**HTML5核心**

1. From API

1.1、新表单元素（录入数据相关）

全部都是由input标签组成

1.1.1 email

1.1.2 url 注意：输入的内容必须以http://开头

1.1.3 search 普通文本框，特点：多了一个可以删除全部文本的的删除按钮

1.1.4 rang 范围 

属性：min：最小值

Max：最大值

Step：步长

Value：值

1.1.5 number 只接收数字 

属性：min：最小值

Max：最大值

Step：步长

Value：值

1.1.6 color 通过windows的调色板选取颜色

1.1.7 date 弹出日历控件，选择年月日

值：2015-12-26

1.1.8 mouth 弹出日历控件，选择年月日

值：2015-12

1.1.9 week 弹出日历控件，选择年月日

值：2015-W13

1.1.10 datetime 弹出日历控件，选择年月日

值：2015-12-26T09:15Z

1.1.11 datetime-local

1.2新表单属性

1.2.1 required 必须的，非空验证 <input type=’text’ required=’false’>

1.2.2 multiple 在一个表单元素中，允许录入多段类型相同的数据，由‘，’由 作为分隔符

1.2.3 pattern 根据用户指定的正则表达式，对数据进行验证

1.2.4 autofocus 自动获取焦点

1.2.5 form 在表单之外，提交表单元素 <form id=”name”></form>

<input form=”name”></input>

1.2.6 placeholder 网页加载时，表单元素上会默认显示一段文字

注意：只有在用户录入数据时，placeholder的值才会消失

1.3新表单元素（显示数据相关）

1.3.1 datalist 构建一个数据列表，可以提供给文本框类型的控件使用

<option value=””>Display</option>

<option value=”” label=”Display”></option>

与datalist相关联：

<input type=”text” list=”datalistID”>

1.3.2 progress 进度条控件

重要属性：max：进度完成后的值

Value：当前的进度值

注意：多数情况下，要与setInterval或setTimeout联系

1.3.3 meter 刻度表

属性：min：设置整个刻度的下限值

Max：设置整个刻度的上限值

Value：当前的刻度值

Low：设定合理范围下限值，大于等于min

High：设定合理范围上限值，小于等于max

Optimum：设置最佳位置值

1.3.4 output 用于显示表单元素间，所计算得到的值

属性：for：关联相互计算的控件id值

Value：显示的值

======================================================================

1.4.新API方法、事件、

1.4.1 setCustomvalidity();

作用：自定义错误信息，一旦设置该属性，则不允许提交。一旦提交，显 示自定义错误信。

清空错误验证消息，一旦数据被修改正确人，需要通过 setCustomvalidity(“”);清空错误

验证信息，使得表单 允许被提交

语法：input.setCustomValidity(“错误消息”);

一旦设置了错误消息，表单是不允许被提交的。

如果想清除错误消息(清除错误)：input.setCustomValidity(“”);

1.4.2 checkValidity();

作用：在不提交表单的情况下对表单或者表单元素进行验证

如果有问题的话，返回值为false，否则返回值为true

1.4.3 invalid事件

每当用户提交表单时，如果检测到无效域时，就会触发invalid事件

它主要关注发生错误的元素

使用：

Document.forms[0].addEventListener(“invalid”,form\_invalid,true);

Function form\_invalid(e){

Var elem = e.target;//产生错误的元素

//调整elem

}

1.5.新属性

1.5.1 validateState对象是通过validity属性获取的，并且可以通过它的几种状态 对表单元素进行验证

语法：var varCheck = elem.validity

状态：表单API提供的一组状态检查属性，可用于实现自定义验证过程

1. 如何获取ValidateState

foremElement.validity;

1. 错误状态值
2. 判断是否通过所有的验证规则

formElement.validity.valid

通过所有的验证后，返回值为true，否则为false

1. 判断是否为空

formElement.validity.valueMissing

注意：验证有required属性的控件，值是否为空值为空，返 回true，否则返回false

1. 判断数据是否违反类型

formElement.validity.typeMismatch

看输入的数据，与type属性是否相符。若不符合返回true， 否则返回false

1. 判断数据是否违反pattern属性的约束

formElement.validity.patternMismatch，违反为true，否则false

1. 判断数据是否违反了定义的step

多数使用在<input type=’number/’>类型上

formElement.validity.stepMismatch，违反为true，否则为false

1. 输入的数据长度超出了maxlength所定义的长度

formElement.validity.tooLong

1. 输入的值小于定义的min值

formElement.validity.rangeUnderflow

1. 输入的值大于定义的max值

formElement.validity.rangeOverflow

1. 判断表单元素是否已经设置了自定义错误消息

formElement.validity.customError

Valid状态

valueMissing状态

typeMismatch状态 适合验证邮箱

patternMismatch状态

tooLong状态 相当于maxlength属性

rangUnderflow状态 输入内容小于min属性声明的值,则该状态为 true

rangOverflow状态 输入内容大于min属性声明的值,则该状态为 true

stepMismatch状态 如果给定的值与min，max，step不一致，那么该 状态就是true

customError状态 如果设置了自定义错误，那么该状态是true

formElement.setCustomvalidity(“格式不正确”)；

formElement.customError；返回值为true

formElement.setCustomvalidity(“”)；

formElement.customError；返回值为false

1. 媒体（视频、音频）API

H5之前播放视频 <embed src=””/>标签

H5标签

<video src = ‘abc.mp4’></video>

<video>

<source src=”abc.swf”/>

<source src=”abc.mp4”/>

</video>

Video元素

属性：

Controls：出现控件界面

Autoplay：设置这个属性时，加载视频后自动播放

Loop：设置这个属性时，浏览器会反复播放该视频

Poster：该属性指定一个URL，在视频等待播放时显示一副图像

Preload：该属性可以设置三个值

None：不缓存视频，为了减少不必要的流量

Metadata：推荐浏览器抓取一些资源的信息

Auto：默认值，要求浏览器尽可能快的下载视频

常用事件

Progress：用于更新媒体的下载进度，会周期性的触发

Conplaythrough：当整个媒体可以顺利播放时，就会触发这个事件

Canplay：不考虑整体状态，只要下载了一定的可放帧会触发这个事件

Ended：媒体到达末尾时触发

Pause：媒体暂停时触发

Play：媒体开始播放时触发

Error：媒体播放出现错误时触发

常用方法

Play（）：播放媒体文件

Pause（）：暂停播放

Load（）：加载媒体文件，动态应用程序可使用该方法提前加载

canPlayType（type）：查看浏览器是否支持这种文件格式的媒体文件

H5新增的针对视频元素的处理属性如下：

Paused：媒体处于暂停或未播放状态，这个值为true

Ended：如果媒体已经结束播放，这个值为true

Duration：返回媒体时长，以秒为单位

currentTime：获取或设置媒体播放位置

Audio元素

Src：指定播放文件的URL，可通过<source>

Controls：可激活各浏览器提供的默认页面

Autoplay：加载音频后自动播放

Loop：设置这个属性时，浏览器会反复播放该音频

Preload：缓存设置

1. Canvas

Canvas元素

<canvas></canvas>

在使用<canvas>元素时，要调用getContext()方法，其目的是得到画布的绘图上下文，通 过这个引用，就可以使用其他的API进行绘图操作

语法：

Function initial(){

Var elem = document.getElementById(“canvas”);

Var canvas = elem.getContext(“2d”);

}

1、绘制矩形方法（生成基础形状方法）：

--fillRect（x,y,width,height） （x，y）是指矩形左上角的顶点坐标

--strokeRect(x,y,width,height) 绘制边框

--clearRect(x,y,width,height) 擦除指定区域

2、颜色相关：可通过以下属性指定绘图颜色

--strokeStyle：声明形状线条的颜色

elem.strokeStyle=”rgb(255,0,0)”

--fillStyle:声明形状内部区域的颜色

elem.strokeStyle=”rgba(255,0,0,0.3)”

--globalAlpha：透明度属性。可以设置画布上图行的透明度

3、渐变相关：

语法：

--createLinearGradient(x1,y1,x2,y2)；在画布上创建一个渐变对象

--createRadialGradient(x1,y1,x2,y2,r2):使用两个圆形在画布上创建一个渐变对象

--addColorStop（position，color）：指定渐变颜色值

Eg：var ctx = elem.createLinearGradient(0,150,400,150)；

ctx .addColorStop(0,’red’);

ctx .addColorStop(0.2,’yellow’);

ctx .addColorStop(1,’blue’);

1. 创建路径（复杂图形）

方法：

--beginPath():开始一个新的形状描述，创建路径之前，必须先调用此方法

--closePath():关闭路径，用直线将最后一个点与原点相连，如果想保留开放路径， 则不需要调用这个方法

--stroke():将路径绘制为轮廓形状

--fill():将路径绘制为实心形状。使用该方法时可以不用closePath关闭路径。该方法会通过直线连接最后一个点与第一个点实现封闭

--clip():在上下文中设置裁剪区域

--moveTo():将笔尖移到指定的位置

--lineTo()：绘制一条直线，连接当前笔尖位置到x和y属性声明的新位置

--rect(x,y,with,height):生成一个矩形路径

--arc(x,y,radius,startAngle,endAngle,irection):这个方法可以在位置(x,y)上生成弧线或圆形，半径和弧度分别由属性指定，direction是个布尔类型，表示顺时针或逆时针方向

画布方法

状态方法：

--save();保存当前画布属性，状态

--restore();恢复画布属性、状态

转换方法：

--scale();缩放当前绘图更大或更小

--translate()；重新映射画布上的(0,0)位置

--rotate();旋转当前画布，公式为degrees\*Math.PI/180;

线型：

属性：--lineCap：指定线条端点形状，支持的值有以下三个

Butt:默认，向线条的每个末端添加平直的边缘

Round：向线条的每个末端添加圆形线帽

Square：向线条的每个末端添加正方向线帽

注意：round和square会使线条略变微长

--lineJoin:指定两条线之间的连接点形状，可选值

Round：创建圆角

Bevel：创建斜角

Miter：默认，创建尖角

--miterLimit：与lineJoin一起使用，当lineJoin设置为miter时，可用于确定线条交接点的延伸范围

1. 拖拽相关API 离线存储API
2. Geolocation API

Baidu Geolocation

Baidu Map