一、HTML5基本内容

1、HTML5的声明

<!DOCTYPE html>

2、HTML5的特点

移动端浏览器相比PC端浏览器对H5的支持更好

3、Flash技术

场景

视频网站、网页游戏

问题

浏览器原生不支持Flash，需要第三方组件

性能极差

不支持移动端

二、HTML5的新表单

1、INPUT新类型

(1)email类型 -- 判断字符串中是否包含"@"符号

例如 <input type="email">

输入框内不能以@开始，也不能以@结束

(2)搜索类型 -- search

<input type="search">

(3)URL类型 -- 判断字符串中是否包含"http:"

以"http:"开始，或以"http:"结束都可以验证通过

(4)电话号码类型 -- tel

只有在手机端浏览器访问时才有效果

(5)数字类型 -- number

允许输入非数字内容，但不允许提交

属性

min，设置数字的最小值

max，设置数字的最大值

step，设置步长

表示每次增加或减少的量值

当设置了min和max时，允许输入范围外的值，但不允许提交

(6)范围 -- range，滑动条效果

属性

min、max、step，与number一样

value，当前默认值

(7)颜色 -- color

(8)日期 -- date

格式 yyyy/MM/dd

周类型 -- week

月份类型 -- month

2、表单新元素

(1)datalist -- 下拉列表，与select效果相似

用法：

配合input元素使用，在input元素中定义list属性(值为datalist元素的id值)

例如 <datalist id="cities">

<option>请选择</option>

<option>北京</option>

<option>上海</option>

</datalist>

<input type="text" list="cities">

相对于select元素的好处：

数据与结构的分离

(2)progress -- 进度条

max，设置进度条的最大值

value，设置进度条当前初始值

(3)meter -- 刻度

min、max、value

high、low，设置预警值

(4)output -- 输出框

for，与要输出的元素进行关联

3、表单新属性

(1)placeholder

实现input输入框的默认提示信息，相比value属性值更好用

(2)autofocus

自动获取焦点

只需要定义属性名，没有属性值

(3)multiple

允许输入框输入多个值，多个值之间使用","隔开

只需要定义属性名，没有属性值

(4)form

允许表单元素定义在表单之外

它的值是相关联表单的id属性值

4、表单新验证

(1)验证属性

①required，验证是否为空

返回true，表示当前元素值不为空；反之

②pattern，验证正则表达式

定义正则表达式时，不能添加"/ /"

正则表达式不能验证是否为空，还得依靠required

例如 <input type="text" pattern="^[0-9]{6,8}$">

用来验证6-8位的数字密码

③min和max，验证最小值和最大值

只和number类型的input元素配合使用

④minlength和maxlength，验证最小长度和最大长度

maxlength的效果是限制最大长度，输入的内容长度不能大于maxlength的值

⑤validity，HTML5提供表单验证的接口

通过该属性得到validityState对象

该对象提供了一系列的有效状态

有效状态可用于表单验证

elem.validity，得到该对象

(2)有效状态

①valid，表示验证是否通过

返回true，表示验证通过；反之

②valueMissing，表示值是否为空

返回true，表示值为空；反之

该状态配合required使用

③typeMismatch，表示元素类型是否匹配

返回true，表示元素类型不匹配；反之

该状态配合email、url、number等使用

④patternMismatch，表示正则表达式是否匹配

返回true，表示不匹配；反之

该状态配合pattern使用

⑤tooLong，表示元素内容长度是否过长

返回true，表示过长；反之

该状态配合maxlength使用

⑥rangeUnderflow，表示元素值是否小于min的值

返回true，表示元素值小于min的值；反之

该状态配合min属性使用

⑦stepMismatch，表示元素值与step值是否不符

返回true，表示不符；反之

该状态配合step使用

⑧customError，自定义错误

自定义错误提示信息

setCustomValidity(自定义错误提示信息);

一旦调用该方法，默认认为就是错的

上述所有的有效状态都返回错误值

如何解决：验证正确时，调用该方法，将错误信息置为空

三、视频处理

1、HTML5支持的视频格式

(1)MP4，扩展名.mp4

(2)OGG，扩展名.ogv，手机端的视频格式

(3)WebM，目前唯一一个支持【超高清】的视频格式

该视频格式出自Goggle公司

2、如何处理视频 -- <video>元素

(1)引入单个视频文件

属性：

src，指定视频文件的地址

width、height

视频的宽高【比例】是不会改变的

在video元素内需要定义不支持的视频格式的提示内容

(2)引入多个视频文件

<source/>元素可以引入多个格式的视频文件

保证各浏览器的兼容性

3、video的属性

(1)autoplay，自动播放

(2)controls，提供控制面板

(3)loop，循环播放

(4)poster，播放之前显示一张图片

(5)preload，预加载(视频)

none，不加载

auto，默认值，自动(尽快加载完毕)

metadata，只加载除视频之外的信息(比如宽、高)

4、video的高级用法

(1)方法

①play()，播放

②pause()，暂停

③load()，加载

④canPlayType()，判断浏览器是否支持该视频格式

(2)事件

①play，播放视频时触发

②pause，暂停视频时触发

③ended，结束播放时触发

④error，播放错误时触发

⑤canplay，可播放(有预加载内容)时触发(不考虑整体)

⑥canplaythrough，整体播放顺利时触发

⑦progress，下载进度

(3)属性

①paused，判断视频当前是否暂停

返回true，表示当前为暂停状态；反之

②ended，判断视频当前是否结束播放

返回true，表示当前为结束播放状态；反之

③duration，当前视频的总时长

④currenTime，获取或设置当前视频播放的位置

四、音频处理

1、目前HTML页面能够实现的音频处理

将Windows Media Player播放器内置在页面中

使用Flash技术

2、HTML5支持的音频格式

mp3、ogg、wav

3、如何使用音频 -- <audio>元素

跟video类似

4、属性

autoplay、controls、loop、preload

5、高级用法

(1)事件

play、pause

(2)方法

play()、pause()

(3)属性

paused

五、视频及音频注意事项

IE8及之前的版本都不支持

解决视频处理：提供Js库(video.js)

官网：http://videojs.com/

视频或音频所支持的格式，不同浏览器情况可能不同

六、Canvas(画布)

1、基本内容

替代引入的图片(图形)

完成HTML页面中的图形绘制

实现网络游戏或单机游戏(网页游戏)

在HTML页面中绘制图表

饼状图、柱状图、折线图等

\*Canvas作为HTML页面的元素出现，在Canvas绘制的图形与HTML页面是无关的

所以，不能利用DOM定位绘制的图形；元素的事件也不能绑定绘制的图形

2、如何使用Canvas

(第1步)在HTML页面中定义canvas元素

设置canvas元素的高度和宽度

style属性，绘制的图形会被拉伸

width、height属性，绘制的图形不会被拉伸

(第2步)在Js代码中

①获取HTML页面中的canvas元素

例如 var canvas=document.getElementById("canvas");

②通过getContext()方法创建画布对象

该方法接收一个参数：2d(二维图形)或3d(三维图形)

参数的类型为字符串

2d或3d中的"d"必须小写

目前基本上都是2d效果

例如 var context=canvas.getContext("2d");

③利用画布对象绘制图形

例如 context.fillRect(10,10,100,100);

3、绘制图形

(1)绘制矩形

fillRect(x,y,width,height) -- 绘制实心矩形

x、y，绘制矩形的左上角的坐标值(x,y)

width、height，绘制矩形的宽和高

strokeRect(x,y,width,height) -- 绘制空心矩形

clearRect(x,y,width,height) -- 清除指定区域的矩形

(2)设置样式

fillStyle，设置填充样式

strokeStyle，设置描边样式

globalAlpha，设置透明度

注意：

必须先设置样式，再绘制图形

每改变一次样式，都需要重新设置样式

填充样式与描边样式互不干扰

例如 context.fillStyle="yellow";

context.fillRect(10,10,100,100);

context.fillStyle="green";

context.fillRect(10,120,100,100);

(3)渐变

①线性渐变

createLinearGradient(x1,y1,x2,y2)

线性渐变有一个基准线，x1、y1为起点线的坐标；x2、y2为终点线的坐标

②射线(扇形)渐变

createRadialGradient(x1,y1,r1,x2,y2,r2)

射线渐变具有两个基准圆

如何实现：

设置线性渐变或射线渐变

返回渐变对象

渐变对象设置渐变的颜色

addColorStop(position,color);

position，设置当前颜色的位置

取值范围为0~1(类似于透明的取值)

该方法由渐变对象调用

设置填充或描边样式为渐变

绘制图形

4、绘制文字

(1)绘制方法

fillText(text,x,y) -- 绘制实心文字

text，设置绘制文字的内容

x、y，设置绘制文字的坐标值

strokeText(text,x,y) -- 绘制空心文字

(2)设置属性

font，用法与CSS中的用法一致

(3)对齐方式

textAlign，水平对齐

left，基准线在文字的左边

center，基准线在文字的中间

right，基准线在文字的右边

textBaseline，垂直对齐

top，基准线在文字的上边

middle，基准线在文字的中间

bottom，基准线在文字的下边

hanging，悬挂基线

alphabetic，字母基线

5、设置阴影

shadowColor，设置阴影颜色

shadowOffsetX，设置阴影水平位置

正值，阴影向右移动

0，不移动

负值，阴影向左移动

shadowOffsetY，设置阴影垂直位置

正值，阴影向下移动

0，不移动

负值，阴影向上移动

shadowBlur，设置阴影的程度

值的类型为number，值越大，阴影效果越明显

例如 context.shadowColor="red";

context.shadowOffsetX=-30;

context.shadowOffsetY=30;

context.shadowBlur=50;

6、创建路径绘制图形

(1)绘制矩形或圆形

开始标识方法 -- beginPath()

设置图形

rect(x,y,width,height)，矩形

arc(x,y,radius,startAngle,endAngle,direction)，圆形或弧形

x、y，圆形的圆心坐标值

radius，半径

startAngle，开始位置

范围在0-2PI之间

endAngle，结束位置

direction，顺时针或逆时针

Boolean值，默认值为false，顺时针

例如 context.arc(280,60,50,0,Math.PI\*3/2,true);

结束标识方法 -- closePath()

此方法可有可无

\*绘制空心弧形时：

如果增加此方法，弧形为闭口

如果无此方法，弧形为开口

绘制方法

fill()

stroke()

(2)绘制线型

绘制直线

开始标识方法 -- beginPath()

设置直线的起点坐标值

moveTo(x,y)

设置直线的终点坐标值

lineTo(x,y)

结束标识方法 -- closePath()

绘制方法 -- stroke()

绘制折线

开始标识方法 -- beginPath()

设置直线的起点坐标值

moveTo(x,y)

设置直线的终点(起点)坐标值

lineTo(x,y)

设置直线的终点坐标值

lineTo(x,y)

结束标识方法 -- closePath()

该方法不写，则是折线效果

该方法写，则是闭口，多边形效果(空心)

绘制方法 -- stroke()

7、设置线条

lineWidth，设置线的宽度

默认值为1px

lineCap，设置线的端点形状

butt，默认值，表示平角

round，圆角

square，正方

lineJoin，设置线的连接点形状

miter，默认值，尖角

miterLimit，设置尖角的程度

round，圆角

bevel，斜角

8、绘制图片

drawImage(img,x,y)

img，绘制的图片内容

x、y，绘制图片的坐标值

drawImage(img,x,y,width,height)

width、height，设置绘制图片的宽和高

设置宽和高的比例时，尽量按照原图的高宽比设置，否则图片会被拉伸或者缩小

一些浏览器，可能会出现图片加载和绘制速度不正确，导致绘制失败

解决：将加载的图片绑定onload事件，在onload事件的处理函数中进行绘制

9、平铺图片

createPattern(img,type)

img，平铺的图片内容

type，平铺方式

no-repeat，不平铺

repeat，平铺

repeat-x，水平平铺

repeat-y，垂直平铺

实现步骤：

设置平铺方式

返回一个平铺对象

设置填充或描边样式为平铺

绘制图形

例如 var ptn=context.createPattern(img,"repeat");

context.fillStyle=ptn;

context.fillRect(0,0,canvas.width,canvas.height);

10、切割图片

使用创建路径方式，步骤：

beginPath();

rect()或arc();

closePath();

clip();

11、画布方法

(1)状态方法

save()，保存方法

将画布设置的内容进行保存

只能保存一次；如果保存多次，前面的将会被覆盖

restore()，读取(恢复)方法

读取之前保存的画布设置内容

(2)转换方法

scale(x,y)，表示缩放(放大或缩小)

值大于1，表示放大

值小于1，表示缩小

translate(x,y)，表示平移

该方法是相对平移 -- 例如(100,150)

相对于上次的位置，向x方向平移100，向y方向平移150，与上次坐标值有关

rotate(degrees)，表示旋转画布

角度公式：degrees\*Math.PI/180

旋转中心点为画布左上角

画布旋转方向与图形旋转方向【相反】

\*中心点旋转固定套路：

context.translate(canvas.width/2,canvas.height/2);

七、Chart.js库

专门用于绘制各种图标(使用HTML5的canvas)

官网 http://www.chartjs.org/

提供6种图表：

饼状图

data结构

[

{

value:50, //饼状图的值

color:"red", //颜色

highlight:"deeppink", //鼠标悬停的颜色

label:"Red" //文字提示内容

}, ...

]

相关方法 -- Pie(data,options)

柱状图

data结构

{

labels:[柱状图的x轴的提示内容],

datasets:[

{

label:柱状图的提示内容,

fillColor:填充颜色,

strokeColor:描边颜色,

highlightFill:鼠标悬停填充颜色,

highlightStroke:鼠标悬停描边颜色,

data:柱状图的数据

}, ...

]

}

相关方法 -- Bar(data,options)

曲线图

data结构

相关方法 -- Line(data,options)

环形图

data结构

相关方法 -- Doughnut(data,options)

雷达图

data结构

相关方法 -- Radar(data,options)

极地区域图

data结构

相关方法 -- PolarArea(data,options)

options选项

通过图表绘制的方法的选项

作用范围：当前Chart图表起作用

对Chart对象的全局选项

作用范围：所有Chart图表都起作用

Chart.defaults.global.选项名称=值;

使用Chart.js库的步骤

在HTML代码中

<head>中引入Chart.js库文件

定义<canvas>元素

在Js代码中

创建画布对象

通过画布对象，创建Chart对象

设置图表的相关数据

调用Chart.js库提供的API方法

八、SVG

1、基本内容

全称为Scalable Vector Graphics

是一种使用XML技术描述二维图形的语言

SVG与HTML5的关系

早在HTML5之前，就存在SVG技术

SVG文件的扩展名为".svg"

在HTML5出现之前，要在HTML页面中引入SVG文件；在HTML5出现之后，将SVG内容直接定义在HTML页面中

SVG的标准也是W3C定制的

SVG的优势

可以使用文本编辑器创建和修改

可被搜索、索引等

绘制的图像(被放大时)不失真

SVG与Canvas的区别

SVG

SVG不依赖于分辨率

可以使用DOM或绑定事件

不能实现网页游戏

实现大型渲染区域的应用，比如百度地图

可以使用style属性里面的width、height设置宽高，图片不会被拉伸

Canvas

Canvas依赖于分辨率

不能使用DOM或绑定事件

运行时以图片的形式出现，比如.jpg

做网页游戏

2、设置样式

通过元素的属性方式

fill，设置填充样式

fill-opacity，设置填充透明度(范围0~1)

stroke，设置描边样式

stroke-width，设置边框的宽度

transform属性，用于设置转换效果

方法： rotate()

scale()

translate()

通过style属性，使用类似于CSS属性的设置方式

3、绘制图形

(1)矩形元素 -- <rect/>

<rect x="" y="" width="" height="" rx="" ry=""/>

x、y，绘制矩形的左上角的坐标值

width、height，设置绘制矩形的宽和高

rx、ry，设置圆角

(2)圆形元素 -- <circle/>

<circle cx="" cy="" r=""/>

cx、cy，绘制圆形的圆心坐标值

如果cx、cy不设置的话，默认为(0,0)

r，圆形的半径

(3)椭圆元素 -- <ellipse/>

<ellipse cx="" cy="" rx="" ry=""/>

rx，椭圆水平方向的半径

ry，椭圆垂直方向的半径

(4)线条元素 -- <line/>

<line x1="" y1="" x2="" y2=""/>

默认描边颜色为白色，需要stroke="black"，但是fill="black"不管用

(5)折线元素 -- <polyline/>

<polyline points=""/>

points，折线的所有点的坐标值，都设置在该属性中

描边颜色默认为白色，填充颜色默认为黑色

(6)多边形元素 -- <polygon/>

<polygon points=""/>

4、特效元素

(1)渐变元素

①线性渐变

先定义<defs>元素，渐变元素定义在该元素内

定义<defs>元素的内容，允许重复使用

定义<defs>元素内的内容，不会直接显示在页面中

<defs>元素是起始标签

再定义<linearGradient>线性渐变元素

<linearGradient>元素是起始标签

线性渐变的基准线：起点(x1,y1)和终点(x2,y2)

值必须是百分比

定义id属性值，绘制图形时，赋值给fill属性

设置渐变的颜色，使用<stop>元素

offset，设置渐变颜色的位置

值必须是百分比

stop-color，设置渐变的颜色

绘制图形时，使用fill属性设置渐变样式

fill="url(#渐变元素的id)"

②射线渐变

<radialGradient>

fx、fy，起点基准圆的圆心(没有半径值)

cx、cy，终点基准圆的圆心

r，终点基准圆的半径

\*基准线(圆)的坐标值必须是百分值，可为负，可大于100%

\*设置渐变颜色位置必须是百分值，offset范围只能为0%~100%

(2)滤镜元素 -- 高斯模糊

滤镜元素 -- <filter>

定义filter元素时，需要定义id属性值，绘制图形时，将其设置为图形的样式

绘制图形时，使用filter属性设置滤镜效果，filter="url(#滤镜元素的id)"

高斯模糊 -- <feGaussianBlur>

in="SourceGraphic"

stdDeviation，设置高斯模糊的程度，Number类型

值越大，模糊程度越大

例如 <feGaussianBlur in="SourceGraphic" stdDeviation="5"/>

九、Tow.js库

1、基本内容

概念：在HTML页面中进行二维图形的绘制

支持方式：

Canvas，以图片形式出现，绘制二维图形

SVG，绘制二维图形(Two.js默认支持方式)

WebGL，绘制三维图形

官网：http://jonobr1.github.io/two.js/

特点：

简单地实现矢量图及动画效果

不支持文字和图片

专门提供SVG解释器

2、使用Two.js的步骤

HTML页面中

引入Two.js库文件

定义容器元素<div>

JavaScript代码中

获取页面的容器元素<div>

创建Two.js库的对象

将Two对象添加到页面容器中

使用Two.js库提供的API完成需求

3、Two对象

构造器 -- new Two(params)

选项：

type，创建哪种类型，默认为SVG

默认格式为Two.Types.svg

Canvas格式为Two.Types.canvas

4、绘制静态图形

绘制直线 -- makeLine(x1,y1,x2,y2)

该方法返回Two.Line对象

绘制矩形 -- makeRectangle(x1,y1,width,height)

该方法返回Two.Rectangle对象

绘制圆形 -- makeCircle(x,y,r)

该方法返回Two.Circle对象

绘制椭圆 -- makeEllipse(x,y,width,height)

该方法返回Two.Ellipse对象

绘制多边形 -- makePolygon(ox,oy,r,sides)

该方法返回Two.Polygon对象

5、制作动画效果

play()，执行循环动画

update()

6、Two.js的高级用法

分组 -- Two.Group

例如 var group=two.makeGroup(circle,rect);

十、拖放(拖拽)API

1、基本内容

HTML默认是不允许实现拖放效果的

源元素 -- 鼠标拖放【的】哪个元素

源元素事件

dragstart，当开始拖放时被触发

drag，拖放过程中不断被触发

dragend，当结束拖放时被触发

目标元素 -- 鼠标拖放【到】哪个元素

目标元素事件

dragenter，只是鼠标到达目标元素时被触发(一次)

dragover，当鼠标到达目标元素时触发(多次)

\*必须调用事件对象event的preventDefault()方法

表示阻止页面的默认行为

drop，当鼠标进行投放时被触发

默认情况下，不被触发

\*只有在dragover事件的处理函数中，调用事件对象event的preventDefault()方法时，才会被触发

dragleave，当鼠标离开目标元素时触发

2、dataTransfer对象

作用：提供类似于剪切板的功能

方法：

setData(type,data)

type，类似于标识

data，操作的数据内容

getData(type)

clearData(type)

该对象被增加在event事件对象中

十一、Web Workers

1、基本内容

概念：是一个运行在后台的JavaScript，独立于HTML页面运行的一段JavaScript代码

特点：

同时运行几个逻辑，并不需要顺序执行

独立于HTML页面的循序执行

Worker之间相对是独立的

缺点：

不能操作页面，是单纯的JS逻辑

导致不能使用BOM

不能使用DOM

\*主流浏览器不允许读取本地文件

例如 file:///H:/笔记/Text/Html5%20core/Unit%206/worker1.js

Chrome，直接报错

Firefox，不报错，但不能用

IE，不报错，但不能用

2、Worker API

检测浏览器对Worker支持情况

创建Worker文件

创建Worker对象

与Worker进行通信

onmessage事件

postMessage(data)方法

作用：用于向HTML页面(onmessage事件的处理函数)传递数据内容

使用方法：

在HTML页面中 - 通过Worker对象进行调用

在Worker文件中 - 允许直接调用该方法

data参数：是postMessage()方法传递的数据

终止Worker

worker.terminate()方法

十二、Web存储API

1、基本内容

数据库产品 - 持久性存储数据

浏览器内存 - JavaScript的变量与函数

Cookie - 存储在用户的硬盘(跨浏览器)

Cookie的文件最大限制4KB

每个Web应用最多只能使用200多个Cookie文件

Cookie是以明文存储数据(未加密)

HTML5的数据库

Web SQL

IndexedDB，轻量级数据库产品

HTML5的存储

localStorage(本地存储) - 替代Cookie

数据由用户决定是否删除

sessionStorage(会话存储) - 替代Session

当页面关闭时，数据会被清空

一次会话

客户端与服务器端建立连接，直到关闭连接的过程

2、sessionStorage

属性

length，返回当前Storage存储的数据个数

方法

setItem(key,value)，保存数据到Storage

key，保存数据的标识(唯一，不可重复)

如何生成key - 时间戳

new Date().getTime();

如果key存在，就是修改指定数据内容

如果key不存在，就是新增指定数据内容

value，保存的数据

getItem(key)，获取从Storage存储的数据

要想获取数据，必须知道key值

removeItem(key)，根据key删除指定数据

key(index)，根据索引值返回key值

index，索引值

clear()，清空

通过window对象的sessionStorage属性，返回的是(session)Storage对象

使用window对象的属性和方法时，可省略"window."，sessionStorage对象可直接得到

3、localStorage

属性、方法与sessionStorage一致

storage事件

作用：实现多页面之间共享数据

\*该事件目前只有Safari浏览器支持

十三、Web Socket

1、基本内容

Socket，就是一种网络协议，类似于HTTP

Socket与HTTP的区别

Socket

实时连接、有状态

HTTP

短连接、无状态

HTML5提供的Web Socket底层封装Socket协议

2、Web Socket API

创建WebSocket对象

var socket=new WebSocket(url);

url，服务器端的地址

协议：不是"http://"协议，而是"ws://"或"wss://"

作用：与服务器端建立连接

方法

send(data)方法，向服务器端发送请求数据

data，请求数据

close()方法，关闭与服务器端的连接

事件

onmessage，用于监听服务器端的状态

onopen，用于监听打开连接

触发：打开连接时被触发

作用：该事件的处理函数完成必要的初始化

onclose，用于监听关闭连接

触发：关闭连接时被触发

作用：该事件的处理函数完成资源释放