

新增标签

结构标签

块状元素 — 有意义的 div

<article>

<header> 级别高

<nav>

<section> 一个区域

<aside> 例边栏

<hgroup> 一个区组的相关信息

<figure> 一组媒体内容以及标题

<figcaption> figure 元素的标题

<footer>

<dialog> 一个对话框 open

多媒体标签

不使用插件、操作媒体文件

• <video>

• <audio>

音频

• <source> 媒体

autoplay

loop = "-1" 重复

controls

<audio>

<source> 单标签

mp3 => type = "audio/mpeg"

</audio>

- `canvas` 画布
- `<embed>` 外部的可交互的内容或插件，flash

状态标签

		optimum 标准
<code><meter></code>	实时状态显示	<code>value min max low high</code>
<code><progress></code>	任务进程	

列表标签

`<datalist>` 为 `input` 定义一个下拉，配合 `option`

```
<input list="">
<datalist id="">
```

`<details>` 一个元素的详细内容 `open`
配合 `summary`

注释标签

`<ruby>` 定义注释或者标

`<rt>` 定义对 `ruby` 的注释内容文本

```
<ruby> <rt></rt> <ruby>
```

```
<rp> (</rp> <rt> ... </rt><rp>) </rp>
```

其它

`<mark>` 黄色选中

`<output>` 定义一些输出类型
计算表单结果 配合 `oninput` 事件

<form> oninput : 实时监听文本框输入变化

重定义标签

<dd>

<dt>

与 details , figure , dialog 一起用

<menu> 定义用户界面的菜单，配合 command 或 menuitem

页面布局

<section>

<aside> </aside>

<article> </article>

</section>

拔河效应

- 应用于 position 中，脱离文档流

top: 0

bottom: 0

⇒ 垂直居中

margin: auto

- first - of - type

属性变化

Input

- <input type="email" name="email">
- type="url"
- type="tel"
- type="number"

- Date Pickers

data 年月日
month 年月

week
time

datetime 不兼容
datetime-local

- range max, min

- search

- color

表单属性

- autocomplete

<form ~="on"> 自动完成填写
<input ~>

- autofocus = "autofocus"

multiple 输入域中可选多个值 multiple = "multiple"
email, file

- placeholder

- required = "required" 必填

链接属性

- sizes <link>

- target <base target="" > base 属性 head 中
-blank 新窗口

- <a> 超链接

media = " " handheld tv

hreflang = "zh" 设置语言

rel = "external" 设置引用

↓
外部链接 (当前文档与被链接文档的关系)

其它

- script

⇒ defer : 当页面加载完后，才会执行脚本

⇒ async : 一旦脚本可用，则会异步执行

- ol 有序列表

start

reversed 倒序

- <html manifest = "cache.manifest" > 页面离线应用

- <style scoped> </style> 内嵌 CSS

音视频

video

- mp4, webm, ogv

height, width, autoplay, loop

poster : 视频封面

muted : 静音

controls

source

- 兼容

可以放好几个 src

Video API Javascript 函数 ↓

• video. play() chrome 不允许自动有声播放

• pause()

返回视频总长度

• currentTime

直接快进

= xx ; 直接加载到 xx 秒

返回当前播放长度

• src

• volume 设置返回当前音量

可用 input 的 range , oninput 值发生变化时

• controls

• muted

• networkState

返回当前网络状态

- { 1 视频已加载, 未使用网络
- 2 正在加载
- 3 未找到
- 0 未初始化

• currentSrc

返回当前 URL, 无法赋值

(可以 播放时)

• ended

是否结束

监听

video. addEventListener ('ended', function(){})

- loop 是否重新播放
- playbackRate 查看或设置播放速度
- readyState 返回当前视频就绪状态
 - 0 : 没有视频就绪信息.
 - 1 : 有数据 但快不足以支撑
 - 2: 马上无法播放
 - 3. 正在缓冲
 - 4 : 可以播放
- timeupdate 视频播放状态
- seeked 移动/跳跃到视频新位置时
已经完成操作
- seeking 开始移动时
- volumechange 声音发生改变时
- Request Full screen 全屏
 - webkitRequestFullScreen chrome
 - mozRequestFullScreen Firefox

- load() 重新加载
- canplay 视频已准备好可以开始播放

audio

- mp3、wav、ogg → safari 不兼容
- controls
- 可以用 JS 生成 audio 对象
 - new Audio()
 - chrome 不能直接 play()，要加事件交互
- 标签属性
 - src
 - controls
 - autoplay
 - loop 循环
 - muted (不会自动播放)

audio API

- play()
- pause()
- duration 音频长度
- canplay
- current Time
- volume
- network State

- currentSrc
- ended
- loop
- playbackRate 播放倍速
- readyState 音频状态
- timeupdate 监听音频播放
- { seeked
 | seeking
- volumechange
- webkitrequestFullScreen —— 必须只用html的audio标签
- load 重新加载

videojs

第三方框架

1. 引入 CSS 和 js
2. class = video-js
3. data-setup = "{}"
4. width, height
5. vjs-big-play-centered
6. poster
7. preload = "auto"

播放 videojs

videojs('id'). ...

1. currentTime() 设置时间，传参
2. ready()
3. duration()
4. buffered()
5. volume()
6. width(), height() 获取宽高
7. 监听
ended, pause, play, volumechange
timeupdate seeking

表单验证

基本表单验证特性

1. input type = file
要加上 enctype
发送到服务器之前 对表单数据进行编码
2. label for = " " ↓
input id = " " ;
可以在 radio, checkbox 中使用
3. pattern 正则表达式

约束验证 API

willValidate

checkValidity()

validity

validationMessage

setCustomValidity() → 自定义验证信息

1. validity 属性输出 ValidityState 对象



step="0.01" value="0.02" 保留小数

2. checkValidity() 返回 true or false

3. setCustomValidity()

伪类

自带验证美化

1. : required

: optional

2. : in-range

: out-of-range

3. :valid

: invalid

4. :read-only

: read-write

选择器

+ 相邻兄弟

~ 所有兄弟

- :focus { outline }

- event.preventDefault() 阻止默认气泡

- querySelectorAll (" :invalid ") 选中所有