和属性
 - 拥有一个强大的技术集，这些技术集是指：HTML5、CSS3、javascript，这也是广义上的 HTML5

2. HTML5 拓展了哪些内容

- 语义化标签
- 本地存储
- 兼容特性
- 2D、3D
- 动画、过渡
- CSS3 特性
- 性能与集成

3. HTML5 的现状

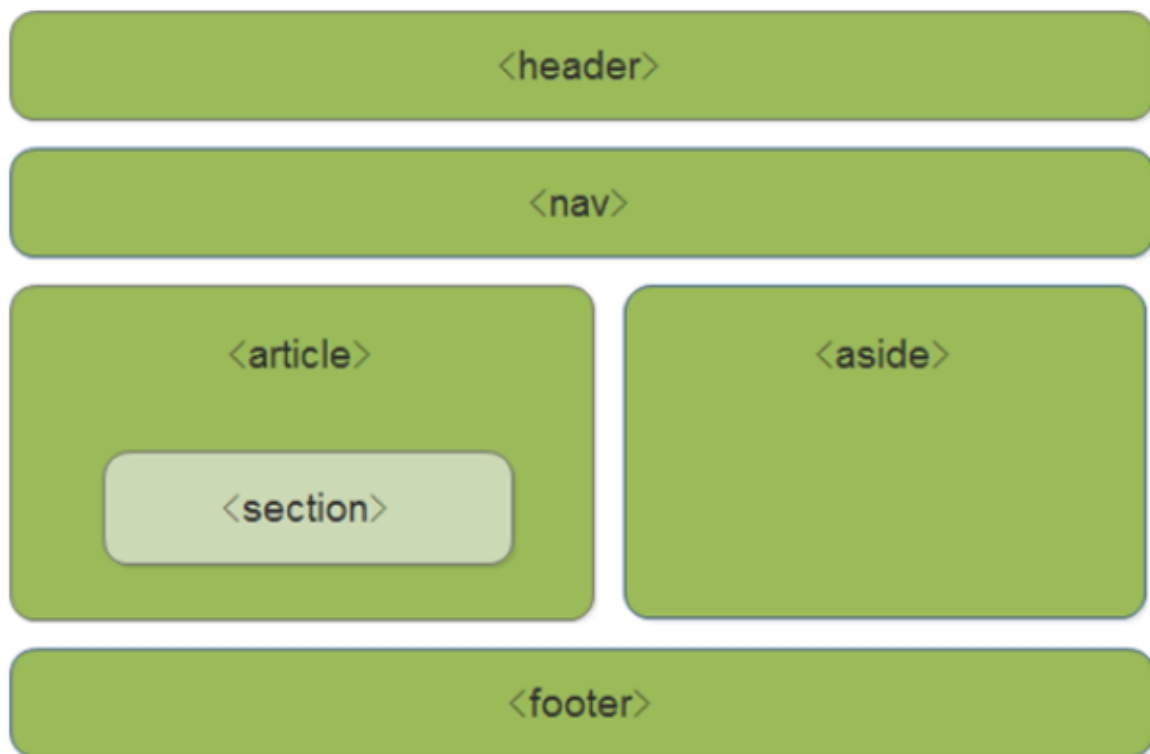
绝对多数新的属性，都被浏览器所支持，最新版本的浏览器已经开始陆续支持最新的特性，总的来说：HTML5 已经是大势所趋

二、HTML5 新增标签

1. 什么是语义化

2. 新增了那些语义化标签

- header --- 头部标签
- nav --- 导航标签
- article --- 内容标签
- section --- 块级标签
- aside --- 侧边栏标签
- footer --- 尾部标签



3. 使用语义化标签的注意

- 语义化标签主要针对搜索引擎
- 新标签可以使用一次或者多次
- 在 `IE9` 浏览器中，需要把语义化标签都转换为块级元素
- 语义化标签，在移动端支持比较友好，
- 另外，`HTML5` 新增了很多的语义化标签，随着课程深入，还会学习到其他的

三、多媒体音频标签

1. 多媒体标签有两个，分别是

- 音频 -- `audio`
- 视频 -- `video`

2. `audio` 标签说明

- 可以在不使用标签的情况下，也能够原生的支持音频格式文件的播放，
- 但是：播放格式是有限的

3. `audio` 支持的音频格式

- `audio` 目前支持三种格式

格式	IE9	Firefox3.5	Opera10.5	Chrome3.0	Safari3.0
Ogg Vorbis		√	√	√	
MP3	√			√	√
Wav		√	√		√

4. `audio` 的参数

属性	值	描述
autoplay	autoplay	如果出现该属性，则音频在就绪后马上播放。
controls	controls	如果出现该属性，则向用户显示控件，比如播放按钮。
loop	loop	如果出现该属性，则每当音频结束时重新开始播放。
src	url	要播放的音频的 URL。

5、audio 代码演示

```
<body>
<!-- 注意：在 chrome 浏览器中已经禁用了 autoplay 属性 -->
<!-- <audio src="./media/snow.mp3" controls autoplay></audio> -->

<!--
    因为不同浏览器支持不同的格式，所以我们采取的方案是这个音频准备多个文件
-->
<audio controls>
    <source src="./media/snow.mp3" type="audio/mpeg" />
</audio>
</body>
```

四、多媒体视频标签

1. video 视频标签

- 目前支持三种格式

格式	IE	Firefox	Opera	Chrome	Safari
Ogg	No	3.5+	10.5+	5.0+	No
MPEG 4	9.0+	No	No	5.0+	3.0+
WebM	No	4.0+	10.6+	6.0+	No

2. 语法格式

```
<video src="./media/video.mp4" controls="controls"></video>
```

3. video 参数

属性	值	描述
autoplay	autoplay	视频就绪自动播放（谷歌浏览器需要添加muted来解决自动播放问题）
controls	controls	向用户显示播放控件
width	pixels(像素)	设置播放器宽度
height	pixels(像素)	设置播放器高度
loop	loop	播放完是否继续播放该视频，循环播放
preload	auto（预先加载视频） none（不应加载视频）	规定是否预加载视频(如果有了autoplay 就忽略该属性)
src	url	视频url地址
poster	Imgurl	加载等待的画面图片
muted	muted	静音播放

4. video 代码演示

```
<body>
  <!-- <video src="./media/video.mp4" controls="controls"></video> -->

  <!-- 谷歌浏览器禁用了自动播放功能，如果想自动播放，需要添加 muted 属性 -->
  <video controls="controls" autoplay muted loop poster="./media/pig.jpg">
    <source src="./media/video.mp4" type="video/mp4">
    <source src="./media/video.ogv" type="video/ogg">
  </video>
</body>
```

5. 多媒体标签总结

- 音频标签与视频标签使用基本一致
- 多媒体标签在不同浏览器下情况不同，存在兼容性问题
- 谷歌浏览器把音频和视频标签的自动播放都禁止了
- 谷歌浏览器中视频添加 muted 标签可以自己播放
- 注意：重点记住使用方法以及自动播放即可，其他属性可以在使用时查找对应的手册

五、新增 input 标签

属性值	说明
type="email"	限制用户输入必须为Email类型
type="url"	限制用户输入必须为URL类型
type="date"	限制用户输入必须为日期类型
type="time"	限制用户输入必须为时间类型
type="month"	限制用户输入必须为月类型
type="week"	限制用户输入必须为周类型
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
type="color"	生成一个颜色选择表单

六、新增表单属性

属性	值	说明
required	required	表单拥有该属性表示其内容不能为空，必填
placeholder	提示文本（占位符）	表单的提示信息，存在默认值将不显示
autofocus	autofocus	自动聚焦属性，页面加载完成自动聚焦到指定表单
autocomplete	off / on	当用户在字段开始键入时，浏览器基于之前键入过的值，应该显示出在字段中填写的选项。 默认已经打开，如 autocomplete="on" 关闭 autocomplete="off" -需要放在表单内同时加上name属性 -同时成功提交
multiple	multiple	可以多选文件提交

七、CSS3 属性选择器(上)

1. 什么是 CSS3

- 在 CSS2 的基础上拓展、新增的样式

2. CSS3 发展现状

- 移动端支持优于 PC 端
- CSS3 目前还草案，在不断改进中
- CSS3 相对 H5，应用非常广泛

3. 属性选择器列表

选择符	简介
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

4. 属性选择器代码演示

```
button {
  cursor: pointer;
}
button[disabled] {
  cursor: default;
}
```

八、CSS3 属性选择器(下)

1. 代码演示

```
input[type=search] {
  color: skyblue;
}

span[class^=black] {
  color: lightgreen;
}

span[class$=black] {
  color: lightsalmon;
}

span[class*=black] {
  color: lightseagreen;
}
```

九、结构伪类选择器

1. 属性列表

选择符	简介
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

2. 代码演示

```
ul li:first-child {
  background-color: lightseagreen;
}

ul li:last-child {
  background-color: lightcoral;
}

ul li:nth-child(3) {
  background-color: aqua;
}
```

十、nth-child 参数详解

1. nth-child 详解

- 注意：本质上就是选中第几个子元素
- n 可以是数字、关键字、公式
- n 如果是数字，就是选中第几个
- 常见的关键字有 even 偶数、odd 奇数
- 常见的公式如下(如果 n 是公式，则从 0 开始计算)
- 但是第 0 个元素或者超出了元素的个数会被忽略

公式	取值
2n	偶数
2n+1	奇数
5n	5 10 15 ...
n+5	从第5个开始 (包含第五个) 到最后
-n+5	前5个 (包含第5个) ...

2. 代码演示

```
<style>
  /* 偶数 */
  ul li:nth-child(even) {
    background-color: aquamarine;
  }

  /* 奇数 */
  ul li:nth-child(odd) {
    background-color: blueviolet;
  }

  /* n 是公式, 从 0 开始计算 */
  ul li:nth-child(n) {
    background-color: lightcoral;
  }

  /* 偶数 */
  ul li:nth-child(2n) {
    background-color: lightskyblue;
  }

  /* 奇数 */
  ul li:nth-child(2n + 1) {
    background-color: lightsalmon;
  }

  /* 选择第 0 5 10 15, 应该怎么选 */
  ul li:nth-child(5n) {
    background-color: orangered;
  }

  /* n + 5 就是从第5个开始往后选择 */
  ul li:nth-child(n + 5) {
    background-color: peru;
  }

  /* -n + 5 前五个 */
  ul li:nth-child(-n + 5) {
    background-color: tan;
  }
</style>
```

十一、nth-child 和 nt-of-type 的区别

1. 代码演示

```
<style>
```



```

div :nth-child(1) {
  background-color: lightblue;
}

div :nth-child(2) {
  background-color: lightpink;
}

div span:nth-of-type(2) {
  background-color: lightseagreen;
}

div span:nth-of-type(3) {
  background-color: #fff;
}
</style>

```

2. 区别

- `nth-child` 选择父元素里面的第几个子元素，不管是第几个类型
- `nth-of-type` 选择指定类型的元素

十二、伪元素选择器

1. 伪类选择器

选择符	简介
<code>::before</code>	在元素内部的前面插入内容
<code>::after</code>	在元素内部的后面插入内容

2. 伪类选择器注意事项

- `before` 和 `after` 必须有 `content` 属性
- `before` 在内容前面，`after` 在内容后面
- `before` 和 `after` 创建的是一个元素，但是属于行内元素
- 创建出来的元素在 `Dom` 中查找不到，所以称为伪元素
- 伪元素和标签选择器一样，权重为 1

3. 代码演示

```

<style>
  div {
    width: 100px;
    height: 100px;
    border: 1px solid lightcoral;
  }

  div::after,
  div::before {

```

```

        width: 20px;
        height: 50px;
        text-align: center;
        display: inline-block;
    }
    div::after {
        content: '德';
        background-color: lightskyblue;
    }

    div::before {
        content: '道';
        background-color: mediumaquamarine;
    }
</style>

```

十三、伪元素的案例

1. 添加字体图标

```

p {
    width: 220px;
    height: 22px;
    border: 1px solid lightseagreen;
    margin: 60px;
    position: relative;
}
p::after {
    content: '\ea50';
    font-family: 'icomoon';
    position: absolute;
    top: -1px;
    right: 10px;
}

```

十四、2D 转换之 `translate`

1. 2D 转换

- 2D 转换是改变标签在二维平面上的位置和形状
- 移动: `translate`
- 旋转: `rotate`
- 缩放: `scale`

2. `translate` 语法

- x 就是 x 轴上水平移动
- y 就是 y 轴上水平移动

```
transform: translate(x, y)
transform: translateX(n)
transform: translateY(n)
```

3. 重点知识点

- 2D 的移动主要是指 水平、垂直方向上的移动
- `translate` 最大的优点就是不影响其他元素的位置
- `translate` 中的100%单位，是相对于本身的宽度和高度来进行计算的
- 行内标签没有效果

4. 代码演示

```
div {
  background-color: lightseagreen;
  width: 200px;
  height: 100px;
  /* 平移 */
  /* 水平垂直移动 100px */
  /* transform: translate(100px, 100px); */

  /* 水平移动 100px */
  /* transform: translate(100px, 0) */

  /* 垂直移动 100px */
  /* transform: translate(0, 100px) */

  /* 水平移动 100px */
  /* transform: translateX(100px); */

  /* 垂直移动 100px */
  transform: translateY(100px)
}
```

十五、让一个盒子水平垂直居中

- 看代码

十六、2D 转换 rotate

1. rotate 旋转

- 2D 旋转指的是让元素在二维平面内顺时针或者逆时针旋转

2. rotate 语法

```
/* 单位是: deg */
transform: rotate(度数)
```

3. 重点知识点

- `rotate` 里面跟度数，单位是 `deg`

- 角度为正时，顺时针，角度为负时，逆时针
- 默认旋转的中心点是元素的中心点

4. 代码演示

```
img:hover {  
  transform: rotate(360deg)  
}
```