

HTML 5 day01

3.1: html5 十个新特性

- 新的语义标签[header;footer...]
- 增型表单
- 视频与音频
- 绘图 canvas(统计图;分析图;特效;小游戏)
- 绘图 svg
- 地理定位
- 拖放 API
- Web Worker
- Web Storage
- Web Socket

3.2: html5 新特性-增型表单

(1)新的 input type <input type=?>

h4:text;password;radio;checkbox;button;submit

h5:number;email;url;color;date;month;week

(2)新的元素 element

①-datalist 建议列表

配合 input 创建建议列表, 当用户不清楚如果输入内容

提供建议

<datalist id="list3"> 默认情况 datalist 不可见

<option>xx</option>

<option>yy</option>

</datalist>

<input type="text" list="list3"/>

②-progress 进度条

显示一个进度条两种形式

<progress></progress> 左右晃动进度条

<progress value="0.7" /> 具有指定进度值

进度条

练习:使用定时器+进度条实现一个可以动态

前进的前度条到 100%停止

定时器 1s 修改 value+0.1 到 1 结束

③-meter 刻度尺

<meter min="最小值" max="最大值" low="下限" high="上限"

optimum="最佳值" value="当前值" />

当前值离最佳值非常远 (红)危险

当前值离最佳值比较近 (黄)警告

当前值离最佳值非常近 (绿)正常

④-output 输出:语义标签,没有任何外观样式

外观同 span

商品单价:¥3.50

购买数量:<input type="number" value="2" />

小计:<output>¥3.50</output>

(3)新的属性 attr <input ? />

h4:id;class;name;value;style;readonly;checked

h5:

-autofocus: 自动获取输入焦点

-placeholder: 占位符

-form: 用于把输入域放置在 form 外部

<form id="f5">

</form>

<input type="text" form="f5"/>

-multiple: 允许输入多值(用逗号分隔) email

a@a.com,b@b.com

-验证相关

required:必填项, 内容不能为空

minlength;maxlength: 最小字符串长度

min;max

数值最大值与最小值

pattern

输入正则表达式

3.3: html5 新特性-多媒体视频

Flash

Flash 绘图(AS/Flex) -> Canvas+SVG

<https://echarts.baidu.com/>

echarts 第三方绘图库

<https://d3js.org.cn/>

d3 绘图库

Flash 动画(游动) -> Canvas+第三方游戏平台

<https://www.cocos.com>

微信小游戏

Flash 视频(音频) -> video/audio (90%)

Flash 存储 -> h5 WebStorage

3.4: html5 新特性-多媒体视频使用

<video src="x.mp4"></video>

知识补充:

(1) 现在常用视频格式哪些

mp4 .flv .webm .ogg

(2)如果浏览器需要播放指定格式视频,需要安装一个解码器

软件

x.mp4 -> mp4 解码器

x.flv -> flv 解码器

(3)如果低版本浏览器安装解码器

<video src="x.mp4" />

问题:我们希望大多数浏览器都能播放视频

-x.mp4 使用工具软件 "格式工厂" x.flv

x.webm x.ogg

<video>

<source src="x.mp4" />

<source src="x.flv" />

<source src="x.webm" />

<source src="x.ogg" />

您的浏览器版本太低, 请升级!!!

</video>

3.5: html5 新特性-多媒体视频-常见属性

controls:false	是否显示播放控件	<code><video controls></code>
autoplay:false	是否自动播放	<code><video autoplay></code>
#autoplay	兼容性非常差	
loop:false	是否循环播放	
muted:false	是否静音播放	
poster:""	在播放第一帧画面之前显示海报	
preload	视频预加载策略	
-auto	预加载一定时长视频和元数据	
-metadata	只预加载元数据(时长;尺寸;第一帧画面)	
-none	不预加载任何数据	

3.6: html5 新特性-多媒体视频-js 属性

volume:1	当前音量
playbackRate:1	回放速度大于 1 快放小于 1 慢播
paused:false	当前视频是否处于暂停状态
play()	播放视频
pause()	暂停播放
onplay	当视频开始播放时触发事件
onpause	当视频暂停播放时触发事件



练习 1:不使用 video 自带 controls 自定义 播放/暂停按钮

鼠标移入视频区域显示按钮, 鼠标移出视频区域

隐藏按钮

练习 2: 为按钮绑定点击事件 播放/暂停 切换图片

```

<style>
body{
  background-color:#ffff;
  text-align:center;
}
.container{
  position:relative;
}
.ctrl{
  position:absolute;
  top:50%;
  left:50%;
  margin-top:-60px;
  margin-left:-60px;
}
.ctrl img{
  width:120px;
  height:120px;
}
.ad{
  position:absolute;
  width:400px;
  height:300px;
  left:50%;
  top:50%;
  margin-left:-200px;
  margin-top:-150px;
}
</style>
<div>
<!-- 父元素 -->
<div class="container">
  <!-- 视频元素 -->
  <video src="src/res/x.mp4" id="v3"></video>
  <!-- 广告 -->
  
  <!-- a 标签超链接 -->
  <a href="javascript:;" class="ctrl">
    <!-- 图片 -->
    
  </a>
</div>
</div>
<script>
var container=document.querySelector(".container");
var v3=document.getElementById("v3");
var ctrl=document.querySelector(".ctrl");
var img=document.querySelector(".ctrl img");

//2. 为父元素绑定鼠标 进入离开事件
container.onmouseenter = function(){
  ctrl.style.display = "block";
};
container.onmouseleave = function(){
  ctrl.style.display = "none";
};

//3. 为ctrl绑定点击事件
ctrl.onclick=function(e){
  //3.1阻止默认行为 a submit image button
  e.preventDefault();
  //3.2判断当前视频暂停状态true
  if(v3.paused){
    //3.3 播放 切换图片
    v3.play();
    img.src = "src/res/pause.png";
  }else{
    //3.4 暂停 切换图片
    v3.pause();
    img.src = "src/res/play.png";
  }
};

//4. 广告显示和隐藏
var ad=document.querySelector(".ad");
v3.onplay=function(){
  ad.style.display = "none";
};
v3.onpause=function(){
  ad.style.display = "block";
};
</script>

```

练习 3: 创建表单登录表单 背景循环播视频

提示:最外层父元素 相对定位

表单

绝对定位

背景视频

绝对定位 z-index:-1 loop

muted

作业 1:

购物车商品后删除按钮

(1)为删除按钮绑定点击事件

(2)要从购物车表中指定商品删除

(3)脚手架将对应商品删除(用户看不到)

HTML 5 day02

3:HTML5 新特性—视屏高级特性/音频/绘图(重点)

3.1: HTML5 新特性—视屏高级特性-样式

video 元素专用样式 object-fit
fill:“填充” 默认值 将视频拉伸操作填满 content box
contain:“包含” 保持原有尺寸比例, 容器空白
cover:“覆盖” 保持原有尺寸比例, 宽度和高度至少一个和容器一致
注意: **cover;contain** 使用比较多一些
 如果创建视频, 与录制视频工程师 750*1334
 iphone6 屏幕(全屏播放)

3.2: HTML5 新特性—视屏高级特性-特殊属性

#扩展知识: 苹果 IOS/谷歌 Android (同层播放)
 ios <video webkit-playsinline="true">
 Android <video playsinline="true">
 结果
 <videowebkit-playsinline="true"playsinline="true">
 (横屏错位)
 Android
 <video x5-video-player-fullscreen="true"></video>

3.3: HTML5 新特性—视屏高级特性-特殊事件

-timeupdate 当视频在播放时候时间发生变化不断触发事件
#currentTime 当视频播放时间
-ended 当视频播放结束时候
-canplaythrough 当视频加载结束可以播放时触发事件
#duration 视频时长

3.4:html5 新特性--音频

<audio src="x.mp3"></audio>
 #建议:为了兼容性问题使用格式工厂将音频文件转换多种格式
 <audio>
 <source src="x.mp3" />
 <source src="x.wav" />
 <source src="x.ogg" />
 </audio>
 属性方法事件大多数与视频相同
 练习:创建复选框控制背景音乐播放
 当复选框选中 播放背景音乐 play()
 当清空复选框 背景音乐停止 pause()
 cb.change cb.checked==true

音频

默认情况下音频文件不可见 -->

<audio src="../../day01/src/res/bg.mp3" controls></audio>

播放背景音乐<input type="checkbox">

音频的控件, 一般是隐藏的, 如果加上就显示音频控件

//练习: 创建一个复选框控制背景音乐播放 (获取背景音乐)

var au=document.querySelector("body>audio");

var ck=document.querySelector("body>input");

console.log(ck)

//当复选框中 播放背景音乐 play()

ck.onchange=function(){

 if(this.checked){

 au.play();

 }else{

 //当清空复选框 背景音乐停止 pause()

 au.pause();

 }

</script>

src/04_audio.html

0.15 / 0.59

播放背景音乐 ☒

复选框选中状态就播放音频

3.5:html5 新特性--绘图--(重点)

网页中绘制图形;股票走势图;分析图;网页特效;网页游戏...
 都需要使用绘图技术完成
 网页中绘制图形三种技术
 (1)svg 绘图技术:2d 矢量图绘制技术
 2000 出现纳入 h5 标准
 (2)canvas 绘图技术:2d 位图绘制技术
 (3)webgl 绘图技术:3d 位图绘制技术, 尚未纳入 h5 标准

3.6:html5 新特性--canvas 绘图难点二个[坐标系;单词多]



3.7:html5 新特性--canvas 绘图矩形

(1)#创建画布

```
<canvas id="c3" width="500" height="400"></canvas>
```

注意事项:画布宽度和高度只能使用属性或js赋值,不能用 css

样式来赋值(css 缩放)

(2)#通过 js 程序获取画布

```
var c3 = document.getElementById("c3");
```

(3)#通过 js 程序获取画笔[上下文对象]

```
var ctx = c3.getContext("2d");
```

#注意事项:一个画笔对象对应一个画布对象

-ctx.lineWidth = 1; 描边宽度(空心矩形边框宽度)

-ctx.strokeStyle = "#fff"; 描边样式

-ctx.fillStyle = "#fff"; 填充样式(实心矩形内容样式)

-ctx.strokeRect(x,y,w,h); 描边一个矩形

-ctx.fillRect(x,y,w,h); 填充一个矩形

#注意事项:x y 矩形左上角位置 w h 宽度和高度

#注意事项:矩形定位点在自己左上角

-ctx.clearRect(x,y,w,h); 清除一个矩形范围内所有元素

①练习:在画布左上角右下角左下角右下角居中位置绘制

五个填充矩形颜色不同



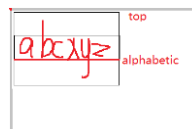
```
<style>
  body{text-align:center;}
  canvas{background-color:#ddd;}
</style>
</head>
<body>
  <h3>canvas 绘图工具</h3>
  <canvas id="c3" width="500px" height="400px"></canvas>
  <script>
    //获取画布
    var c3=document.getElementById("c3");
    //获取画笔
    var ctx=c3.getContext("2d");
    ctx.fillStyle = "#f00";
    ctx.fillRect(0,0,100,80);
    ctx.fillStyle = "#a00";
    ctx.fillRect(400,0,100,80);
    ctx.fillStyle = "#faa";
    ctx.fillRect(0,320,100,80);
    ctx.fillStyle = "#ffa";
    ctx.fillRect(400,320,100,80);
    ctx.fillStyle = "#faf";
    ctx.fillRect(200,160,100,80);
  </script>
```

②练习:在画布上绘制可以左右移动 100*80 矩形
提示:使用定时器先清除画布已有内容,再重新绘制矩形

(x 不停增加)

```
<canvas id="c3" width="500" height="400"></canvas>
<script>
  //1.获取画布
  var c3= document.getElementById("c3");
  //2.获取画笔
  var ctx=c3.getContext("2d");
  //3.创建变量保存矩形x值
  var x=0;
  //3.1:创建一个变量代表方向 1向右 -1向左、
  var xDirection = 1;
  //4: 创建定时器 1s
  var t=setInterval(function(){
    //4.1:清除画布
    ctx.clearRect(0,0,500,400);
    //4.2:修改x值
    x+=1 * xDirection;
    //4.3: 绘制矩形
    ctx.fillStyle = "#faf";
    ctx.fillRect(x,0,100,80);
    if(x>400){ //矩形移动到最右边 向左
      xDirection = -1;
    }
    if(x<0){
      xDirection = 1;
    }
  },15);
</script>
```

3.8:html5 新特性--canvas 绘图文本



ctx.textBaseline = "top"; 设置文本基线
ctx.font = "19px SimHei"; 设置文本大小和字体

var str = "abcxyz"; 创建文本对象
ctx.fillText(str,x,y); 绘制填充文本
ctx.strokeText(str,x,y); 绘制空心文本
ctx.measureText(str); 测量文本宽度

{width:x}

#注意:先绘制文本对象再测量文本宽度

练习一:画布四个角绘制文本

练习二:依据服务器端程序数据绘制销售统计图

(1)创建服务器端程序

(2)发送 ajax 请示获取销售数据并且

绘制统计图

src/08_app.js 服务器端程序

public/index.html 客户端程序

HTML 5 day03

今天学习的内容

1:复习昨天重点

2:作业

3:今天的目标

3.1:销售统计图

(1)node.js

(2)index.html

3.2:html5 新特性 -- canvas 绘图-路径

path: 由多个坐标点组成任意图形，图形不可见

可以使用"描边" "填充"

ctx.beginPath() 开始一条新路径

ctx.closePath() 闭合当前路径

ctx.moveTo(x,y) 移动到指定点

ctx.lineTo(x,y) 从当前点到指定点画

一条直线

ctx.arc(cx,cy,r,start,end); 绘制圆拱形

cx,cy 圆心

r 半径

start 起始角度

end 结束角度

#注意事项:程序不使用角度->弧度

#0 180 360 角度

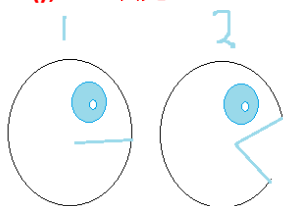
#0 PI 2*PI 弧度

#通用解决方案:公式 角度转弧度

#角度*Math.PI/180 = 弧度

ctx.stroke(); 描边

ctx.fill(); 填充



练习:

创建二个函数:openMouth() 张嘴笑脸

closeMouth() 闭嘴笑脸

每隔一秒钟切换另一个笑脸

```
<h3>两个笑脸</h3>
<canvas id="c3" width="500" height="400"></canvas>
<script>
var c3 = document.getElementById("c3");
var ctx = c3.getContext("2d");
console.log(ctx);
function closeMouth(){
  //1:绘制外围轮廓
  ctx.beginPath();
  ctx.arc(250,200,100,0,2*Math.PI);
  ctx.lineTo(250,200);
  ctx.stroke();
  //2: 绘制眼球蓝色
  ctx.beginPath();
  ctx.arc(265,155,25,0,2*Math.PI);
  ctx.fillStyle = "blue";
  ctx.fill();
  //3: 绘制眼神白色
  ctx.beginPath();
  ctx.arc(270,152,5,0,2*Math.PI);
  ctx.fillStyle = "#fff";
  ctx.fill();
}
function openMouth(){
  //1:外围的轮廓
  ctx.beginPath();
  ctx.arc(250,200,100,30*Math.PI/180,345*Math.PI/180);
  ctx.lineTo(250,200);
  ctx.closePath();
  ctx.stroke();
  //2: 蓝色眼神
  ctx.beginPath();
  ctx.arc(265,150,25,0,2*Math.PI);
  ctx.fillStyle = "blue";
  ctx.fill();
  //3: 白色眼球
  ctx.beginPath();
  ctx.arc(270,142,5,0,2*Math.PI);
  ctx.fillStyle = "#fff";
  ctx.fill();
}
var index = 0;
setInterval(function(){
  ctx.clearRect(0,0,500,400);
  index++;
  if(index % 2 ==0){ 清除画布
    openMouth();
  }else{
    closeMouth();
  }
},500);
</script>
```

3.3:html5 新特性 -- canvas 绘图-图像

canvas 属于客户端技术(运在浏览器)但是图片保存服务器中

，所以浏览器必须先下载要绘制图片，且等待下载完成再

绘制。

#图片为什么保存服务器

(1)图片有版权软件项目宝贵资源

(2)图片通常数量巨大

#操作流程

①-创建图片对象 var p3 = new Image();

②-下载图片 p3.src = "x.png";

③-绑定事件[下载完成] p3.onload = function(){

④-绘制图片 ctx.drawImage(p3,x,y,w,h);

拉伸大小

ctx.drawImage(p3,x,y); 原始大小

练习:在画布左上角 右上角 左下角 右下角
绘制四架小飞机

练习:在画布上绘制一个可以随鼠标移动而移动小飞机

```
<h3>小飞机移动</h3>
<canvas id="c3" width="500" height="400"></canvas>
<script>
    var c3 = document.getElementById("c3");
    var ctx = c3.getContext("2d");
    //0: 创建图片对象下载图片绑定事件下载完成
    var p3 = new Image();
    p3.src = "res/p3.png";
    p3.onload = function(){
        var w = p3.width;
        var h = p3.height;
        //1: 为画布绑定鼠标移动事件 mousemove
        c3.onmousemove = function(e){
            //2: 获得x y
            var x = e.offsetX;
            var y = e.offsetY;
            //3: 依据x y 绘制图片
            ctx.clearRect(0,0,500,400);
            ctx.drawImage(p3,x-w/2,y-h/2);
            //4:
        }
    }
</script>
```

获取小飞机自身的宽度和高度

获取鼠标的坐标

使得鼠标在飞机的中心位置移动

把鼠标的坐标值赋给绘制小飞机的位置坐标, 当鼠标移动的位置绘制小飞机

3.4:html5 新特性 -- canvas 绘图-变形(旋转)

canvas 绘图技术可以针对一个图像在绘制过程中进行

变形(旋转)操作

ctx.rotate(弧度); 旋转画笔对象, 旋转轴心画布原点

ctx.translate(x,y); 将整个画布原点平移到某处

练习:以 250, 200 为轴心不停旋转小飞机每隔 1 秒旋转 10 度

练习:绘制二架小飞机分别是小飞机 1(10)和小飞机 2(20)

小飞机 2 旋转速度小飞机 1 2 倍

注意事件:

通用规则:当一个画布上绘制一个以上的元素画之前先"保存"画之后"恢复"这种操作使不同元素之间不受影响.

ctx.save(); 保存画笔状态

ctx.restore(); 恢复画笔状态

```
<h3>两件小飞机</h3>
<canvas id="c3" width="600" height="600"></canvas>
<script>
    var c3 = document.getElementById("c3");
    var ctx = c3.getContext("2d");
    //1: 创建两个变量保存不同飞机的角度
    var deg1 = 10;
    var deg2 = 20;
    //2: 创建图像对象
    var p3 = new Image();
    //3: 下载图像
    p3.src = "res/p3.png";
    //4: 绑定事件onload
    p3.onload = function(){
        var w = p3.width;
        var h = p3.height;
        //5: 创建定时器
        setInterval(function(){
            //5.1: 保存画笔状态-----小飞机1 start
            ctx.save();
            //5.2: 平移圆点左上角
            ctx.translate(100,100);
            //5.3: 旋转10度
            ctx.rotate(deg1*Math.PI/180);
            //5.4: 绘制图像
            ctx.drawImage(p3,0-w/2,0-h/2);
            //5.5: 恢复画笔状态
            ctx.restore();
            //5.6: 累加小飞机1角度-----小飞机1 end
            deg1 +=10;
            //
            // 第二架飞机
            ctx.translate(300,300);
            //5.1: 保存画笔状态-----小飞机2 start
            ctx.save();
            //5.2: 平移圆点左上角
            ctx.translate(300,300);
            //5.3: 旋转10度
            ctx.rotate(deg2*Math.PI/180);
            //5.4: 绘制图像
            ctx.drawImage(p3,0-w/2,0-h/2);
            //5.5: 恢复画笔状态
            ctx.restore();
            //5.6: 累加小飞机1角度-----小飞机2 end
            deg2 +=10;
            //
        },100);
    }
</script>
```

初始化两架飞机旋转的弧度

获取飞机的本身的高

由于 平移 移动的是画笔的原点, 当第二架飞机在绘制过程中, 画笔的原点变化, 只能在第二架飞机的原点位置绘制, 无法回到绘制第一架飞机的原点, 每一张画布对应一个画笔, 所以, 每个要保持画笔的状态, 每次绘画完毕之后, 恢复画笔的原点, 就可以继续在当前位置绘图了

HTML 5 day04

3.1:html5 新特性--canvas-渐变对象(过渡效果)

-创建渐变对象

var g = ctx.createLinearGradient(x1,y1,x2,y2);

x1 y1 起点坐标

x2 y2 终点坐标

-添加颜色点

g.addColorStop(offset,color);

示例:

g.addColorStop(0.3,"green");

-将渐变对象赋值样式(描边;填充)

ctx.fillStyle = g;

ctx.strokeStyle = g;

-绘制图形

3.2:html5 新特性--svg

	canvas	svg
类型	2d 位图	2d 矢量
如何绘图	使用 js 代码绘图	使用标签绘图
事件绑定	只能将事件绑定画布	每个图形都可以绑定事件
应用场景	特效,小游戏	地图

W3C html <html> <HTML>

.. 灵活
XML 淘汰 java 配置文本 传递
html5 标准 (xml svg 绘图)

3.2:html5 新特性--svg-开发流程

-创建画布

```
<svg id="s3" width="500" height="400">
```

创建图形标签

```
</svg>
```

-在画布中添加标签(图形)-矩形

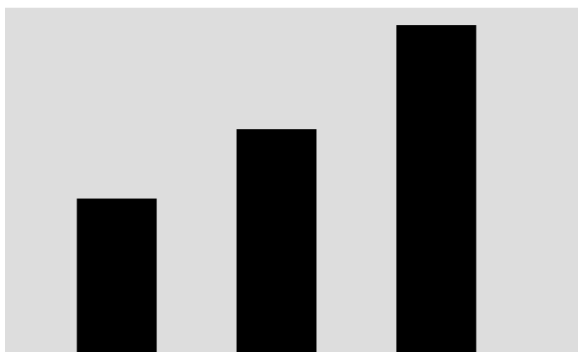
```
<rect x="" y="" width="" height="" fill="" fill-opacity=""
```

```
stroke="" stroke-opacity=""></rect>
```

练习:左上角 右上角 左下角 右下角 居位置绘制

五个矩形, 颜色不同从服务器请求数据

销售统计图



```
const express = require("express");
var app = express();
app.use(express.static("public"));
app.listen(3000);
app.get("/sales", (req, res) => {
  var rows = [
    {id: 1, name: "东口", value: 180},
    {id: 2, name: "南口", value: 260},
    {id: 3, name: "地铁口", value: 380}
  ];
  res.send(rows)
});
```

```
<h3>销售统计图</h3>
<svg id="s3" width="500" height="400"></svg>
<script>
//1: 创建xhr对象
var xhr = new XMLHttpRequest();
//2: 绑定事件 onreadystatechange
xhr.onreadystatechange = function(){
//2.1: 判断 readState==4&&status==200
if(xhr.readyState==4 && xhr.status==200){
//2.2: 接收 xhr.responseText
var result = xhr.responseText;
//2.3: 将字符串转化为js对象
var obj = JSON.parse(result);
console.log(obj);
//2.4: 获取画布
var s3 = document.getElementById("s3");
//2.5: 创建循环
for(var i=0; i<obj.length; i++){
//2.6: 创建rect x y w h
var item = obj[i];
var rect = document.createElementNS("http://www.w3.org/2000/svg", "rect");
//2.7: 将rect 添加到画布上
rect.setAttribute("x", i*100+50);
rect.setAttribute("y", 400-item.value);
rect.setAttribute("width", 50);
rect.setAttribute("height", item.value);
s3.appendChild(rect);
}
//3: 打开链接 open()
xhr.open("get", "http://127.0.0.1:3000/sales", true);
//4: 发送请求
xhr.send(null);
</script>
```

svg 绘制图形特性

1: 所有图形默认只有填充色(黑色)

2: svg 图形的样式可以用元素属性声明, 也可以使用 css

样式声明, 但是 CSS 声明只能使用 SVG 专用样式, 不能

css 样式

如边框只能用 stroke 而不能使用 border

3: 图形可以使用 js 赋值, 但不能使用 html dom 只能使用

核心 dom

```
r5.width = 300; r5.height=100; error
```

```
r5.setAttribute("width",300); ok
```

```
r5.setAttribute("height",100); ok
```

4: 动态添加 svg 图形

(1)html 字符串拼接

```
var html = "<rect></rect>";
```

```
svg.innerHTML = html;
```

(2)创建对象

```
var rect = document.createElementNS(
"http://www.w3.org/2000/svg", "rect");
svg.appendChild(rect);
```

3.3:html5 新特性--svg-开发流程-圆形

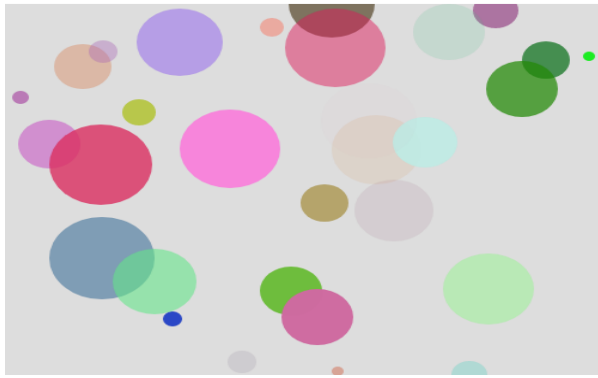
```
<circle r="" cx="" cy=""></circle>
```

练习:在画布上创建 30 个大小随机,位置随机,颜色随机,

透明度随机圆形

练习:当用户点击某一个圆形此圆形慢慢变大变淡直至消失 从 dom 删除

随机在画布上画圆，变淡变大，最后消失



```

<svg id="s3" width="500" height="400"></svg>
<script>
    var s3 = document.getElementById("s3"); // 获取画布
    //1: 创建一个循环30次
    for (var i=0; i<30; i++){
        //2: 创建圆
        var c = document.createElementNS("http://www.w3.org/2000/svg", "circle");
        //3: 大小随机
        c.setAttribute("r", rn(5,45)); // 在此范围内随机生成一个半径
        //4: 位置随机
        c.setAttribute("cx", rn(0,500)); // 随机的原点在一定的范围内
        c.setAttribute("cy", rn(0,400));
        //5: 填充颜色随机
        c.setAttribute("fill", rc(0,255));
        //6: 透明度随机
        c.setAttribute("fill-opacity", Math.random());
        //7: 将圆添加到画布
        s3.appendChild(c);
        //8: 点击事件
        c.onclick = function(){
            //8.0: 阻止当前圆形再次被点击调用函数
            this.onclick = null;
            //8.1 创建一个定时器
            var t = setInterval(()=>{
                //8.2 当前的圆变大
                var r = this.getAttribute("r");
                this.setAttribute("r", r*1.05); // 定时器改变圆形半径动态改变圆的大小
                //8.3: 变淡
                var p = this.getAttribute("fill-opacity");
                this.setAttribute("fill-opacity", p*0.8); // 获取当前透明度的值
                //8.3: 删除
                if(p<0.05){
                    s3.removeChild(this);
                    clearInterval(t);
                }
            }, 100);
        };
    }
    //大小随机 位置随机 颜色随机 透明
    //rn随机整形数值 min;max 范围
    function rn(min,max){
        var num = Math.random()*(max-min)+min;
        return Math.floor(num);
    }
    //rc随机颜色 rgb(255,255,255)
    function rc(min,max){
        var r = rn(min,max);
        var g = rn(min,max);
        var b = rn(min,max);
        return `rgb(${r},${g},${b})`;
    }
    // 拼接一个颜色的rgb0中三个随机的参数值
</script>

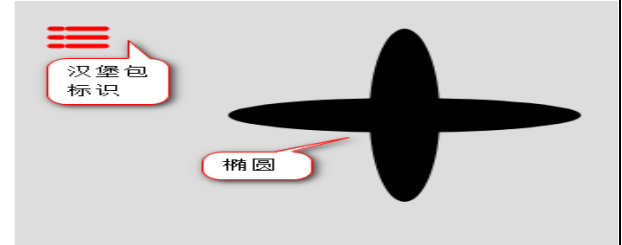
```

3.4:html5 新特性--svg-开发流程-椭圆

```

<ellipse rx="" ry="" cx="" cy=""></ellipse>
cx,cy 圆心
rx:水平半径 ry:垂直半径

```



```

<svg id="s3" width="500" height="400">
    <!-- 椭圆 -->
    <ellipse rx="100" ry="20" cx="250" cy="200"></ellipse>
    <!-- 直线 -->
    <line x1="0" y1="50" x2="100" y2="50" stroke="red"
        stroke-width="5" stroke-linecap="round"></line>
    <!-- 练习: 使用直线绘图, 绘图一个汉堡包 -->
    <g stroke-width="5" stroke="red" stroke-linecap="round">
        <line x1="50" y1="100" x2="55" y2="100"></line>
        <line x1="60" y1="100" x2="80" y2="100"></line>
        <line x1="50" y1="110" x2="55" y2="110"></line>
        <line x1="60" y1="110" x2="80" y2="110"></line>
        <line x1="50" y1="120" x2="55" y2="120"></line>
        <line x1="60" y1="120" x2="80" y2="120"></line>
    </g>
</svg>

```

3.5:html5 新特性--svg-开发流程-直线

```

<line x1="" y1="" x2="" y2=""></line>
x1,y1 起点坐标
x2,y2 终点坐标
练习:使用直线绘图图标 "汉堡包"

```

3.6:html5 新特性--svg-开发流程-折线

```

一条折线上可以有任意多个连续点
<polyline points="50,50 100,50 ..."
stroke="" fill="transparent"></polyline>

```

3.7:html5 新特性--svg-开发流程-文本/图像

```

svg 画布不能使用 span p 标签不可以用
<text font-size="" fill="" stroke=""> 文本内容
</text>
<image xlink:href=" x.png" x="" y="" width=""
height=""/>

```

HTML 5 day05

3:今天学习的内容

3.1:html5 新特性--svg--文本图像

```

<text font-size="" fill="" stroke=""> 文本内容
</text>
<image xlink:href=" 图片地址 " x="" y=""
width="" height="" />

```

3.2:html5 新特性--svg--渐变对象

```

<defs> 定义特效对象:
    <linearGradient id="g3" x1="" y1="" x2=""
y2="">
        <stop offset="" stop-color="" />
    </linearGradient>
</defs>

```



```
<ANY fill="url(#g3)"></ANY>
```

```
<svg id="s3" width="500" height="400">
  <defs>
    <linearGradient id="g3" x1="0%" y1="100%" x2="100%" y2="0%">
      <stop offset="0%" stop-color="red"></stop>
      <stop offset="100%" stop-color="blue"></stop>
    </linearGradient>
  </defs>
  <text font-size="39" x="250" y="200" fill="url(#g3)">Tarena</text>
```

以对角线的方向开始渐变
起始的坐标位
结束的坐标位置
从什么颜色到什么颜色，每一帧的变化
在文档中引入过滤器

3.3:html5 新特性--svg--滤镜(高斯模糊滤镜)

```
<defs>
  <filter id="f3">
    <feGaussianBlur stdDeviation="模糊度" />
  </filter>
</defs>
<ANY filter="url(#f3)"></ANY>
```

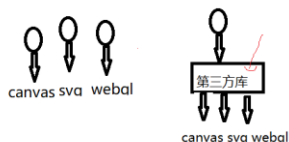
<https://developer.mozilla.org/zh-CN/docs/Web/SVG/Element/filter>

```
<svg id="s3" width="500" height="400">
  <defs>
    <filter id="f3">
      <feGaussianBlur stdDeviation="3"></feGaussianBlur>
    </filter>
  </defs>
  <text font-size="70" y="100">Tarena</text>
  <text font-size="70" y="200" filter="url(#f3)">Tarena</text>
```

模糊度

3.4:html5 新特性-canvas/svg--第三方绘图库--(重点)

<https://d3js.org/> d3 开源免费库
<https://echarts.baidu.com/> 百度开源免费库



3.5:html5 新特性-第三方绘图库-echarts

- (1)-下载并且在程序引入 echarts.js 文件
- (2) - 创建图形容器 `<div id="main" style="width;height"></div>`
- (3)-通过 js 程序获取容器并且创建 echarts 对象

```
var main = document.getElementById("main");
```

```
var mychart = echarts.init(main)
```

(4)-创建配置项

```
var option = {
```

```
  title:{text:"echart 入门示例"},//标题
```

```
  xAxis:{data:["衬衫","袜子","雪纺衫"]},
```

```
  yAxis:{},
```

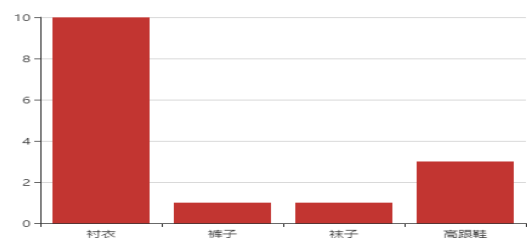
```
  series:[{type:"bar",data:[100,1,3]}]
```

```
} //数据源
```

(4) -将配置项添加 echarts 对象中

```
mychart.setOption(option);
```

echart 示例一



```
<body>
  <div id="main" style="width:500px;height:400px"></div>
  <script>
    //1: 获取容器
    var main = document.getElementById("main");
    //2: 创建echarts对象
    var mycharts = echarts.init(main);
    //3: 创建配置项
    var option = {
      title:{text:"echart 示例一"},
      xAxis:{data:["衬衣","裤子","袜子","高跟鞋"]},
      yAxis:{},
      series:[{type:"bar",data:[10,1,1,3]}]
    }
    //4: 将配置项添加echarts对象中
    mycharts.setOption(option);
  </script>
</body>
```

x轴的数据显示项
图形样式
柱形图
数据源

3.6:html5 新特性-第三方绘图库-折线图

```
series:[{type:"line"}]
```

3.7:html5 新特性-第三方绘图库-仪表盘图

```
series:[{
  type:"gauge",
  detail:{formatter:'{value}%'}
  data:[{value:32,name:"任务完成率"}]
}]
```



```

<div id="main" style="width:500px ; height:400px"></div>
<script>
//1:创建容器
var main = document.getElementById("main")
//2: 获取容器创建echart对象
var mycharts = echarts.init(main);
//3: 创建option 配置项
var option = {
  series:[{
    type:'gauge', //只有数据源
    detail:{formatter:'{value}'}, //仪表图类型
    data:[{value:32,name:"汽车速度"}] //指针位置
  }]
};
//4: 将option添加到echarts对象
mycharts.setOption(option);
</script>

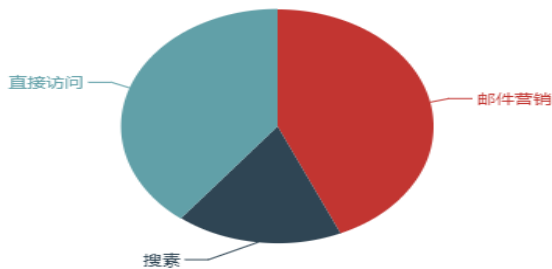
```

3.8:html5 新特性-第三方绘图库-饼图

```

series:[{
  type:"pie",
  radius:"50%", //半径
  center:["50%","50%"], //圆心[水平,垂直]
  data:[{value:325,name:"邮件营销"},{value:129,name:"搜索"}]
}]

```



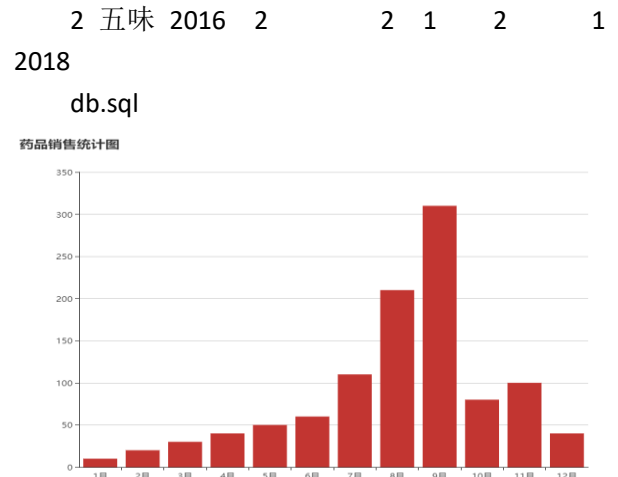
```

<script src="echarts.min.js"></script>
</head>
<body>
<div id="main" style="width:500px ; height:400px;
border:1px solid red"></div>
<script>
var main = document.getElementById("main");
var mycharts = echarts.init(main);
var option = {
  series:[{
    type:"pie", //圆饼图形
    radius:"50%", //半径
    center:["50%","50%"], //圆心
    data:[{value:325,name:"邮件营销"},{value:129,name:"搜索"}, //x轴和y轴的位置
    {value:295,name:"直接访问"}] //每一部分用对象表示
  }]
};
mycharts.setOption(option);
</script>

```

练习 1:医疗行业销售统图

药品名称	1	2	3	4	5	6	..12	年份	生产日期	保质期
六味地黄丸	2	1						2018	2017	1
功能一:创建药品表										
创建药品表						创建销售表				
id	name	ctime	rtime					id	yid	m[月]
sales[销售量] y										
1	六味	2017	1					1	1	2
2018										



服务端数据

```

1 #为药品销售统计图创建二张表
2 #1: 药品表 t_yaopin
3 #id 药品编号
4 #name 药品名称
5 #ctime 药品生产日期
6 #rtime 药品保质期
7 #2: 销售表 t_sales
8 #id 销售id
9 #yid 药品id
10 #y 销售年份
11 #m 销售月份
12 #s 销售量
13 CREATE TABLE t_yaopin(
14   id INT PRIMARY KEY AUTO_INCREMENT,
15   name VARCHAR(255),
16   ctime DATETIME,
17   rtime INT
18 );
19 CREATE TABLE t_sales(
20   id INT PRIMARY KEY AUTO_INCREMENT,
21   yid INT,
22   y INT,
23   m INT,
24   s DECIMAL(15,2)
25 );
26 INSERT INTO t_yaopin VALUES(null,"痢特灵",now(),1);
27 INSERT INTO t_sales VALUES(null,1,2018,1,10);
28 INSERT INTO t_sales VALUES(null,1,2018,2,20);
29 INSERT INTO t_sales VALUES(null,1,2018,3,30);
30 INSERT INTO t_sales VALUES(null,1,2018,4,40);
31 INSERT INTO t_sales VALUES(null,1,2018,5,50);
32 INSERT INTO t_sales VALUES(null,1,2018,6,60);
33 INSERT INTO t_sales VALUES(null,1,2018,7,110);
34 INSERT INTO t_sales VALUES(null,1,2018,8,210);
35 INSERT INTO t_sales VALUES(null,1,2018,9,310);
36 INSERT INTO t_sales VALUES(null,1,2018,10,80);
37 INSERT INTO t_sales VALUES(null,1,2018,11,100);
38 INSERT INTO t_sales VALUES(null,1,2018,12,40);
39
40 SELECT y.name,s.y,s.m,s.s
41 FROM t_yaopin y,t_sales s
42 WHERE y.id = s.yid AND y.id = 1;

```

```
//药品销售统计图
const mysql = require("mysql");
const express = require("express");
var pool = mysql.createPool({
  host:"127.0.0.1",
  password:"",
  user:"root",
  database:"xz"
});
var app = express();
app.listen(3000);
app.use(express.static("public"));
app.get("/sales",(req,res)=>{
  var sql = " SELECT y.name,s.y,s.m,s.s";
  sql+=" FROM t_yaopin y,t_sales s";
  sql+=" WHERE y.id = s.yid AND y.name = '痢特灵'";
  pool.query(sql,(err,result)=>{
    if(err) throw err;
    res.send(result);
  })
})
})
```

绘制统计图

```
<div id="main" style="width:800px;height:600px"></div>
<script>
  var main = document.getElementById("main"); //获取容器
  var mycharts = echarts.init(main); //创建一个echarts对象
  var xhr = new XMLHttpRequest();
  console.log(1111);
  xhr.onreadystatechange = function(){
    if(xhr.readyState == 4 && xhr.status == 200 ){
      var html = xhr.responseText;
      var rows = JSON.parse(html);
      console.log(rows);
      var mrows = [] //保存销售月份
      var srows = [] //保存销售额度
      for(var item of rows){
        mrows.push(item.m+"月"); //保存月份的数组
        srows.push(item.s); //保存销售额
      }
      //创建option的配置项
      var option = {
        title:{text:"药品销售统计图"},
        xAxis:{data:mrows},
        yAxis:{},
        series:[{type:"bar",data:srows}]
      };
      //将option添加到echarts对象中即可
      mycharts.setOption(option);
    }
  }
  var url = "http://127.0.0.1:3000/sales";
  xhr.open("get",url,true);
  xhr.send(null);
</script>
```

3.9:html5 新特性--地理定位

geolocation:地理定位使用 js 获取当前浏览器所处地理坐标

(经度;纬度;海拔;速度)数据,用于实例 LBS Location Base Service

基于位置服务:饿了么,滴滴打车..

手机浏览器如何获取定位信息

(1)首选手机 GPS 芯片与网络连接定位精度在米

(2)次选手机通基站

HTML5 中提供一个新对象,用于获取当前浏览

器定位信息

```
window.navigator.geolocation{
  getCurrentPosition:fn 获取当前定位
  watchPosition:fn 监听位置变化
  clearPosition:fn 取消监听
}
```

国内三家地理定位服务商

(1)百度地图

(2)高德

(3)腾讯地图

3.10:html5 新特性--地理定位--百度地图

-注册百度开发者帐户 (手机)

<http://lbsyun.baidu.com/>

-百度分配 AccessKey 访问密钥

u70A5pnNrRtRT1XAgwM5jL2YIVj1Gv97

-在自己网站中嵌入百度地图 API

```
<script src="http://api.map.baidu.com/ak=你的密钥"></script>
```

```
<div id="container"></div>
```

```
<script>
```

1:创建地图对象

```
var map = new BMap.Map("container")
```

2:创建坐标点

```
var point = new BMap.Point(经度,纬度);
```

3:初始化地图并且指定地图显示级别 1-19

```
map.centerAndZoom(point,15)
```

```
</script>
```

```
<!DOCTYPE html>
<html>
<head>
  <meta name="viewport" content="initial-scale=1.0, user-scalable=no" />
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <title>Hello, World</title>
  <style type="text/css">
    html{height:100%;}
    body{height:100%;margin:0px;padding:0px;}
    #container{height:100%;}
  </style>
  <script type="text/javascript">
    src="http://api.map.baidu.com/api?v=2.0&ak=u70A5pnNrRtRT1XAgwM5jL2YIVj1Gv97"
    //v2.0版本的引用方式: src="http://api.map.baidu.com/api?v=2.0&ak=您的密钥"
  </script>
</head>
<body>
  <div id="container"></div>
  <script type="text/javascript">
    var map = new BMap.Map("container");
    // 创建地图实例
    var point = new BMap.Point(116.398814,39.915799);
    // 创建点坐标
    map.centerAndZoom(point, 15);
    // 初始化地图,设置中心点坐标和地图级别
    // 导航控件
    map.addControl(new BMap.NavigationControl());
    //缩放控件
    map.addControl(new BMap.ScaleControl());
    //地图类型控件
    map.addControl(new BMap.MapTypeControl());
  </script>
```

3.11:html5 新特性--拖放 API

Drag & Drop 拖动和释放

HTML5 为拖放操作提供 7 个事件,分为两组
拖动源对象(会动)

dragstart 拖动开始

drag 拖动中

dragend 拖动结束

整个过程:dragstart*1+drag*n+dragend*1

拖动目标对象(不会动)

dragenter 拖动进入

dragover 拖动悬停

dragleave 拖动离开

drop 拖动释放

整个过程 1:dragenter*1+dragover*n+dragleave*1

整个过程 2:dragenter*1+dragover*n+drop*1

注意事项:dragover 事件有默认行为悬停结束后立即

触发离开事件

解决问题:阻止 dragover 事件默认行为

#(2)创建新线程执行 1.js

</script>

<button></button>

注意事项 1:Worker 不能双击网页运行,报错

Failed to construct 'Worker'

解决问题:必须在网络情况下运行
http://127..

注意事项 2:浏览器不允许 Worker 执行任何 DOM/BOM 对象.

原因:浏览器只允许 UI 主线程操作 DOM/BOM,类似 JQUERY

程序不能在 Worker 执行.

3.2: html5 新特性---web Worker--数据传递

-Worker 线程可以发送消息给 UI 线程

--Worker 发送消息

postMessage(stringMsg)

--UI 接收消息

var w = new Worker("1.js");

w.onmessage = function(e){e.data}

-UI 线程可以发送消息发 Worker 线程

--UI 发送消息

var w = new Worker("1.js");

w.postMessage(stringMsg)

--Worker 接收

onmessage = function(e){e.data}

练习:1:在网页创建 input 请输入数值 100

2:在 input 创建按钮 [开始计算累加]

3:在按钮下 <div>计算结果:5050</div>

4:我们希望通过 Worker 完成计算工作

HTML 5 day06

3.1:html5 新特性---web Worker

程序:指可以被 CPU 执行代码存储磁盘中 1.js
1.html

进程:指程序被 OS 调用内存中并且分配执行空间

线程:线程是进程内部执行单位

chrome 浏览器线程模型

一个 chrome 进程内至少有 6 个线程,可以向 web 服务器发起

请求 -- 资源请求线程

还有一个线程负责将所有资源绘制浏览器上
并且执行 js 程序

--UI 主线程

<button></button>

<script src="1.js"></script>

<button></button>

现象:js 扩行过程中,按钮 1 可见按钮 2 不可见

原因:浏览器执行 js 的绘制按钮同一个线程

解决方法:创建一个新线程由它执行耗时 js 任务,UI 线程

负责网页渲染.

<button></button>

<script>

new Worker("1.js");

#以上代码完成二个任务

#(1)创建 Worker 对象

3.3:html5 新特性---web Storage

在浏览器中存储当前用户专有数据:购物车,定制样式

在客户端存储数据可以使用技术

(1)cookie 技术:浏览器兼容性好,不能超过 4kb,操作复杂

(2)Flash 存储:依赖于 Flash 播放器

(3)H5 WebStorage:不能超过 8MB,操作简单

(4)IndexDB:可能存储大量数据,还不是标准技术

Session 会话

"操作过程"

浏览器打开某一个网站**第一个页面(会话开始)**中
间可能打开多个网页(**会话中**)直到关闭浏览器(**会话结束**)

整个过程称为"**浏览器与 web 服务器一次会话**"

WebStorage 技术提供二个对象存储数据

① sessionStorage

将数据保存 sessionStorage 会话开始保存数据, 同一个会话中

其它网页可以获取 sessionStorage 数据, 一旦关闭浏览器(会话结束)

sessionStorage 数据清空对象失效

作用: 同一个会话中所有页面共享数据

示例: 登录用户名, 昵称

-sessionStorage[key] = value; 保存数据

-sessionStorage.setItem(key,value); 保存数据

-var value = sessionStorage[key]; 获取数据 value

-var value = sessionStorage.getItem(key); 获取数据 value

-sessionStorage.removeItem(key); 删除指定 key/value

-sessionStorage.clear(); 清除数据

-sessionStorage.length; 数据个数

-var key = sessionStorage.key(i); 获取数据 key

练习 1: 03.html 将数据保存

sessionStorage

04.html 获取 sessionStorage

练习 2: 完成用户登录判断

index.html 右上角提示 "请登录" 超链接

```
<body>
<h3>首页</h3>
<div class="content">
  <a href="login.html">请登录</a>
</div>
<script>
  var content = document.querySelector(".content");
  //1: 获取sessionStorage 数据
  var uname = sessionStorage.getItem("uname");
  //2: 如果用户存在
  if(uname){
    //3: 获取保存用户名
    var html = "欢迎回来"+uname+"<a href='logout.html'>退出</a>";
    //4: 修改content 欢迎回来 ***
    // <a href="">退出</a>
    content.innerHTML = html;
  }
</script>
```

login.html 输入用户名, 密码 提交按钮

"登录成功, 3s 后自动跳转首页"

setTimeout+location.href

```
<body>
<h3>登录页</h3>
<form action="#">
  用户名: <input type="text" name="uname" id="uname"><br>
  密码: <input type="password" name="upwd" id="upwd"><br>
  <!--<button id="btn">登录</button-->
  <input type="button" value="登录" id="btn">
</form>
<script>
  //1: 获取元素 用户名和按钮
  var uname = document.getElementById("uname");
  var upwd = document.getElementById("upwd");
  var btn = document.getElementById("btn");
  //2: 为按钮绑定点击事件
  btn.onclick = function(){
    //3: 将用户名保存 sessionStorage
    sessionStorage.setItem("uname",uname.value);
    sessionStorage.setItem("upwd",upwd.value);
    alert("3s跳转到首页");
    setTimeout(function(){
      //4: 3s跳转首页
      location.href = "index.html";
    },3000)
  }
</script>
```

button 和 submit 都有默认行为, 所以我们都用 input button

3s之后本地跳转到index的页面

index.html 右上角 "欢迎回来 xxx 退出登录"

logout.html 提示 "您已退出 3s 自动跳转首页"

```
<h3>退出页</h3>
<script>
  //1: 清除 sessionStorage uname
  sessionStorage.removeItem("uname");
  //2: 3s 跳转 index.html
  alert("退出登录");
  setTimeout(function(){
    location.href = "index.html";
  },3000)
</script>
```

退出登录的时候清除 sessionStorage 保存的用户名

3s之后本地跳转到首页

index.html 右上角 "请登录" 超链接

② localStorage

本地存储(跨会话级别存储)--永久保存

作用: 购物车

- localStorage[key] = value; 保存数据

- localStorage.setItem(key,value); 保存/修改数据

-var value = localStorage[key]; 获取数据 value

-var value = localStorage.getItem(key); 获取数据 value

- localStorage.removeItem(key); 删除指定 key/value

- localStorage.clear(); 清除数据

- localStorage.length; 数据个数

-var key = localStorage.key(i); 获取数据 key

注意:

localStorage 如果数据发生修改,触发一次 **window.onstorage**

事件,可以监听此事件,实现监听 **localStorage** 数据改变的目标

不能监听 **sessionStorage**

练习 1:保存和获取数据 08.html 09.html

练习 2:用户自定义样式

创建 **index.html** 有一个下拉菜单"请选择您喜欢主题"

"蔚兰天空", "芭比公主", "暗黑主题" 各对应一个 class 名称

.blue{background:#ddf,color:#33a};

.pink{background:#fdf,color:#a3a};

.dark{background:#333,color:#eee};

用户选中某个项 (select.onChange) 当前 body 元素指定样式

this.value "blue"

document.body.className = "blue"

```
<style>
  .blue{ background: #ddf;color: #33a }
  .pink{ background: #fdf;color: #a3a }
  .dark{ background: #333;color: #eee }
</style>
</head>
<body>
  <h3>用户自定义样式</h3>
  <select id="select">
    <option value="blue">--请选择--</option>
    <option value="blue">蔚兰天空</option>
    <option value="pink">芭比公主</option>
    <option value="dark">暗黑主题</option>
  </select>
  <script>
    var select = document.getElementById("select");
    select.onChange = function(){
      //this.value = "blue";
      var v = this.value;
      document.body.className = v;
      localStorage.setItem("style",v);
    };
    var style = localStorage.getItem("style");
    if(style){
      document.body.className = style;
    }
  </script>
```

当点击下拉选的时候保存当前选择的背景样式在localStorage中,永久保存

获取 localStorage 保存的样式

如果localStorage保存了背景样式,那么就把这个样式赋给body的样式, class

当这个页面的主体颜色改变时,用户中心的背景也随着改变 **usercenter.html**

```
<style>
  .blue{ background: #ddf;color: #33a }
  .pink{ background: #fdf;color: #a3a }
  .dark{ background: #333;color: #eee }
</style>
</head>
<body>
  <h3>用户中心</h3>
  <script>
    //1: 获取定义的样式 应用当前网页
    var style = localStorage.getItem("style");
    if(style){
      document.body.className = style;
    }
    //2: 绑定事件window.onstorage
    window.onstorage = function(){
      var style = localStorage.getItem("style");
      document.body.className = style;
    }
    //3: 重新获取样式立即应用
  </script>
```

获取自定义样式应用

window.onStorage监听localStorage的对象,当localStorage改变事件自动触发,重新获取localStorage的值,在应用到页面中

HTML 5 day07

3.3:html5 新特性—WebSocket

http 网络协议:

协议网络工作标准依靠软件实例(node.js 服务器/浏览器)

http:工作模型:请求-响应,只有客户端先发起请求,服务器才会返回 (3次握手 4次挥手)

响应消息,没有请求就没有响应

http 作用: 传输网页中资源 (html;css;js;img;avi;mp3....)

俗称 http 网页搬运工

http 协议在有一些应用中不适合, 股票实时走势图

解决问题:

websocket 协议:属于"广播-收听"

客户端连接到服务器上就不再断开永久连接,双方都可以

随时向对方发送消息.

ws:适合:股票走势图,在线聊天室

3.4:html5 新特性--WebSocket--实现此功能

-ws 服务器[java/php/node.js]

--02_ws_server.js

websocket 服务器

--需要下载第三方模块 [ws][async-limiter]

指定监听端口 9001

①接收客户请求

②向对方发送消息

③接收对方消息

1: -ws 客户端

html5 中新对象 WebSocket 发请求 发消息 接收消息

(1) --创建 websocket 对象[创建对象;发请求建立连接]

var socket

=new WebSocket("ws://127.0.0.1:9001");

(2) --向服务器发送消息

socket.send(stringMsg);

(3) --接收服务器消息

socket.onmessage = function(e){e.data}

(4) --关闭连接

socket.close();

4:小游戏(大鱼和小鱼[canvas])

分析游戏:

大鱼游动时可以吃食物, 如果吃到蓝色食物

100 分 吃橙色食 200 分

大鱼喂小鱼

4.1:游戏目录结构

game -- 游戏所有文件都保存在此处

index.html 唯一 html

src 保存游戏中的图片

大鱼眼睛: bigEye0.png bigEye1.png

大鱼身体: bigSwim0.png ... bigSwim7.png

大鱼尾巴: bigTail0.png .. bigTail7.png

蓝色食物: blue.png

橙色食物: fruit.png

大海背景: background.jpg

js 保存游戏中的程序

main.js main.js 游戏入口

commonFunctions.js **公共函数库文件**

background.js 游戏背景

mom.js 大鱼

baby.js 小鱼

ane.js 海葵

friut.js 食物

....

4.2:游戏唯一 index.html

创建二个画布大小一相同一个在前一个在后 800*600

前面画布:大鱼;小鱼;分数... z-index:1

后面画布:背景;海葵;食物... z-index:0

加载所有 commonFunction.js main.js

background.js

复制图片 src

4.3:创建游戏程序入口 main.js

创建游戏所有需要用到对象(背景;食物;大鱼;..)

将所有对象画在画布上

4.4:大海背景图

src/background.jpg 绘制第二画布

1:main.js 声明全局变量保存图片

var bgPic;

2:init 创建图片对象下载图片

bgPic = new Image();

bgPic.src = "src/background.jpg"

3:将绘制图片放在函数 background.js

drawBackground(){

ctx2.drawImage(bgPic,0,0)

}

4:创建定时器 100 绘制画布上所有元素

定时器:setInterval(fn,100);

智能定时器:requestAnimationFrame(fn)

兼容性差:解决问题 commontFunctions.js

自定义兼容性很好函数

数:requestAnimFrame(fn)

main.js gameloop

function gameloop(){

requestAnimFrame(gameloop);

drawBackground();

}

常见错误:

(1) background.jpg:1 404 (Not Found)

原因:图片路径不正确

图片不存在

4.5:海葵

一共 50 根海葵

海葵基本

高度 200

海葵与海葵之间水平间距 16

始点坐标 x+16+random

终点坐标

在 js 目录创建文件 ane.js

大鱼小鱼程序