

JavaScript及Web网页前端开发

第11讲 webapi及复习

唐大仕

相关网站

■ canvas 画布

- https://www.w3school.com.cn/tags/html_ref_canvas.asp

- 官网 <https://www.w3.org/TR/2dcontext/>

■ web storage 存储

- https://www.w3school.com.cn/html5/html_5_webstorage.asp

■ geolocation 地理定位

- https://www.w3school.com.cn/html5/html_5_geolocation.asp

课程进度

11 webapi综合应用及复习

10 ajax及node

9 Bootstrap

8 Jquery

5 HTML5

6 CSS3

7 Javascript进阶

2 HTML

3 CSS

4 Javascript基础

1 序论

本讲内容

■ Web API （了解）

- storage、location

■ Web绘图 （了解）

- canvas、svg、echarts

■ 项目构建 （了解）

- parcel、rollup、webpack

■ 课程复习

Web API

HTML5 Web API

■ API

- API (Application Programming Interface, 应用程序编程接口)

■ html5 增加了一些api, 常用的有:

- storage 存储 (更快)

- location 位置 (更准)

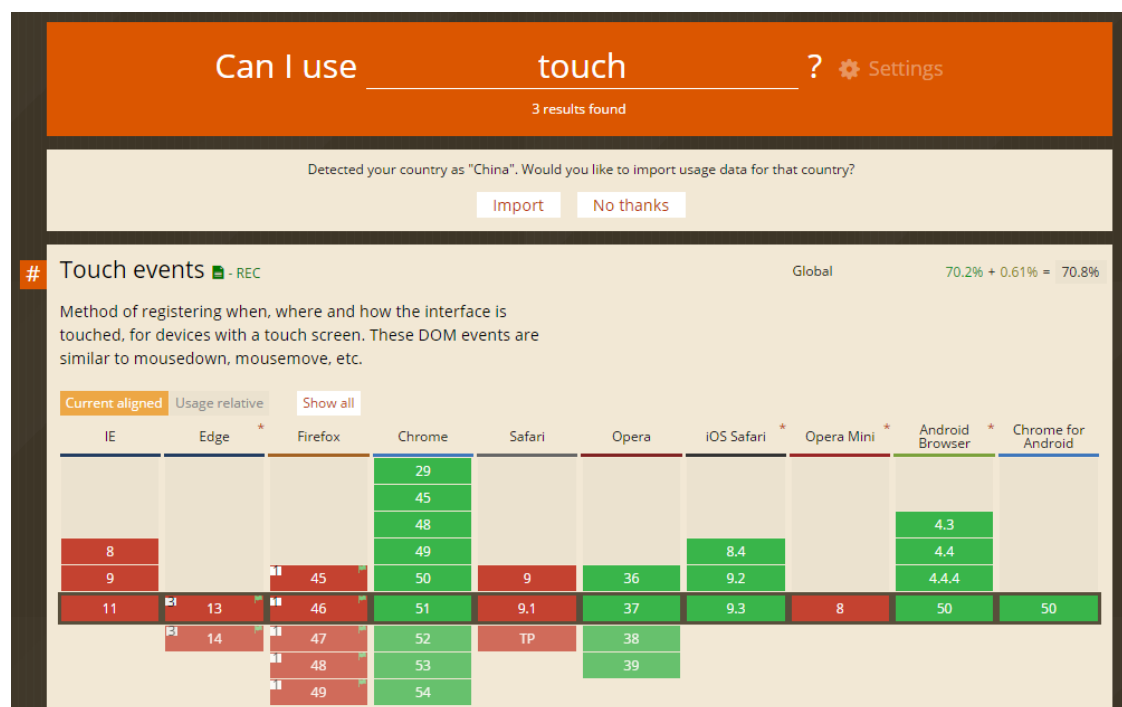
- canvas 绘图 (更美)

- 其他, 如全屏、摄像头、录音、震动、电池状态等

Can I use

■ <https://caniuse.com>

□ 可以查询某个特性在不同浏览器中是否可用



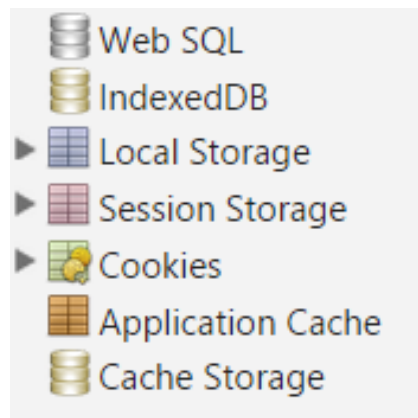
Web存储

■ Web存储

- 又称本地存储、离线存储
- 其作用是：保存状态、减轻网络负载

■ 小技巧

- 在chrome中 F12, Application
- 或Resources



Web存储

■ HTML5 API中有几种存储方式

□ cookie 早期的存储方式

- key=value; 有有效期; 存储量有限; 自动传递到服务端

□ localStorage, sessionStorage

- key=value的方式; 分别是长期、会话期

□ IndexedDB, WebSQL

- 以数据库方式存储

□ 以上几种方式都是按网站（域名）来区分的

Web 存储

HTML5 提供了两种在客户端存储数据的新方法：

- localStorage - 用于持久化的本地存储，除非主动删除数据，否则数据是永远不会过期。
- sessionStorage - 用于存储一个会话（session）中的数据，这些数据只有在同一个会话中的页面才能访问，当会话结束后数据也随之销毁。

```
localStorage.length;  
localStorage.key(index);  
localStorage.setItem('foo', 'bar'); localStorage.getItem('foo');  
localStorage.removeItem('foo');  
localStorage.clear();
```

以前，这些都是由 Cookie 完成的。但是 Cookie 有4KB的大小限制，而且会随HTTP请求一起被传递，无形中拖慢网页速度而且效率不高。

localStorage的方法

■基本用法

- localStorage.setItem(key,value) 增加数据
- localStorage.getItem(key) 获取值
- localStorage.key(n): 获取第n个key
- localStorage.removeItem(key): 删除一个数据
- localStorage.clear() 清空

变通用法

■ localStorage.变量名

- 获取、设置变量

■ 如果是对象

- localStorage.setItem(变量名, JSON.stringify(对象))

- var a = JSON.parse(localStorage.getItem(变量名))

应用示例

■ 示例1：记录用户的选项

- 要点：localStorage

■ 示例2：留言本小应用

- 复习：

- json
- 日期
- 数组
- 循环

geolocation

■ 地理位置

- 在移动应用中的地位越来越重要
- 现在浏览器得到了广泛支持，特别是移动的浏览器

navigator.geolocation

■基本的对象是

- navigator.geolocation

- 它有一个getCurrentPosition(success, error, option)方法

```
navigator.geolocation.getCurrentPosition(function(location) {  
    alert(location.coords.latitude + "," + location.coords.longitude);  
}, function(){  
    alert("未能获取地理位置");  
});
```

前提条件

■ 获得地理位置的前提条件

- 页面须用http方式访问，不能以文件方式打开
- 要用户允许浏览器访问位置
- 浏览器要能获得位置

■ chrome中模拟位置

- 在新的chrome中，F12, 点右边的三点按钮，选more tools---sensors--emulate geolocation coordinates
- 北京的位置： lat(纬度) 39.91 lon(经度) 116.4

应用示例

■ 访问百度地图

□ 要点:

- 引用baidu map api的javascript
- 使用相应的类, 如 BMap.Map, BMap.Point, BMap.Marker

□ 百度map api

- 使用介绍 <https://lbsyun.baidu.com/index.php?title=jspopular>
- 地图类参考 <https://developer.baidu.com/map/reference/index.php>
- 服务类参考 <https://lbsyun.baidu.com/index.php?title=webapi>

Web 绘图

Canvas 画布元素

传统的网页，总是使用GIF或者JPEG来显示图像，这种图形是需要事先画好的“静态”的图像。而Canvas，则是用Javascript的一种绘图手段。

可以用它来画图、合成图象、或做动画。

```
<canvas id="canvas" width="300" height="300"></canvas>

<script>
  var ctx = document.getElementById("canvas").getContext("2d");
  ctx.fillRect(20, 25, 150, 100);

  ctx.beginPath();
  ctx.arc(220, 110, 100, Math.PI * 1/2, Math.PI * 3/2);
  ctx.lineWidth = 15;
  ctx.lineCap = 'round';
  ctx.strokeStyle = 'rgba(255, 127, 0, 0.5)';
  ctx.stroke();
</script>
```



什么是 Canvas?

■ Canvas元素

- 是HTML5中新增的一个重要元素
- 专门用来绘制图形
- 在页面放置一个canvas元素，就相当于在页面上放置了一块“[画布](#)”，可以在其中进行图形的描绘

■ 利用 JavaScript 编写绘画的脚本

在HTML页面中放置Canvas

■ `<canvas id="canvas" width="400" height="300"> </canvas>`

□ 属性(特别注意, 这3个属性都是必须的)

- id: canvas元素的ID
- width: 宽度
- height: 高度

绘制

■ 公共步骤：

- 获取Canvas元素 `var canvas = document.getElementById(id);`
- 取得上下文 (context) `var ctx = canvas.getContext("2d");`

■ 绘制矩形

- `ctx.fillRect(50, 50, 100, 100);` 绘制矩形
 - `ctx.strokeRect(50, 50, 100, 100);` 绘制矩形边框
 - `ctx.clearRect`——使矩形区域中的颜色全部变为透明。
-
- (矩形起点的横坐标, 纵坐标, 宽, 高)

属性

- fillStyle —— 填充的颜色
- strokeStyle —— 线的颜色
- lineWidth —— 线的宽度

绘制圆

■ 步骤

□ 开始创建路径

```
context.beginPath();
```

□ 创建图形的路径
true);

```
context.arc(100, 100, 30, 0, Math.PI*2,
```

□ 路径创建完成后，关闭路径

```
context.closePath();
```

□ 设定绘制样式，调用绘制方法，绘制路径。

➤ `context.fillStyle='rgba(255, 0, 0, 0.5)';`

绘制线条

- `moveTo (x , y) ;` 绘制直线的起点坐标;
- `lineTo (x , y) ;` 指定直线的终点坐标, 绘制直线;
 - 注释: 当再一次使用`lineTo`方法时, 当前终点坐标会转化为起点坐标。

线性渐变



■ 获取LinearGradient对象

- `var grd = context.createLinearGradient(0 , 0 , 400 , 300);`

■ 设定渐变颜色

- `grd.addColorStop(0 , 'rgb(255 , 0 , 0)');`

- `grd.addColorStop(1 , 'rgb(0 , 255 , 0)');`

■ 设定绘制样式

- `context.fillStyle = grd;`

- `context.fillRect(0 , 0 , 400 , 300);`

cheat sheet

- <https://simon.html5.org/dump/html5-canvas-cheat-sheet.html>

使用SVG

■ SVG是矢量图

- svg中的g

- svg中的path

 - **M** = move to **L** = line to **H** = horizontal line to **V** = vertical line to

 - **C** = curve to **S** = smooth curve to **Q** = quadratic Bézier curve

 - **T** = smooth quadratic Bézier curve to **A** = elliptical Arc

 - **Z** = closepath

■ 示例：

- 北大网页上的SVG及动画

综合应用：eCharts

■ eCharts

□ 官网 <https://echarts.apache.org/>

□ 要点

- `<script src="https://cdn.bootcss.com/echarts/3.1.10/echarts.common.js"></script>`
- `var myChart = echarts.init(document.getElementById('main'));`
- `var option = {`
- `series: [{`
- `type: 'pie',`
- `radius: ['50%', '70%'],`
- `data: mydata`
- `}]`
- `};`
- `myChart.setOption(option);`

□ 可以使用ajax/mockjax来设置数据

前端项目开发

ES5、ES6的几个新语法

- 箭头函数
- 类及继承
- 对象的简写及拆解
- ...的使用
- 新的方法
- 新的数据类型 (Set Map等)

项目构建

■ 打包（构建）的作用

- 对项目文件进行合并、压缩、Uglify等

■ 打包的工具

- rollup, webpack, gulp, grunt 可定制化强，要进行复杂的配置
- fis3, parceljs 相对简单

parcel的使用

■ 安装node.js

- 官网 <https://nodejs.org>

■ 安装parcel.js

- `npm install -g parcel-bundler`

■ 打包

- 建立项目文件 `npm init -y`
- 调试 `parcel *.html` 或 `parcel ./**/*.html`
- 访问看效果 `http://localhost:1234/`
- 打包 `parcel build *.html --public-url .`

使用前端框架

■ 流行的前端框架

- Vue React Angular 等
- 双向绑定、虚拟DOM、单页应用

■ Vue的使用

- 直接引用vue.js 或者用vue-cli脚手架编译
- Vue2 或 Vue3
- 能用于Vue的UI组件库：Element-ui, Vant-ui, Mint UI 等

课程复习

课程内容

11 webapi综合应用及复习

10 ajax及node

9 Bootstrap

8 Jquery

5 HTML5

6 CSS3

7 Javascript进阶

2 HTML

3 CSS

4 Javascript基础

1 序论

HTML及HTML5

■ HTML

- 文<p> 图 表<table> <dl> 链<a>
- 块级<div> 行内
- 表单<form>

■ HTML5

- 语义元素 <header> <footer> <aside> <section> <article> <addr>
- 多媒体 <audio> <video>
- 输入 <input>

CSS (重点)

■ 选择器(重点)

- p .class # div >

■ 字体 font (重点)

- font-family: "微软雅黑", Arial, sans-serif; font-size: 10px; text-align: center;

■ 颜色 color、背景 background、边框 border(重点)

- border: solid red 3px; border-top: none;

■ 位置、大小(重点的重点)

- display: block 或 inline 或 inline-block; display:none不显示

- float: 用于div并列, clear用于不再并列, clearfix 经典的解决

- position: absolute 其父级元素用 relative

- width, height, margin, padding 其中 margin:0 auto; 加上宽度, 就可以居中显示

CSS3增强

■ 装饰

- ❑ border-radius: 5px; 50%
- ❑ box-shadow: 10px 20px 5px red;
- ❑ background: linear-gradient(20%, center, 30%, center, from(blue), to(yellow)) no-repeat;

■ 行为(重点)

- ❑ :hover :focus :checked
- ❑ :after :before

■ 图标字体、sprite图

CSS3动态

■ 变换(重点)

- `transform: rotate(30deg);`
- `transform: scale(0.5, 2.0);`
- `transform: skew(-30deg);`
- `transform: translate(30px, 0);`

■ 过渡(重点)

- `transition: all 1s ease-out ;`

■ 动画(重点)

- `animation: myAnim 3s linear 1s infinite alternate;`
- `@keyframes myAnim { 0%{...} 20%{...} 100%{...} }`

javascript 语言(重点)

- 变量 `var`

- 运算符 `+` `-` `*` `/` `%` `&&` `||` `!` 连接`+`

- 语句

 - `if() { } else { }`

 - `for(var i=0; i<a.length; i++) { }`

- 数组 `a = [];`

- 函数 `function foo(){ }`

javascript对象(重点)

■ 对象

- 三要素：属性、方法、事件

■ json对象

- { name:"Li Ming", age: 180, hobby:["football", "dance"] }

■ 内置对象

- Math .sqrt(a) .pow(a, n) .random() .round(a) (重点)
- Date var now = new Date(), now.getHours(), now.getMinutes() (重点)
- window parseFloat(a) parseInt(a) eval(a) (重点)
- setInterval(fun, 1000) alert prompt confirm (重点)
- location, navigator
- document .getElementById (重点)

jQuery (重点)

- `$(function(){ 初始化 });`

- 选择

 - `$("#mydiv")` `$("ul>li")` `$(".class")`

- 操作

 - `.html()` `.append()`

 - `.val()` `.attr()` `.css()` `.hide()` `.show()`

- 事件

 - `$(xxx).on("click", "li", function(){ });` `$(xxx).hover(function(){...}, function(){...});`

- 效果

 - `slideUp()` `slideDown()` `slideToggle()` `fadeIn()` `fadeOut()`

- 过滤

 - `.parent()` `.children()` `.siblings()` `.next()` `.find()` `.first()` `.last()` `.eq(n)`

bootstrap

■ 响应式网页(重点)

- ❑ `<meta name="viewport" content="width=device-width, initial-scale=1.0">`
- ❑ `@media screen and (max-width: 980px) { }`

■ 栅格系统

- ❑ `.row .col-md-6 .col-sm-8`

■ 样式

- ❑ `.btn .btn-primary`

■ 组件

- ❑ `dialog form calendar`

ajax

■ XmlHttpRequest

■ jQuery.ajax

- \$.ajax({ url: , data: , success:function(t){ } }) (重点)

- \$.get(url,data,function), (重点)

- \$.post, \$.getJSON, \$(...).load(url, data)

■ mock.js

- Mock.mock(url, {})

WEB API (了解)

■ canvas

- `var ctx = xxxx.getContext("2d");`
- `ctx.fill` `ctx.stroke`
- `ctx.fillStyle` `ctx.strokeStyle` `ctx.lineWidth`

■ localStorage

- `.setItem(k, v)` `.getItem(k)`

■ geolocation

- `.getCurrentLocation(function(loc){`
 - `loc.coords.longitude,` `.latitude`

总结

- html/htm5
- css/css3
- javascript
- jquery
- bootstrap 响应式
- ajax
- web api

js之深

■ 只用6个字符来写出所有的JavaScript

□ [] () ! +

□ <http://www.jsfuck.com/>

谢谢！