李文华

QQ: 2180265268

Email: 2180265268@qq.com

第一阶段： HTML + CSS

第二阶段： JS + DOM&BOM

第三阶段： jQuery + Server + HTTP + AJAX + HTML5

第四阶段： 框架

提示：第三阶段特点——凌乱

# 浏览器是一根 筋，

# 只会从上往下读取数据

# 5.30 jQ简介 选择器

**(jQ是用来操作DOM的)**

1. 复习：DOM

DOM：Document Object Model，是浏览器根据HTML文档创建的一批对象，以及这些对象间关系。

W3C DOM标准分为三部分：

(1)核心DOM

(2)HTML DOM

(3)XML DOM

程序员掌握DOM为了操作所有的DOM元素。

DOM的常见操作：

(1)查找元素/对象（增删减除）

document.getElementById()

document.getElementsByXxx()

document.querySelector()

e.parentNode.parentNode.nextSibling.children()

(2)操作元素的属性

e.setAttribute(name, value)

e.getAttribute(name);

(3)操作元素的内容

e.innerHTML

e.innerText e.textContent

(4)操作元素的值

e.value

(5)添加元素

document.createElement(name)

parent.appendChild( newChild )

(6)删除元素

parent.removeChild( oldChild )

(7)替换元素

parent.replaceChild( newChild, oldChild )

(8)事件处理

"冒泡模型” "捕获-冒泡模型"

(9)动画

**DOM标准的两个问题：**

1. 操作名称/思路略显**繁琐**
2. 操作存在浏览器**兼容性问题**
3. jQuery概述

jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. jQuery has changed the way that millions of people write JavaScript.

**jQuery是一个JS函数库，用于简化DOM操作，屏蔽浏览器兼容性问题。**

格言：Write Less， Do More

版本：

1.12.x 文件较大，功能少，兼容老IE

2.2.x 文件较小，功能多，放弃老IE

使用思路——与DOM操作一样：

先查找元素，再操作元素

1. ZenCoding功能

div +TAB

div#id +TAB

div.class +TAB

div\*3 +TAB

ul>li +TAB

lorem +TAB 假文生成

3.jQuery.js引入位置

(1)在head标签内引入

(2)在body标签的最后引入——**推荐写法**

1. **jQuery对象和DOM对象的关系和区别？——难点&重点(！面试！)**

**DOM对象(W3C DOM标准)**：每个HTML标签在浏览器中都对应一个DOM对象。

**jQuery对象（jQuery函数返回的对象）**： $()/jQuery()函数的**返回值**是一个“jQuery对象”——典型的**类数组对象**，其中封装着所有**选定**的DOM对象。

|

如$(‘li’) ----> 封装着所有**li**的DOM对象。

注意：jQuery对象不能使用DOM对象的属性和方法！

DOM对象不能使用jQuery对象的属性和方法！

**注意区分哪个属性/方法属于DOM/jQuery**

1. 如何把DOM对象封装为jQuery对象：

**$( domObj )**

1. 如何把jQuery对象转换为DOM对象：

**$('div')[ 0 ]**

**$('.box').get( 2 )**

**补充：查询指定的DOM元素在jq对象中的下标**

**1, $('#box').index(‘.li2’)**

**2, Console.log(‘%d+%d=%d’,’21’,’10’,’31’);**

**Console.log(‘总数：%d’,’55’);**

**Console.log(‘%s+%s’,’aa’,’bb’);**

**3， $().each();**

**例如:**

**$(‘li‘).each(function(/\* 默认参数为：每个元素，以及下标 \*/ i，element){**

**Console.log(.......);**

**});**

练习：创建一个按钮，“加载更多”，点击后，下面显示一个隐藏的列表，同时按钮上的文字变为“加载完成”。

5.使用jQuery查找元素——选择器

提示：jQuery使用选择器查找元素，支持CSS3中的选择器，同时还要一些CSS中没有的选择器。只要是jQuery手册中列出的选择器，基本不存在兼容性问题。

1. 基本选择器

#id

.class

element

\*

s1,s2,s3

1. 层级选择器

parent child 后代选择器

parent > child 子代选择器

prev + nextOne 下一个兄弟选择器

prev ~ nextAll 后续所有兄弟选择器

1. 过滤选择器——基本过滤

提示：把所有符合条件的元素放在一个大的集合中，从该大集合进行过滤。序号从0开始

:first

:last

:odd 第奇数个

:even 第偶数个

:eq(i) 第i个

:gt(i) 大于i

:lt(i) 小于i

1. 过滤选择器——内容过滤

:contains(内容) 包含指定内容

:empty 不包含任何内容

:has(s) 有符合特定选择器的

:parent 有内容的

1. 过滤选择器——属性过滤

[name] 具有指定属性

[name=value] 具有指定属性和值

[name!=value] 具有指定属性，不能有指定的值

[name^=start] 具有指定属性，以指定值开头

[name$=end] 具有指定属性，以指定值结尾

[name\*=sub] 具有指定属性，值包含指定内容

1. 过滤选择器——可见性过滤

:hidden display:none + type="hidden"

:visible

1. 子元素选择器

提示：把每个选中的元素放在其父元素对应的小组中，进行筛选；下标从1开始！不是0！！

:first-child 第一个子元素

:last-child 最后一个子元素

:nth-child (数字/odd/even/形如3n+1表达式) 第n个子元素

:only-child 唯一子元素

1. 表单元素选择器

:text 选中<input type="text">

:password 选中<input type="password">

:radio

:checkbox

:submit

:reset

:button 选中<input type="button">

:file 选中<input type="file">

:hidden 选中<input type="hidden">

:image 选中<input type="image">，不能选中<img>

:input 选中表单中所有的输入元素，包括input、select、textarea、button等

1. 表单元素状态选择器

:enabled 选中所有启用的元素，不带disabled属性的元素

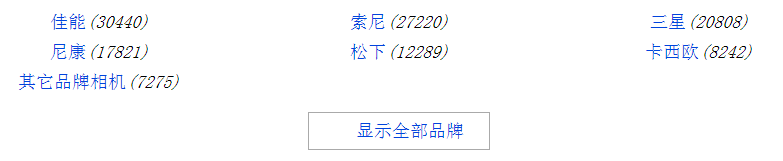
:disabled 选中表单中带 disabled 属性的元素 <input disabled>

:checked 选中<input type="radio" checked> <input type="checkbox" checked>

:selected 选中<option selected>

1603A班 练习内容： jQuery选择器

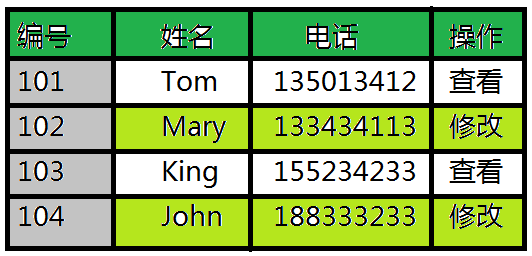
(1)默认只显示部分内容，以及“其它品牌相机”



点击“显示全部品牌”按钮后，所有隐藏的内容都被显示出来：



(2)使用jQuery控制样式，显示出如下外观的表格



# 5.31操作元素(鄙视)事件处理

复习：

jQuery

概述：是一个JS函数库，用于简化DOM操作

查找元素：

操作元素： 属性、内容、值、样式、遍历

添加元素：

删除元素：

替换元素：

事件

动画

AJAX

HTML5

1. 复习：jQuery中的元素查找——选择器——基础

(1)基本选择器

#id .class element \* s1,s2

(2)层级选择器

parent child

parent > child

prev + nextOne

prev ~ nextAll

(3)过滤选择器——基本

:first :last :odd :even :eq(i) :gt(i) :lt(i)

大集合中的第i个，下标从0开始

(4)过滤选择器——内容

:contains("")

:parent

:empty

:has(s)

(5)过滤选择器——属性

[name]

[name="userName"]

[name!="userName"]

[name\*="user"]

[name^="user"]

[name$="user"]

(6)过滤选择器——可见性

:hidden

:visible

(7)子元素选择器

:first-child

:last-child

:nth-child(odd/even/2/3n+1)

:only-child

每个父元素中第i个，下标从1开始

(8)表单元素选择器

:text

:password

:radio

:checkbox

:file

:hidden

:submit

:reset

:image

:input input/select/button/textarea

(9)表单元素状态选择器

:disabled

:enabled

:checked

:selected

1. 操作元素

（1）**操作元素的属性**

jq.attr(name,value) //e.setAttribute(name,value)

jq.attr(name) //e.getAttribute(name)

jq.removeAttr(name)

（2）**操作元素的值**

jq.val('xx'); //input.value = 'xx'

jq.val(); //var v = input.value;

（3）**操作元素的内容**

jq.html('xx'); //e.innerHTML = 'xx';

var h = jq.html(); //var h = e.innerHTML ;

jq.text('xx'); //e.innerText/textContent='xx';

var t = jq.text(); //var t = e.innerText/textContent;

（4）**遍历元素**

jq.parent() //e.parentNode

jq.children([s]) 找直接子元素

jq.find(s) 找所有的子元素(直接+间接)

jq.next() 下一个兄弟

jq.nextAll() 后面所有的兄弟

jq.prev() 上一个兄弟

jq.prevAll() 上面所有的兄弟

jq.siblings() 上面+下面所有的兄弟

提示：上述方法有些情形下可以使用选择器代替；有些情况下，只能使用上述方法，例如：

$('button').click(){

//让当前元素后面的span兄弟隐藏

$(this).next('span').css('display', 'none')

}

(5)**操作元素的样式**

jq.css( name ) 获取样式的值，$('div').css('color');

jq.css( name, value ) 设置样式的值，$('div').css('color', 'red');

jq.css( {name:value, name:value} )

jq.addClass( className ) 为选定元素添加一个class

jq.removeClass( className ) 为选定元素删除一个class

jq.hasClass( className ) 判断选定元素是否有指定class

**jq.toggleClass( className )**  切换class：有则删除，无则添加

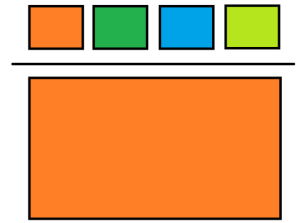
练习： 实现一个双态按钮

打开第1张图片对应的大图 <a href=""></a>

<img src="img/1.jpg" alt="1">

<img src="img/2.jpg" alt="2">

午间练习：



3.添加新元素

DOM中添加新元素：

var child = document.createElement('div');

parent.appendChild( child );

例子：

var \_li = document.createElement('li');  
 var \_a= document.createElement('a');  
 $.data(\_a,'notes',msg[i])  
 \_li.appendChild(\_a);  
 $('#\*\*\*').append(\_li);

jQuery中添加新元素——八种：

方法1：

var child = $('<div id="" class="">ABCD</div>');

$('.parent').append( child ); //追加到最后

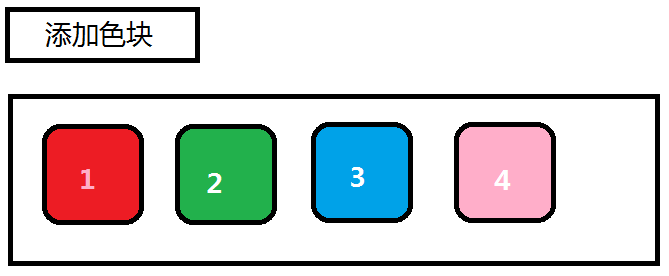
//$('.parent').prepend( child ); //追加到最开始

方法2：

$('.parent').append( '<li><a>ABCD</a></li>' )

$('.parent').prepend( '<li><a>ABCD</a></li>' )

练习：点击按钮，添加随机颜色块



background: rgb(234, 135, 10);

4.删除元素

DOM中：

parent.removeChild( child );

jQuery中：

$('div').remove(); //删除当前选定元素

5.替换元素

DOM中：

parent.replaceChild( newChild, oldChild );

jQuery中：

方法1：

newChild.replaceAll( oldChild );

oldChild.replaceWith( newChild );

6.jQuery中的事件处理

面试题：jQuery中如下几个函数的区别：bind 、click 、delegate、on、live？

jQuery的历史上，先后出现了如下的事件处理函数：

**(1)bind & unbind ：** 为选定元素绑定特定事件的处理函数

jq.bind( '事件名', fn )

**(2)one：**为选定元素绑定一次事件处理函数

jq.one( '事件名', fn )

**(3)click / dblclick / mouseover / mouseout .... ：** bind()函数的简化

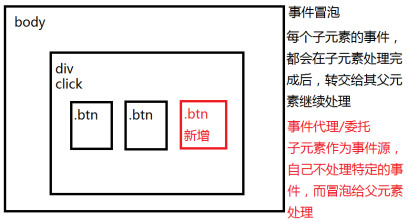
jq.click( fn ) 等价于 jq.bind('click', fn)

注意：DOM所有事件中，只有一部分(23个事件)有简化版本

上述3种方法只能为DOM树已经存在的元素绑定监听函数，无法为后创建的元素绑定监听函数。

**(4)delegate**：子元素委托父元素**代理**处理自己的事件**(两个优点：一个事件代替多个事件；)**

$('父元素').delegate('子元素', '事件名', fn)



**(5)live**：指定子元素把指定的事件全部委托给document对象——jQuery1.8之后删除了此方法。

$(子元素).live('事件名', fn)

**(6)on**：可以实现类似bind / delegate的所有功能

**类似bind的使用方法（把监听函数绑定在事件源上）：**

$('子元素').on('事件名', fn)

**类似delegate的使用方法（把监听函数委托在父元素上）：**

$('父元素').on('事件名', '子元素', fn)

**补充知识点：**

**1,触发： .trigger(‘事件名’); //或 .click(); --->事件名（）；**

如：

$(‘button’).click(function(){

Console.log($(this).html());

});

再设一个按钮触发所有button的事件

$(‘#btn’).click(function(){

//$(‘button’).trigger();

$(‘button’).click();

});

**2, 待所有DOM节点加载完成即触发，html+js, 调用时间比( window.onload() )早。**

**底层实现：window.addEventListener('DOMContentLoaded', fn, false)**

**$(document).ready(function(){ -> == $().ready(fn) == $( fn )**

**.......**

**});**

1. **阻止事件发生：（多用于a的事件）**

**event.preventDefault();**

**4. 克隆元素：**

**jQ： .clone(false/true /\*是否复制元素的事件处理函数\*/);**

**DOM: this.cloneNode(false/true); -->浅 深克隆（浅：只复制元素没内容， 深：复制元素+内容）**

//等同于window.onload(function(){ ---->等页面所有内容加载完后触发,(html+js+flash+音频/视频+图片...)

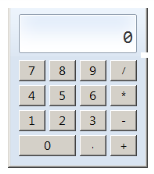
.......

});

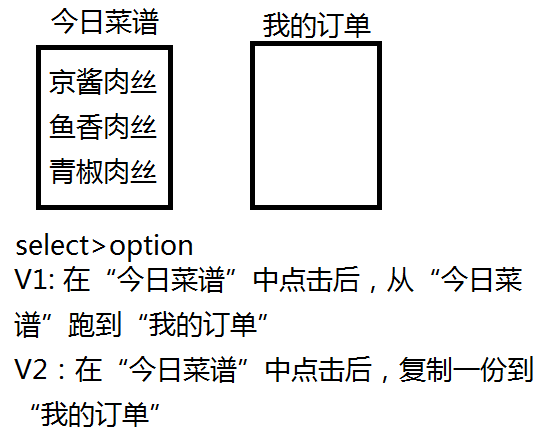
1603A 练习内容：jQueryDOM处理和事件绑定

(1)实现计算器界面和输入功能

要求：所有的按钮必须把事件委托给父元素，从而减少监听函数的数量



(2)点击菜品，进行下单



# 6.1事件 动画 jQ UI

复习：

**jQuery是一个JS函数库，用于简化DOM操作。**

使用思路：先查找元素，再操作元素

**查找元素**

$('选择器') => jQuery对象：类数组对象

基本选择器 #id .class element \* s1,s2

层级选择器 p c p>c p+n p~n

过滤选择器—基本 :first :last :odd :even :eq :gt :lt

过滤选择器—内容 :contains() :parent :empty :has()

过滤选择器—属性 [name] [name=value]

过滤选择器—可见性 :visible :hidden

子元素选择器 :first-child :last-child :nth-child :only-child

表单元素选择器 :text :password ...

表单元素状态选择器 :enabled :disabled :checked :selected

**操作元素——属性**

jq.attr()

**操作元素——值**

jq.val()

**操作元素——内容**

jq.html() jq.text()

**操作元素——样式**

jq.css( )

jq.addClass() jq.removeClass() jq.hasClass() toggleClass()

**遍历元素**

jq.parent() **jq.children()(只能找：直接子元素) jq.find()（找全部）**

jq.next() jq.prev() jq.siblings( )

**添加元素**

jq.append( jq/string ) c.appendTo( p )

jq.prepend( jq/string ) c.prependTo( p )

p.before( n ) n.after( p )

p.inertBefore( n ) n.insertAfter( p )

**删除元素**

jq.remove( )

jq.empty( ) 清空当前选定元素的内容

**替换元素**

newChild.replaceAll( oldChild )

oldChild.replaceWith( newChild )

**事件处理**

(1)bind $('btn').bind('事件名', fn)

(2)one $('btn').one('事件名', fn)

(3)click/mouseover/keyup... $('btn').click( fn )

(4)delegate $('parent').deleagate('子元素','事件名',fn)

(5)live 已废弃

(6)on $('btn').on('事件名',fn) $('parent').on('事件名','子元素', fn)

**动画效果**

1.补充：jQuery中的事件处理

(1)如何使用代码触发事件——了解

$(..).trigger( '事件名' )

或者简写为：

$(..).click( )

(2)面试题：$(document).ready()和window.onload的区别？

**window.onload**：浏览器窗口中**所有的内容加载完成**，包括HTML + CSS + JS + 图片 + 视频/音频 + Flash...

**$(document).ready()**：待所有的**DOM节点**加载完成即触发，**HTML + JS，**底层的实现是:

**window.addEventListener('DOMContentLoaded', fn, false)**

**二者的不同：**

(1)$(document).ready()只等待HTML+JS加载完成即触发，window.onload只有待页面中的所有内容加载完成才触发，前者的触发时机要早于后者。

(2) $(document).ready()可以先后**绑定多个监听函数**，而window.onload只能绑定**一个**监听函数

说明：$(document).ready( fn ) 可以简化为如下形式：

$().ready(fn) 甚至可以进一步简化为：

$( fn )

2.jQuery中的动画函数——次重点

提示：下述动画函数都是“定时器”动画，使用定时器不停的修改特定的属性

**(1)定时器修改width/height/opacity实现动画**

show( string/number , [fn])

hide( string/number , [fn])

toggle( string/number, [fn])

**(2)定时器修改height实现动画**

slideUp(string/number , [fn]) 折叠收起

slideDown(string/number , [fn]) 折叠展开

slideToggle(string/number , [fn])

**(3)定时器修改opacity实现动画**

fadeIn(string/number , [fn]) 枯萎，淡入

fadeOut(string/number , [fn]) 淡出

fadeToggle(string/number , [fn])

午间练习：点击小星星，闪烁3次后消失

**(4)定时器修改*特定属性*实现动画**

animate( {属性名:值, 属性名:值}, string/number, [easing], [fn] )

**默认情况下**，animate函数可以对width / height / opacity / left / top等数字型的属性可以执行动画；非数字型属性不能执行动画，如display / position。此外，颜色类型的属性也不能执行动画

Animate-->相当于css的transition(ie10一下不兼容)

练习：

创建第一个小星星，点击后从左边跑到右边；

创建第二个小星星，点击后从上边跑到下边；

创建第三个小星星，点击后从左上角跑到右下角；

创建第四个小星星，点击后从左边跑到右边，再跑到下边；

3.jQueryUI框架 —— 了解，扩展知识

官网： http://jqueryui.com/

jQueryUI是一个基于jQuery编写HTML**组件库**，是一套jQuery的插件库。为jQuery提供了下面几个方面的增强：

1. **动画效果(Effects):**

1)animate函数支持颜色动画 2)show/hide支持更多特效

blind、bounce、clip、drop、explode、fold、highlight、

puff、pulsate、scale、shake、size、slide

1. **交互组件(Interactions)**

1)draggable 让一个元素可以在页面中任意拖动

2)resizable 让一个元素可以由用户改变大小

3)selectable 让一个列表中的元素可以被选中

4)sortable 让一个列表中的元素可以被排序

1. **页面小部件(Widgets)**

1)Accordion 手风琴组件

2)Autocomplete 自动完成组件

3)Datepicker 日期选择器

4)Dialog 显示一个对话框

5)Tooltip 把页面中所有的title属性显示为一个弹出式div

$(document).tooltip()

1603A班 练习内容：jQuery DOM操作和事件监听

购物车计算器：



要求：(1)点击“添加商品”按钮后，可以在表格中显示一个新的商品，价格随机生成，数量随机生成(整数)，可以自动显示出“小计”；

(2)“数量”显示在一个输入框中，只要用户的输入改变了，后续的“小计”随之改变；

(3)只要“添加商品”、“删除商品”被点击，或者“数量”被改变，必须重新计算出“购物车总金额”。

# 6.2 jq函数 jq插件

复习：

jQuery

day01：概述 查找元素-选择器

day02：操作元素(属性/值/内容/样式)、遍历、添加、删除、替换、克隆

day03：事件处理、动画函数、jQueryUI

day04：自定义插件、第三方插件

AJAX

HTML5

1.补充：jQuery对象——类数组对象的操作

$('li').length

$('li')[2]

$('li').get(2)

$('li').index( document.getElementById('i2') )

$('li').each( function(i, e){ } )

2.扩展jQuery——为jQuery提供插件函数——**重点&难点**

Plugin：插件，基于当前已有的代码，提供更多更丰富的功能。jQuery官方提供了100多个官方函数，用户以及第三方厂家可以编写更多的插件函数。

jQuery中插件函数分为两类：

1. **jQuery全局插件函数**

jQuery.函数名 = function(形参){ .... }

避免了全局对象的污染！

jQuery.lwh = {}; //命名/名称空间

jQuery.lwh.函数名 = function(形参){ }

避免污染全局对象和jQuery对象！

调用方法：

$.lwh.函数名( 实参 )

注意：jQuery全局插件函数本质就是之前的全局函数，为了避免污染全局对象，声明为jQuery成员。—— 可以处理任意的数据。

1. **jQuery对象的插件函数**

var s = $('td').sum( );

编写方法：

jQuery.fn.函数名 = function(){

//this 指代当前的jQuery对象

}

调用方法：

$('选择器').函数名( );

|  |
| --- |
| 面试题：$.each() 和 each()的区别？  $.each()是一个jQuery**全局函数**——可以遍历任意**集合数据**，如整数数组；  each()是一个jQuery**对象函数**——只能用于遍历当前**jQuery对象**中封装的DOM对象； |
| var scoreList = [58, 76, 90]; //为每个数字都+5分  $.each( scoreList, function(i, v){ scoreList[i]+=5 } ) |
| <ul>  <li>10</li>  <li>30</li>  <li>20</li>  </ul>  $('li').each(function(i, element){  var v = $(element).html( );  v = parseInt(v);  v+=5;  $(element).html(v);  }); |

练习：

1. 定义我的jQuery对象插件——下拉菜单
2. 定义我的jQuery对象插件——胶囊式导航

提示：第三方插件的使用无需记忆和背诵，重点掌握使用的思路和过程

2.学习第三方插件的使用——laydate

官网：<http://laydate.layui.com>

是一个日期选择插件，基于原生JS；不基于jQuery。

其它可用的日期插件：jQueryUI DatePicker、My97、layDate...

练习：**一个高校的网站，**让用户选择自己的生日，选择的日期得到的岁数必须在18~60之间

3.学习第三方插件的使用——jQueryValidator

表单验证插件：formValidator、jQuery.validator、easyForm、validate.js

官网：https://jqueryvalidation.org/

4.学习第三方插件的使用——Masonry

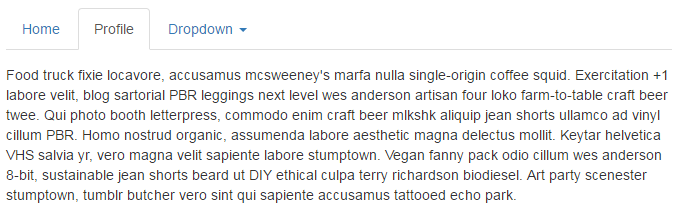
masonry：彩砖墙，瀑布流

官网：http://masonry.desandro.com/

Masonry是一个瀑布流插件，用于在页面中对区块进行“见缝插针”式布局

1603A 练习内容：为jQuery添加一个tabs（标签页）插件函数

使用方法： $('div.tabs').tabs( );



<div class="tabs">

<ul>

<li><a href="#tabs-1">十元套餐</a></li>

<li><a href="#tabs-2">二十元套餐</a></li>

<li><a href="#tabs-3">三十元套餐</a></li>

</ul>

<div id="tabs-1">

<p>Proin elit arcu, rNunc tristique tempus lectus.</p>

</div>

<div id="tabs-2">

<p>Morbi tincidunt, dui sit P tortor et purus.</p>

</div>

<div id="tabs-3">

<p>Mauris elugt.</p>

</div>

</div>

# 服务器

# 6.3 SQL 数据库

复习：

知识体系

1.复习：为jQuery添加插件函数

(1)全局函数

jQuery.函数名 = function( num/string/arr ){ }

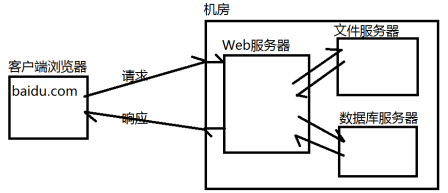
jQuery.函数名( num/string/arr );

(2)jQuery对象函数

jQuery.fn.函数名 = function( ){ }

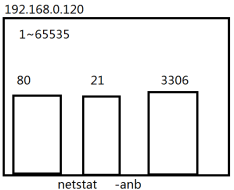
$('选择器').函数名( );

2.Web项目架构



客户端如何访问一台服务器，必须提供三个内容：

1. 协议
2. 主机地址或域名 (万网 / 美城 购买域名。 在命令台中输入：ipconfig 查自己pc的id地址)
3. 端口号



3.数据库服务器

作用：存储项目中的数据，如帖子、订单、通话记录.....

种类：网状数据库=>树型数据库=>关系型数据库=>对象型数据库

常见的关系型数据库软件：

(1)SQLite：微型数据库，用于移动设备

(2)PostgreSQL：中小型数据库，开源

(3)MySQL/MariaDB：中小型数据库，开源

(4)MS SQLServer：中型数据库，只运行Windows

(5)Oracle OracleDB：中大型数据库，商业

(6)IBM DB2：中大型数据库，商业

关系型数据库的物理结构：

1. 数据库**服务器端**：保存数据——机房

确定服务器是否启动，就看端口号是否打开！

1. 数据库**客户端**：向服务器端发送操作指令，增删改查数据——ATM终端

注意：客户端中是没有数据的！客户端用于向服务器发出指令——SQL命令。

·桌面版本客户端

·网页版本客户端

·命令行版本客户端

关系型数据库服务器中的数据在逻辑上的结构：—— 重点

**Server => Database => Table => Row => Column**

LAMP = Linux + Apache + MySQL + PHP

WAMP = Windows + Apache + MySQL + PHP

--------------------------------------------------------------------

XAMPP = Linux/Windows + Apache + MySQL + PHP + Perl

Cross OS

XAMPP是一款服务器软件的套装，包含多款服务器产品。

4.MySQL数据库服务器的使用

**(1)下载并安装MySQL服务器端软件**

启动入口：c:/xampp/mysql/bin/mysqld.exe Daemon精灵/守候

数据目录：c:/xampp/mysql/data 其中的数据不能丢失，不能用记事本打开

**(2)启动MySQL服务器端软件**

运行c:/xampp/mysql/bin/mysqld.exe 再提供一堆的运行参数即可；

也可以点击“启动”按钮。

只要确定3306端口被mysqld.exe占用即可！

**(3)下载并安装MySQL客户端软件**

命令行版客户端默认安装路径：c：/xampp/mysql/bin/mysql.exe

**(4)启动MySQL客户端软件，*远程连接机房中*的数据库服务器**

在命令行中输入：

c:/xampp/mysql/bin/mysql.exe -h127.0.0.1 -uroot -p -P3306

上述命令可简写为：

mysql -uroot

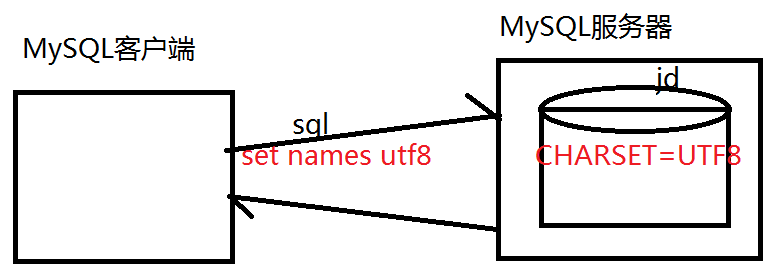
命令提示符变为： mysql> mariadb> 说明客户端已经成功连接到MySQL服务器

**(5)在客户端中输入SQL命令，提交给服务器来执行——增删改查数据**

SQL命令可以在命令行中一行一行的执行；

也可以编写在一个独立的.sql文件中，一次性交给服务器全部执行。

mysql> **source e:/jd.sql**  此句话不能加分号！！！



5.各种SQL命令的使用——最最重点 & 难点

SQL： Structured Query Language，结构化查询语句

SQL语句的分类：

(1)**DDL：Data Define Language，用于定义数据的结构——列**

DROP CREATE ALTER(删除/添加列) TRUNCATE(截断)

**(2)DML：Data Manipulate Language，用于操作表中的记录——行**

INSERT DELETE UPDATE

**(3)DQL：Data Query Language，用于查询表中的记录**

SELECT

**(4)DCL：Data Control Language，用于控制用户的权限**

GRANT(授予权限) REVOKE(收回权限)

6. Web服务器

作用：Web服务器用于向客户端提供Web内容，如.html、.css、.js、.jpg ....

分类：

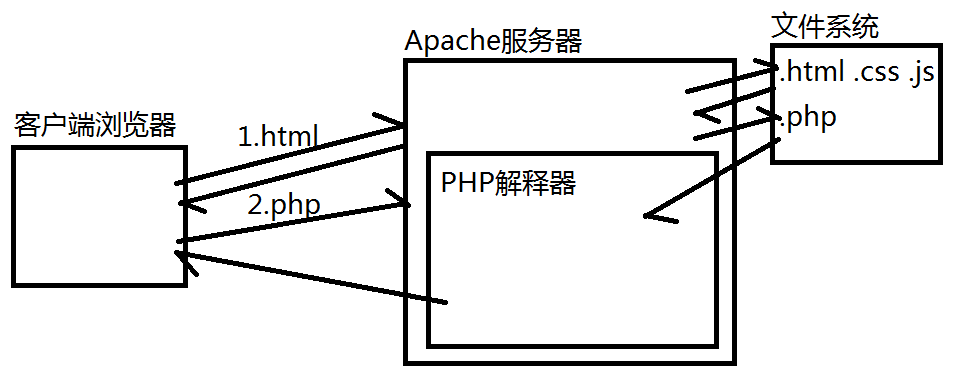
1. 静态Web服务器：只能提供一成不变的网页内容 : html / css / js / jpg / swf (photo shope)
2. 动态Web服务器：提供的内容在不同的时间、不同的用户访问都可能不同

JSP = HTML + Java

PHP = HTML + PHP Personal Home Page

ASPX = HTML + C#

7.如何架设一台PHP动态Web服务器



(1)下载并安装PHP语句的解释器

c:/xampp/php/php.exe

(2)下载并安装一款静态Web服务器，如Apache Httpd

c:/xampp/apache

(3)整合Apache和PHP解释器，修改Apache主配置文件

过程有点儿复杂........

提示：咱们安装的XAMPP已经把二者整合完毕

(4)启动Apache+PHP这款Web服务器

(5)编写网页，保存在一个特定的目录下，供客户端浏览器访问

所有的Web内容，都必须放在：

**c:/xampp/htdocs**

(6)客户端浏览器远程访问Web服务器

http://127.0.0.1:80

1603A 练习内容：SQL语句和PHP语句

(1)创建一个1.php网页，使用PHP语句向客户端输出1个\*

(2)创建一个2.php网页，使用PHP语句向客户端输出50个\*

(3)创建一个3.php网页，使用PHP语句向客户端输出5行10列个\*

(4)创建一个4.php网页，使用PHP语句向客户端输出“九九乘法表”

------------------------------------------------------------------------

(5)创建一个tarena.sql文件，按如下要求编写需要的SQL语句——重点！

删除并重建一个名为tarena的数据库；

创建一个保存部门信息的表dept，保存部门编号、名称、所在地等信息；

向部门表中插入三个部门的记录；

创建一个保存员工信息的表emp，保存员工的编号、姓名、工资、生日、是否已婚、电话号码、所在部门等信息；

向员工表中插入5行记录；

试着修改和删除某些员工的信息。

# 6.6 PHP SQL

**( var\_dump()--->显示出来的是一种独特的格式，用于开发过程中的调试。不可当数据发送 )**

6.3 php

DNS服务器

Sendmail服务器

Samba服务器

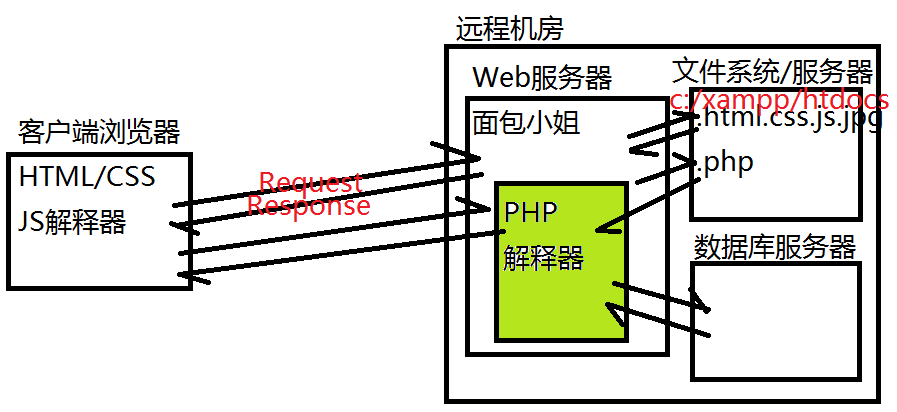
FTP服务器

Web服务器

SSH服务器

DB服务器

....



复习：

1.数据库服务器——MySQL

2.Web服务器

静态Web服务器：html / css / js / jpg / swf ...

动态Web服务器：.jsp / .aspx / .php

PHP服务器的搭建：

(1)服务器端安装PHP解释器——烤箱

c:/xampp/php/php.exe

(2)服务器端安装Apache Httpd静态Web服务器——面包小姐

c:/xampp/apache

(3)服务器端整合Apache和PHP解释器——教面包小姐使用烤箱

c:/xampp/apache/httpd.conf

(4)启动服务器——开门营业

c:/xampp/apache/httpd.exe

(5)服务器端编写网页，保存在c:/xampp/htdocs目录，此目录下不允许有中文和空格文件名/目录名

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

(6)客户端打开浏览器，输入服务器的URL地址，请求其中的网页

1.PHP语法学习

(1)概述

(2)搭建开发环境

(3)hello world

(4)数据类型

(5)变量和常量

(6)运算

(7)逻辑结构

(8)函数

(9)通用小程序

2.PHP中的数据类型

|  |
| --- |
| **重点掌握： string、array、foreach —— 注意与js的不同** |

PHP 支持 8 种原始数据类型。

四种标量类型（值类型）：

◦ boolean（布尔型） true使用echo输出1;false无输出

◦ integer（整型）

◦ float（浮点型，也称作 double)

◦ string（字符串） —— 注意字符串的两种语法

两种复合类型：

◦ array（数组） —— PHP中数组不是对象！没有toString()不能echo

◦ object（对象）

最后是两种特殊类型：

◦ resource（资源）

◦ NULL（无类型）

3.复习：数据库服务器

数据库服务器的作用：保存项目中的数据

MySQL的使用步骤：

(1)服务器端安装MySQL服务器程序：

c:/xampp/mysql/bin/mysqld.exe

其数据默认保存在 c:/xampp/mysql/data目录下

(2)服务器端启动MySQL服务器程序：

保证3306端口被mysqld.exe占用

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

(3)数据库管理员(客户端)远程连接MySQL服务器

c:/Users/Administrator> c:/xampp/mysql/bin/mysql.exe -uroot

(4)客户端登录到MySQL服务器后，提交SQL命令或提交.sql文件

mysql> source e:/jd.sql

练习：编写一个jd.sql文件，其中创建一个jd数据库，包含一个表产品表(product)，包含pid、name、price、birthday、isonsale，插入4行记录

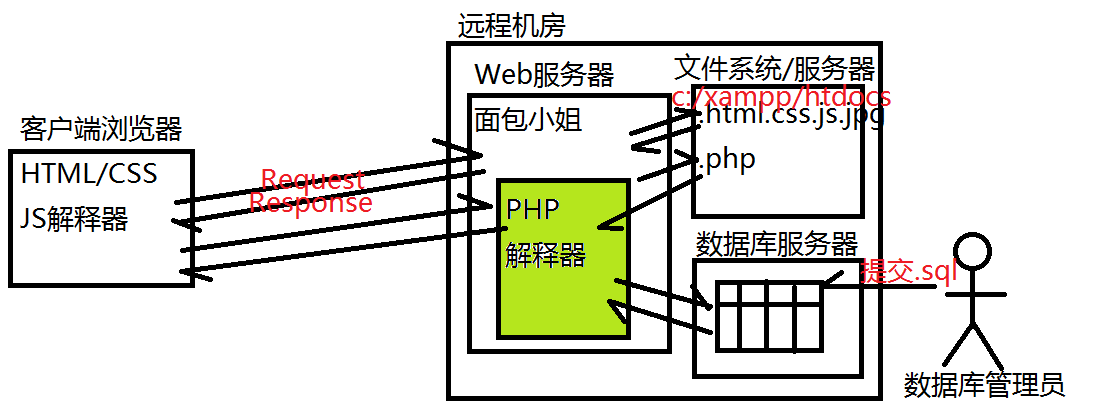
Ctrl+D 删除当前行(Delete Line)

Ctrl+Alt+↓ 复制当前行(Duplicate Line)

Alt+↑/↓ 向上/下移动当前行(Move Up / Move Down)

Ctrl+/ 注释当前行( Line Comment )

4.使用PHP网页增删改查数据库中的数据



CRUD：Create、Retrieve、Update、Delete，增删改查数据

实现对“产品表(product)”的CRUD的步骤：

(1)创建product\_add.html页面，为用户提供的一个添加新产品的表单

(2)创建product\_add.php页面，接收客户端提交的新产品的数据，连接到MySQL服务器，发送一条INSERT指令给服务器。

----------------------------------------------------------

(3)创建product\_del.html页面，为用户提供的一个表单，“请输入您要删除的产品编号：”

(4)创建product\_del.php页面，接收客户端提交的要删除的产品编号，连接到MySQL服务器，发送一条DELETE指令给服务器。

-----------------------------------------------------------

(5)创建product\_select.php页面，连接数据库，查询出所有的产品

5.PHP连接MySQL的步骤

提示：PHP做为客户端连接MySQL服务器过程与命令行中登录MySQL服务器的过程是一样。

(1)连接到MySQL服务器

$conn = mysqli\_connect('127.0.0.1', 'root', '', 'jd', 3306);

(2)提交SQL命令给服务器执行

$sql = "INSERT/DELETE/UPDATE/SELECT.....";

$result = mysqli\_query( $conn, $sql );

1603A 练习内容： 达内留言板

1. 编写一个tarena.sql文件，其中创建数据库tarena，创建一个留言表msg，有mid、userName、phone、content
2. 创建一个msg\_add.html，其中有一个留言的表单
3. 创建一个msg\_add.php，接收客户端提交的留言内容，添加到数据库
4. 创建一个msg\_select.php，在一个表格中显示出当前数据库中所有留言。

PHP中获得当前系统时间使用： time()，返回当前时间距离计算机元年多少秒钟

**spl**

[关于在Windows环境下使用cmd向mysql数据库中导入.sql数据表文件](http://blog.csdn.net/u010722980/article/details/51553478)

<http://blog.csdn.net/u010722980/article/details/51553478>

# 6.7 HTTP协议 重点

复习：

PHP的语法：

数据类型：

string boolean integer float

array object

resource NULL

变量和创建： $变量名 const

运算： 算术 比较 逻辑 赋值 拼接(.)

逻辑结构： for while foreach if ...

函数： function 函数名(){}

PHP连接MySQL服务器的步骤：

(1)$conn = mysqli\_connect(host, user, pwd, db, port);

(2)$result = mysqli\_query($conn, $sql)';

执行失败(如SQL语法错误)，一律返回FALSE；

执行成功： DML返回TRUE，DQL返回查询结果的描述对象

DDL: drop create alter truncate

DML: insert delete update

DQL: select

DCL: grant revoke

作业：留言板

1.客户端浏览器如何给服务器端的PHP页面传递数据

HTTP协议规定，客户端可以在URL后面追加请求参数，实现给服务器端PHP页面传递数据的功能：

*http://127.0.0.1/add.php?k1=v1&k2=v2&k3=v3*

服务器端add.php可以使用$\_REQUEST数组读取到上述数据。

2.基于单表CRUD操作

**C：添加**

product\_add.html

product.add.php

**R：获取**

product\_select.php

**D：删除**

product.del.php

**U：更新**

1. 创建 prodcut\_update\_select.php，执行一次查询操作，显示在一个很像是添加产品的表单中
2. 创建product\_update.php，执行产品修改操作

午间练习：学生成绩管理

包括：录入成绩 查询成绩 修改成绩

3.学习HTTP协议的目标

(1)调试AJAX请求中“看不见摸不着”的错误

(2)Web访问优化

4.URL、URI、URN

**URL：Unified Resource Locator，统一的资源定位符号**

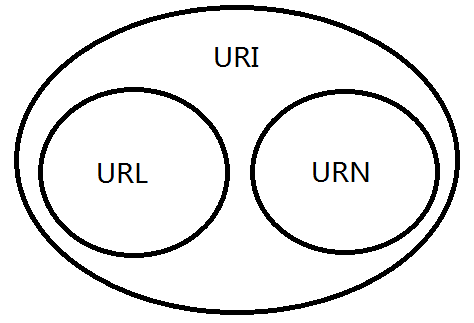
http://tmooc.cn/index.php

**URN：Unified Resource Naming**

mailto: admin@tarena.com

**URI：Unified Resource Identifier**

URI = URL + URN



绝对地址：

<a href="http://tmooc.cn/index.php"></a>

<a href="//tmooc.cn/index.php"></a>

相对地址：

<img src="img/1.jpg">

<link href="../img/favicon.ico">

注意：HEAD标签中的BASE标签可以重新指定相对URL的基础值

<head>

<base href="http://tmooc.cn /">

</head>

**5.HTTP协议详解——重点&难点**

HTTP协议，由IETF制定的Web客户端与Web服务器之间通讯所必需的消息格式。  **http://www.ietf.org/rfc/rfc2616.txt**

RFC：Request For Comment，意见征求稿

练习：使用Chrome的Network查看网络请求消息和响应消息的格式

超文本传输协议（HTTP，HyperText Transfer Protocol)是互联网上应用最为广泛的一种网络协议。所有的WWW文件都必须遵守这个标准。设计HTTP最初的目的是为了提供一种发布和接收HTML页面的方法。1960年美国人Ted Nelson构思了一种通过计算机处理文本信息的方法，并称之为超文本（hypertext）,这成为了HTTP超文本传输协议标准架构的发展根基。Ted Nelson组织协调万维网协会（World Wide Web Consortium）和互联网工程工作小组（Internet Engineering Task Force ）共同合作研究，最终发布了一系列的RFC，其中著名的RFC 2616定义了HTTP 1.1。

1996：HTTP/1.0

1999：HTTP/1.1

|  |
| --- |
| 面试题：HTTP/1.1相对于1.0有哪些改进？   1. **持久连接** Connection: keep-alive 在一次连接上可以传输多个请求-响应消息 2. **虚拟主机** Host: tmooc.cn 在同一个Web服务器上可以同时存在多个不同域名的站点 3. **代理连接**  Proxy: 通过中间的第三方计算机来访问目标网站 |

建立连接：三次握手 断开连接：四次挥手

6.HTTP请求消息的格式——重点

请求消息是客户端发给服务器的消息；内容分为四部分：

(1)请求起始行 请求方法[空格]请求URI[空格]协议版本

(2)请求头部 头部名: 头部值

(3)CRLF

(4)请求主体 k1=v1&k2=v2&k3=v3...

**HTTP请求方法：**

**GET、POST、HEAD、PUT、DELETE、**

**TRACE、OPTIONS、CONNECT**

|  |
| --- |
| 面试题：表单的GET请求和POST请求的区别？  (1)GET请求提交的数据在地址栏能看到，POST看不到  (2)GET请求把请求数据追加在请求**URI后面**，以?开头；POST请求把请求数据追加在**请求主体**中  (3)HTTP协议规定：请求消息起始行总长度不能超过1024字节；请求主体长度没有限制。GET请求提交的数据或是汉字一般也就几十个  (4)语义不同：GET表客户端想获取；POST表客户端想传递数据 |

|  |
| --- |
| GET /ajax\_day03/3.html HTTP/1.1  Host: 127.0.0.1  Connection: keep-alive |
| HEAD /ajax\_day03/3.html HTTP/1.1  Host: 127.0.0.1  Connection: keep-alive |
| DELETE /web/index\_new.html HTTP/1.1  Host: tmooc.cn  Connection: keep-alive |

7.HTTP响应消息的格式——重点

见思维导图

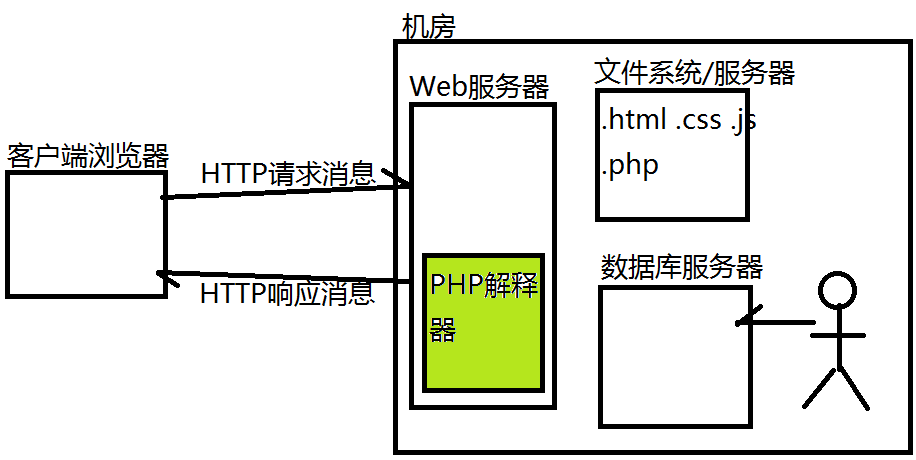
1603A 练习： 仿写TMOOC的注册和登录页面

1. 创建tmooc.sql，创建数据库tmooc，表users，包含uid、email、pwd、nickname、regtime（注册时间）等列。
2. 创建register.html，显示一个注册表单。
3. 创建register.php，用于接收并保存用户的注册信息，提示“注册成功”或“注册失败”。
4. 创建login.html，显示一个登录表单。
5. 创建login.php，用户接收客户端提交的登录信息，显示出“登录成功”后“用户名或密码错误”。

选作功能点：每次登录成功后，显示出自己上一次的登录时间

# 6.8 HTTP AJAX-xhr

复习：



HTTP协议学习目标：

1. 调试AJAX请求错误
2. Web访问优化

1.FORM元素的enctype属性和HTTP请求头

<form action="" method="POST" enctype="">

Encode Type：指定表单数据的编码类型，实际影响的就是POST请求的头部: Content-Type

可取值：

(1)text/plain：表单数据是普通的文本，未经编码——PHP服务器拒绝接收此种类型的请求主体数据！

**(2)application/x-www-form-urlencoded：**表单数据是经过编码后的文本

(3)multipart/form-data：表单数据中包含上传的文件

2.MIME类型

Multipurpose Internet Mail Extention，多用途的互联网邮件扩展

邮件附件的类型：

image/jpeg .jpg .jpeg

text/html .html .htm .xhtml

audio/mpeg3 .mp3 .mpeg3

text/css

appplication/javascript

application/xml

application/json

....

所有HTTP响应消息都需要告诉客户端当前主体的内容类型(MIME类型)

**3.HTTP访问优化——面试题**

网站性能优化最主要的就是要减少HTTP请求及每次响应中内容的长度。可以从连接过程中的下列方面加以考虑：

(1)域名解析

尽可能减少域名解析次数——减少跨站外部资源的引用

(2)创建连接

努力减少连接创建次数——使用持久连接避免重复连接

(3)发送请求

尽力减少请求次数——合理设置Expires时间、资源合并

(4)等待响应

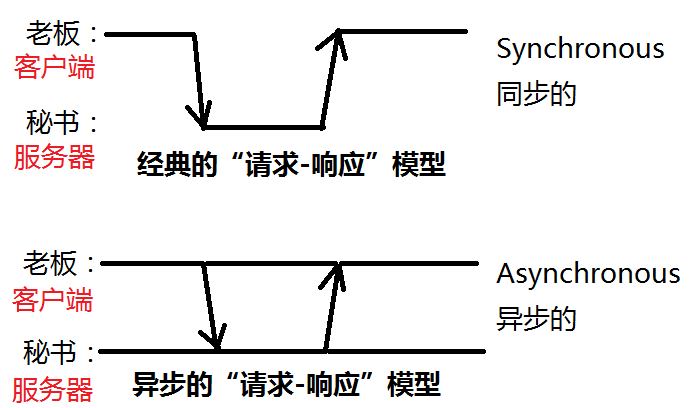
提高服务器端运行速度——提高数据运算及查询速度

(5)接收响应

尽可能减小响应数据长度——启用压缩

**4.AJAX——第三阶段最重点 ！**

Asynchronouse Javascript And XML：异步的JS和XML



异步：浏览器在浏览当前页面的同时，服务器也在运行。

Google 2002年，用于Google Suggest，提出了AJAX概念，涉及到HTML、CSS、JavaScript、DOM、XML、JSON等技术，属于Web前端技术领域。

**AJAX技术的目标：**在客户端输入的同时，服务器也在运行；服务器的响应内容可以在客户端不需要刷新的情况下显示在页面中——无刷新的情况下实现页面内容的局部改变。

**AJAX技术的本质**：在用户浏览页面的同时，浏览器底层向服务器发出HTTP请求，并处理服务器返回的响应消息。

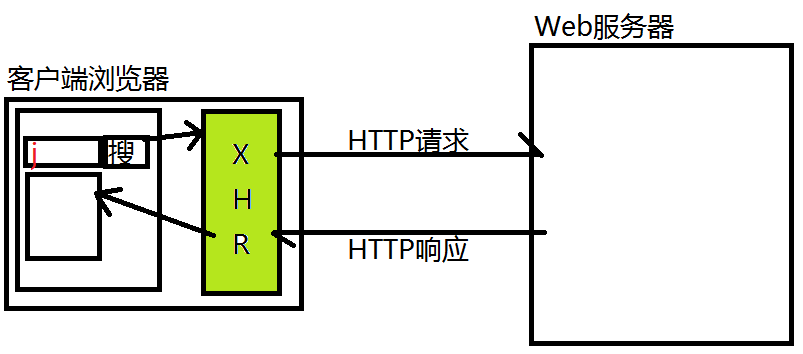
**AJAX的典型应用**：搜索建议、在线股票、在线聊天室、即时的数据验证、级联下拉列表

浏览器发起的HTTP请求可以分为两种类型：

**(1)同步HTTP请求**：地址栏输入URL、超链接跳转、表单提交、JS跳转

**(2)异步HTTP请求**：使用XHR对象发起

XMLHttpRequest：是浏览器底层提供的一个对象，专用于发起异步的HTTP请求。



5.XHR对象的成员

|  |
| --- |
| **XHR对象的成员属性：**  readyState:0 （只读）就绪状态 0-1-2-3-4  response:"" （只读）响应消息的主体内容  responseText:"" （只读）响应消息的主体内容(以文本呈现)  responseType:"" （只读）响应消息主体的内容类型  responseURL:"" （只读）响应消息来自于哪个URL  responseXML:null （只读）响应消息的主体内容（以XML呈现）  status:0 （只读）响应状态码  statusText:"" （只读）原因短句  timeout:0  UNSENT:0 请求尚未发送  OPENED:1 到服务器的连接已经打开  HEADERS\_RECEIVED:2 开始接收响应消息的起始行/头部  LOADING:3 开始加载响应消息的主体  DONE:4 响应消息接收完成 |
| XHR对象的成员事件：  onerror  onload  onloadend  onloadstart  onreadystatechange xhr.readyState属性值发生了改变  ontimeout |
| **XHR对象的成员方法：**  getAllResponseHeaders( ) 获取所有的响应消息头部  getResponseHeader( name ) 获取指定的响应消息头部  setRequestHeader( name, value ) 设置请求消息头部  open( method, url, isAsync ) 打开到服务器的连接  send( data ) 向服务器发送请求消息 |



演示：使用XHR对象发起异步的GET请求——注册时验证用户名是否存在

1. 创建check\_uname.php，接收用户名，返回消息，表明该用户名是否存在。
2. 创建register.html，当用户在用户名输入框中输入完毕后，立即后面显示“该用户名已被占用/可以使用”。

1603A 练习：使用XHR对象发起异步的GET请求——连续成绩录入

<h1>成绩录入系统</h1>

学生姓名： <input id="stuName">

语文成绩： <input id="chinese">

语文成绩： <input id="math">

<input type="button" value="保存成绩">

要求：点击“保存成绩”按钮，发起异步请求，把数据提交给服务器端的PHP页面，INSERT到MySQL数据库，返回'succ'或'err'提示文字。客户端弹出一个alert("保存成功/失败")。用户点击确定后，清空输入框内容，继续录入下一个学生成绩。

(1)创建tarena.sql，创建score表，包含sid、stuName、chinese、math三列

(2)创建score\_add.php，接收客户端数据，执行INSERT

(3)创建score\_add.html，显示输入框及按钮

# 6.12 原生AJAX XML

复习：

1. jQuery——重点&不难
2. AJAX——重点&难点
3. HTML5高级——掌握

AJAX相关知识点：(ajax 是纯服务器技术)

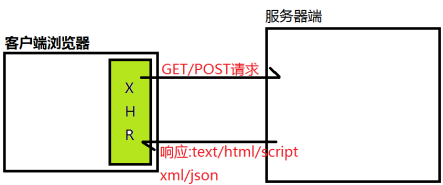
1. 数据库服务器
2. Web服务器
3. HTTP协议
4. 原生AJAX
5. jQuery封装的AJAX

1.复习：AJAX

2002年，Google在GoogleSuggest中提出的概念，包含HTML/CSS/JS/DOM/XML/HTTP等技术，实现“**页面内容无刷新的局部更新（动态页面静态化(如今晚tarena的作业，实现分离前端页面和后台页面)）”**。

请求的种类：

1. 同步请求：客户端请求后，必须等待服务器给响应
2. 异步请求：客户端请求后，仍然有可操作的页面内容



使用**XHR对象**的基本步骤：

(1)创建XHR对象

var xhr = new XMLHttpRequest( );

(2)监听XHR对象的状态改变事件

xhr.onreadystatechange = function(){

if( xhr.readyState===4){

if(xhr.status===200){ 完成且成功 }

else { 完成但失败 }

}

}

(3)打开到服务器的连接

xhr.open(method, uri, true)

(4)发起请求消息

xhr.send( null/data )

xhr.readyState: 0 1 2 3 4

xhr.status: 0 => 200

xhr.responseText: '' => '响应主体内容'

2.使用XHR发起GET请求

var xhr = new XMLHttpRequest();

xhr.onreadystatechange = function(){

if(xhr.readyState===4){

if(xhr.status===200){ //响应完成且成功

doResponse(xhr.responseText);

}else{ //响应完成但不成功

alert('响应完成但失败！'+xhr.status);

}

}

}

xhr.open('GET','xx.php?k1=v1&k2=v2',true);

xhr.send( null );

注意：GET请求没有请求主体，所以send(null)；若有请求数据，追加为URI后面的查询字符串。

3.使用XHR发起POST请求

var xhr = new XMLHttpRequest();

xhr.onreadystatechange = function(){

if(xhr.readyState===4){

if(xhr.status===200){ //响应完成且成功

doResponse(xhr.responseText);

}else{ //响应完成但不成功

alert('响应完成但失败！'+xhr.status);

}

}

}

xhr.open('POST', 'xxx.php', true); //URI无请求数据

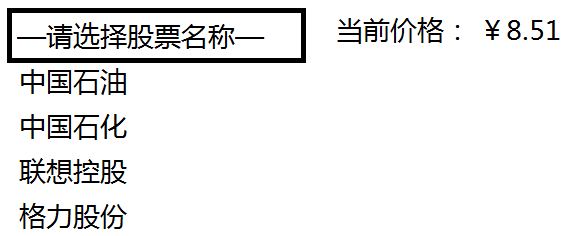
//3.5 设置请求主体的内容类型

xhr.setRequestHeader('Content-Type', 'application/x-www-form-urlencoded');

xhr.send( 'k1=v1&k2=v2' ); //提交请求主体，必须编码后提交

注意：POST请求不在URI后面追加查询字符串——请求数据放在请求主体中send()——请求主体发送前必须设置**Content-Type**请求头部；且请求主体中的中文、特殊标点必须使用**encodeURIComponent()**函数进行编码。

练习：客户端页面中的下拉列表中显示出所有的股票名称，后面的SPAN中显示出该股票当前的价格。



客户端在select.onchange事件中发起异步的GET请求，服务器需要返回一个数值字符串即可

4.使用XHR接收响应——text/plain

服务器端：

header('Content-Type: text/plain');

echo '文本';

客户端：

xhr.responseText //获得服务器端返回的文本响应（返回String类型值）

午间练习：仿写GoogleSuggest

1. 创建服务器端页面 接收客户端输入的关键字 返回包含此关键字的内容有哪些
2. 创建客户端页面

5.使用XHR接收响应——text/html——重点

服务器端：

header('Content-Type: text/html');

echo '<div>....</div>'; //向客户端输出的是HTML片段

客户端：

element.innerHTML = xhr.responseText; //String

练习：实现一个具有国际化功能的JS输出

根据客户端浏览器首选语言的不同，向客户端输出不同类型的欢迎消息

6.使用XHR接收响应——application/javascript

服务器端 ：

header('Content-Type: application/javascript');

echo 'alert("XXX")'; //向客户端输出一个字符串，内容是JS

客户端：

eval( xhr.responseText ) ; //客户端必须调用eval()执行服务器端返回String类型的js脚本

练习：添加新员工时的小功能点——在页面加载完成后，向服务器请求所有的部门列表

员工姓名：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

员工性别：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

所在部门：—请选择部门—

<option vlaue="101">研发部</option>

<option vlaue="102">市场部</option>

引出的问题：若服务器端需要返回多条有格式的数据，以何种格式合适？

**(1)方案1：text/html格式**

如<option value="">xxx</option>

不足：每个.php只能输出一种格式，只能供特定的父元素使用

**(2)方案2：text/plain格式** (plain:无格式)

如：tom@85@76#mary@77@99 #john@67@78

不足：不够灵活，适用性不广泛

**(3)方案3：application/javascript格式**

一般不用于描述数据。

**(4)方案4：使用XML或者JSON字符串格式**

7. 使用XHR接收响应——application(text)/xml (xml其实是一种文本)

XML：eXtensible Markup Language，可扩展的标记语言，所有的标签都可以自定义，但有开始就必须有结束，标签的属性必须用值，必须用单引号/引起来！

|  |
| --- |
| HTML：用于描述网页的内容，标签固定，语法比较松散  XML：与HTML无关，用于描述数据的一种字符串格式，标签任意，语法很严格 |

在Web应用中，XML格式的意义——在服务器和客户端之间传递批量的复杂数据：

**服务器端(PHP)： PHP数组 => XML字符串**

$data = [ [[],[],[]], [], []]; //复杂数据

//把PHP中的数组转化为符合XML标准的字符串

$string = "<xx>.....</xx>";

echo $string;

**客户端(JS)： XML字符串 => JS对象**

var xmlString = xhr.responseText;

var xmlDocument = xhr.responseXML; //DOM树

//遍历DOM树，取出其中的数据

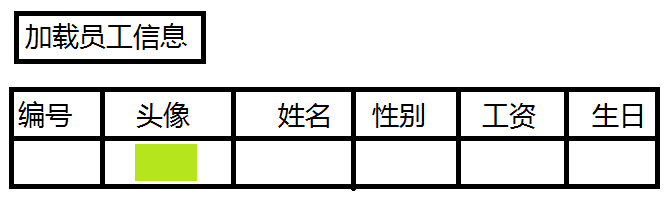
提示：xhr.responseXML是一个XML DOM对象——使用核心DOM操作即可遍历。

|  |
| --- |
| DOM分为三部分：   1. 核心DOM——可以操作HTMLDOM和XMLDOM 2. HTML DOM 3. XML DOM |

1603A 练习内容： 异步加载批量数据

1. 创建tarena.sql，其中创建员工表，插入五条员工记录；
2. 创建emp\_select.php，向客户端返回所有的员工信息，以XML字符串格式；
3. 创建emp\_select.html，点击按钮，向服务器发起异步请求，获取XML数据，并解析，把数据呈现在一个TABLE元素中。

(按钮禁用：this.disabled=true)

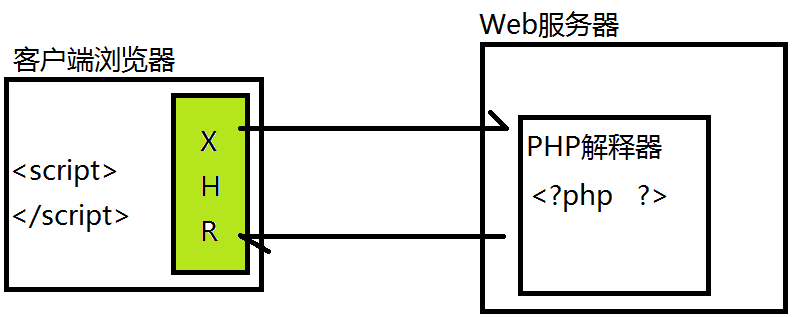


8. 使用XHR接收响应——application/json——最重点 (明天)

# 6.13 AJAX--JSON--JQUERY

复习：

AJAX的目标：实现页面无刷新的局部更新，动态页面静态化，分离前端页面和后台页面。



**使用XHR发起请求——GET**

var xhr = new XMLHttpRequest();

xhr.onreadystatechange = fn;

xhr.open('GET', 'x.php?k=v&k=v', true);

xhr.send(null);

**使用XHR发起请求——POST**

var xhr = new XMLHttpRequest();

xhr.onreadystatechange = fn;

xhr.open('POST', 'x.php', true);

xhr.setRequestHeader('Content-Type', 'application/x-www-form-urlencoded');

xhr.send('k=v&k=v');

**使用XHR接收响应——text/plain**

服务器端：

header('Content-Type: text/plain');

echo 'xzy';

客户端：

xhr.responseText; //string

**使用XHR接收响应——text/html**

服务器端：

header('Content-Type: text/html');

echo '<li>aa</li><li>bb</li>';

客户端：

e.innerHTML = xhr.responseText; //string

**使用XHR接收响应——application/javascript**

服务器端：

header('Content-Type: application/javascript');

echo 'alert("Hello")';

客户端：

eval( xhr.responseText ); //string

**使用XHR接收响应——application/xml**

服务器端：

header('Content-Type: application/xml');

echo '<xx></xx>';

客户端：

var xmlDocument = xhr.responseXML; //object

**使用XHR接收响应——application/json**

1.通用字符串数据格式——XML

2.通用字符串数据格式——JSON

JSON(JavaScript Object Notation) 是一种轻量级的数据交换格式。 易于人阅读和编写。同时也易于机器解析和生成。

最初该字符串格式来自于JS，后来被所有的编程语言所支持，成为了一种通用的字符串数据格式。

JSON字符串有两种格式：

(1)对象： '{"eid":101, "ename":"Tom", "isMarried":true }'

(2)数组： '[10,30,true,"Tom",{}, [ ] ]'

注意：

1) JSON字符串要么是**一个**{}，要么是**一个**[]

2) JSON字符串中所有的string的必需使用""，包括key

3) JSON对象中的value只能是string、number、object、array、

true、false、null，不存在Date等类型

JS语言中：

JSON字符串 => JS对象/数组：  **var obj =** **JSON.parse( str )**

JS对象/数组 => JSON字符串： **var str =** **JSON.stringify( obj )**

PHP语言中：

PHP数组 => JSON字符串： **$str = json\_encode( $arr );**

JSON字符串 => PHP数组： **$arr = json\_decode( $str );**

**PHP数组 => JSON字符串 => echo => xhr.responseText => JSON字符串 => JS数组**

|  |
| --- |
| 两种通用的字符串数据描述格式：XML vs JSON   1. 二者都是字符串格式，可以用于描述字符串数据，在不同的系统间进行传递 2. 历史上XML格式先出现，JSON后出现 3. XML属于重量级格式；JSON属于轻量级格式 4. XML处理比较复杂；JSON的处理比较简单 5. XML还有一些其他的应用领域，JSON一般只用于数据传递领域 |

练习：HTML页面加载完成后，异步请求服务器端的动态数据



AJAX阶段知识点：

1. 服务器： 数据库服务器 + Web服务器
2. HTTP协议： 请求消息、响应消息
3. 原生AJAX： 四步
4. jQuery中封装的AJAX函数

3.jQuery对AJAX操作的封装函数

load(url,[data],[callback]) 加载HTML到当前元素的innerHTML

$.get(url,[data],[fn],[type]) 发起GET请求

$.post(url,[data],[fn],[type]) 发起POST请求

$.getScript(url,[callback]) 发起GET请求，调用eval()执行返回的script

$.getJSON(url,[data],[fn]) 发起GET请求，调用JSON.parse()解析JSON

$.ajax(url,[settings]) 万能AJAX请求函数，上述五个函数底层都调用的是$.ajax——上述五个是此函数的简化版本

一个jQuery对象函数 + 五个 jQuery全局函数

4. jQuery对AJAX操作的封装函数——load

使用方法：

**$('选择器').load( url );**

作用：发起异步的XHR GET/POST请求，把指定URL返回的HTML内容设置为当前选定元素的innerHTML属性。

注意：该方法要求URL返回内容必须是text/html类型的片段；返回的响应主体会覆盖当前选定元素的内容。

演示：把整个网站的头和尾写在独立的文件中，其它页面加载完成后，异步请求页头和页尾内容。—— “页面包含”技术

5. jQuery对AJAX操作的封装函数——$.get

使用方法：

**$.get( url, [data], fn )**

作用：

向服务器指定的URL发起异步的GET请求，获取到成功的响应消息后，自动调用指定的回调函数，$.get('check\_uname.php', 'uname=tom', function(txt, msg, xhr){ //成功响应的回调函数 })

演示：用户输入完想注册的用户名后，立即在后侧显示出该用户名是否可用。

6.jQuery对AJAX操作的封装函数——$.post

使用方法：

**$.post( url, [data], fn )**

作用：

向服务器指定的URL发起异步的POST请求，获取到成功的响应消息后，自动调用指定的回调函数。

注意：该方法的回调函数中，会根据响应消息的Content-Type头部的不同，自动调用JSON.parse()方法、或eval( )

演示：不间断的成绩录入系统

7. jQuery对AJAX操作的封装函数——$.getScript

使用方法：

$.getScript( url, [data], [fn])

说明：

向指定的url发起异步的GET请求，获取服务器端返回的application/javascript类型的响应消息，自动调用eval()函数对响应结果字符串进行执行。

演示：根据客户端浏览器首选语言的不同，向客户端输出不同形式的欢迎消息。

8. jQuery对AJAX操作的封装函数——$.getJSON

使用方法：

$.getJSON( url, [data], [fn])

说明：

向指定的url发起异步的GET请求，获取服务器端返回的application/json类型的响应消息，自动调用JSON.parse()函数把服务器返回的JSON字符串解析为JS对象。

注意：此方法会自动调用JSON.parse()方法解析服务器返回的JSON字符串。

9. jQuery对AJAX操作的封装函数——$.ajax

使用方法：

$.ajax( {

type: 'GET', //POST/PUT/DELETE/...

url: 'xx.php',

data: 'k=v&k=v', //还可以形如{ k:v, k:v }

dataType: 'text', //html/script/json/xml/jsonp

beforeSend: fn, //请求发送之前的回调函数

success: fn, //响应成功后的回调函数

error: fn, //响应失败后的回调函数

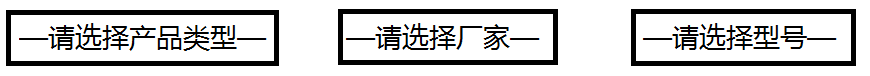
complete: fn //响应完成后的回调函数

} );

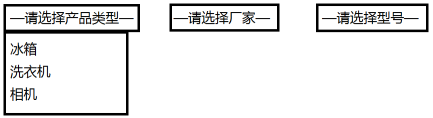
注意：$.ajax的形参都是可以省略的

1603A 练习内容： jQuery对AJAX操作的封装

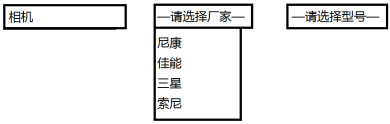
(1)级联下拉列表——提示：可以编写三个PHP页面



待页面加载完后，异步请求产品类型有哪些



待用户选择某个“产品类型”后，异步请求该类型对应的厂家



待用户选择某个“厂家”后，异步请求该厂家对应的产品型号

(2)仿QQ空间，到页面底部点击“加载更多”就可以异步请求出更多的好友消息



# 6.14 JSONP(重点) 京东1

复习：

AJAX：

服务器端

HTTP协议

原生AJAX——四步

jQuery中AJAX的封装：

(1)load() $('div').load(url)

(2)$.get() $.get(url,data,fn)

(3)$.post() $.post(url,data,fn)

(4)$.getScript() $.getScript(url)

(5)$.getJSON() $.getJSON(url,data,fn)

(6)$.ajax({

type: 'GET',

url: 'xx.php',

data: {uname:'tom',age:20},

dataType: 'text',//html/script/xml/json

beforeSend: fn,

success: fn,

error: fn,

complete: fn

})

//1 var xhr = new XMLHttpRequest();

//2 xhr.onreadystatechange = function(){

if(xhr.readyState===4){

if(xhr.status===200){

success();

}else{

error();

}

complete();

}

}

//3 xhr.open();

beforeSend();

//4 xhr.send(null/data);

1.面试题：跨域请求和JSONP——抽象&脑筋急转弯

跨域请求：Cross Domain

当前页面： <http://www.tmooc.cn/1.html> 请求下列资源：

http://www.tarena.cn/list.php



**跨域请求：**请求的下一个资源所在的 协议、域名、端口号 三者之一与当前资源不一致就称为跨域请求。

**浏览器允许跨域请求吗？**

**允许：**<img src="跨域的图片">

**允许：**<link rel="stylesheet" href="跨域的CSS">

**允许：**<a src="跨域的链接">

**允许：**<script src="跨域的JS">

**禁止：**AJAX请求是不允许跨域的！

说明：处于安全考虑，**所有的浏览器默认都禁止使用XHR异步的跨域请求**。

面试题：我们公司有两个域名：如tmooc.cn，和 tarena.cn，如何使用AJAX在tmooc.cn/1.html 请求 tarena.cn/2.php？ —— 这是XHR的跨域请求！浏览器是禁止的！！

解决方案：

方案1：设置“Access-Control-Allow-Origin”头部——百度

方案2：使用JSONP手段，可以巧妙的绕过XHR的跨域限制

方案3：....——百度

JSON：是一种字符串数据格式

JSONP：JSON with Padding，填充式JSON，是一种手段，可以绕过XHR的跨域限制

**JSONP手段的原理：**

在客户端动态的创建一个script标签，代替XHR对象，指定script.src向服务器发起请求，请求的PHP页面除了返回需要的数据外，还需要填充一个指定的函数名。

header('Content-Type: application/javascript');

$json = json\_encode($data);

echo $cb . '(' . $json . ')';

2.jQuery中发起JSONP请求

jQuery中的$.getJSON()和$.ajax()可以使用XHR发起同域的请求，也可以使用SCRIPT标签发起跨域请求。

**同域下的请求：**

$.getJSON('xx.php', fn)

$.ajax({ url: 'xx.php', success: fn})

**跨域请求——JSONP：**

$.getJSON('http://其它域名/xx.php?**callback=?**', fn)

$.ajax({

url: ' http://其它域名/xx.php',

dataType: **'jsonp'**,

success: fn

});

3.AJAX阶段项目 —— 京东“用户中心”

实现顺序：

1. 创建页头和页尾 页面 —— 不能是完整的HTML
2. 创建主页面，加载完成后异步加载页头和页尾 $('#header').load();
3. 弹出登录DIV，进行登录验证 $.post( )
4. 主页中实现三个功能的切换：我的订单、消费统计、幸运抽奖
5. 实现“我的订单” —— TABLE 异步加载当前登录用户的所有订单 $.getJSON('myorder.php')
6. （选作）分页显示“我的订单”

4.模态登录框的编写

<div class="modal"> <!--半透明的背景层-->

<div class="modal-dialog"> <!--定位和宽高-->

<div class="modal-content"> <!--背景和边框-->

模态框内容

</div>

</div>

</div>

------------------------------------

.modal {

position: fixed;

left: 0;

right: 0;

top: 0;

bottom: 0;

background: rgba(0,0,0, .6);

/\*opacity: .5; 会导致所有内容全部半透明\*/

}

.modal-dialog { /\*负责宽和定位\*/

width: 300px;

margin: 100px auto ;

}

.modal-content { /\*负责背景色、边框、阴影\*/

background: #fff;

min-height: 100px;

}

5.jQuery中的表单提交——表单序列化

**var result = $('#login-form').serialize( );**

**表单序列化**：把表单中所有的输入域的name值和用户的输入值拼接为k=v的形式，最终把所有的k=v使用&符号拼接起来，最后的结果形如：

**uname= %E5%95%8A&upwd=123&sex=M&city=BJ**

序列化后的字符串可以直接用于HTTP请求消息。

6.盒子模型尺寸的计算方法

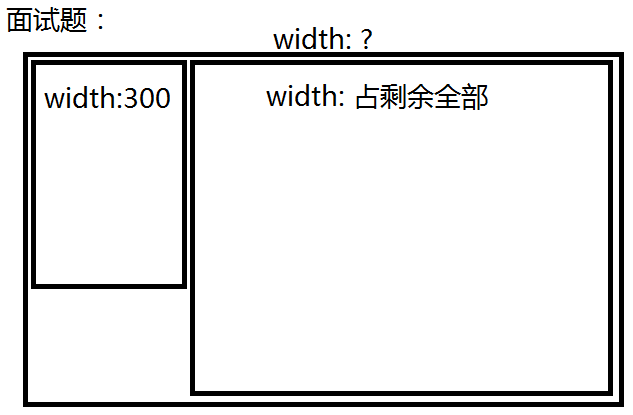
**盒子尺寸计算方法：**

box-sizing: content-box;  **//内容盒子**

总宽度 = width + padding + border + margin

box-sizing: border-box; **//边框盒子**

总宽度 = width + margin //padding和border算在width内



# 6.15 京东数据库(超重点) 面试

复习：

jQuery对AJAX操作的封装

1. $('div').load('x.php')
2. $.get('x.php',{k:v},function(response){})
3. $.post('x.php',{k:v},function(response){})
4. $.getScript('x.php')
5. $.getJSON('x.php',{k:v}, function(obj){})
6. $.ajax({

type: 'GET',

url: 'x.php',

data: {k:v},

beforeSend: fn,

success: fn,

error: fn,

complete: fn

})

跨域请求：

允许跨域：img、link、a、script、iframe

禁止跨域：xhr.open('http://tmooc.cn/x.php')

如何绕过xhr的跨域限制——JSONP：

function donResponse(obj){ }

var s = document.createElement('script');

s.src = 'http://tmooc.cn/x.php?callback=doResponse';

document.body.appendChild(s);

**表单序列化**：把表单中所有的输入域和值拼接为形如“k1=v1&k2=v2”的字符串，用于HTTP请求消息。

var kv = $('#login-form').serialize( ) ;

1.AJAX阶段项目——京东用户中心

(1)把.sql文件提交给MySQL服务器执行

(2)创建header.php和footer.php——只是DIV片段

(3)创建usercenter.html

...<body>

<div id="header"></div>

<div id="main">主体内容</div>

<div id="footer"></div>

</body>...

(4)待页面加载完成，异步请求页头和页尾

(5)使用遮罩层实现异步用户登录，成功后去除遮罩层

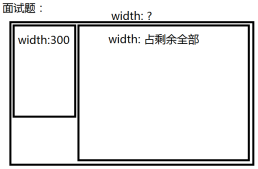
(6)实现main的布局结构

(7)实现主体左侧附加导航的切换

(8)实现“我的订单”的TABLE中的静态内容

(9)(选作)分页显示

**2.“左固定宽，右占剩余”布局方法(！面试 ! )**



.left {

width: 200px;

float: left;

}

.right {

margin-left: 200px;

}

3.补充：复杂的数据库表间关系——难点&非重点

订单表：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| oid | uname | submit\_time | price | payment | shop |  |
| 1 | 强东 | 2015-10-25 | 100 | 在线支付 | 自营 |  |
| 2 | 奶茶 | 2015-10-25 | 50 | 在线支付 | 自营 |  |

商品表：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| pid | pname | price | img |  |
| 101 | 可口可乐600 | 3.5 | 1.jpg |  |
| 102 | 百事可乐600 | 3.2 | 2.jpg |  |

订单详情表：

|  |  |  |  |
| --- | --- | --- | --- |
| 记录编号 | 订单编号 | 产品编号 | 购买数量 |
| 1 | 1 | 101 | 2 |
| 2 | 1 | 102 | 3 |
| 3 | 2 | 102 | 10 |

**订单JSON信息形如：**

[ //订单数组

{ //一个订单

order\_num: 234234234,

user\_name: 强东,

shop\_name: 京东自营,

price: 123,

productList: [ //当前订单中的产品数组

{product\_id: 101, pname:'可口可乐', price:3.5, img:1.jpg},//产品对象

{product\_id: 102, pname:'百事可乐', price:3.2, img:2.jpg},

]

},

{ ... }

]

**SQL问题1：如何查询2/3/5号的商品信息？**

SELECT \* FROM 商品表 WHERE 商品编号=2 OR 商品编号=3 OR 商品编号=5；

SELECT \* FROM 商品表 WHERE 商品编号 IN ( 2, 3, 5 )；

**SQL问题2：如何查询4号订单所购买的产品编号？**

SELECT 产品编号 FROM 订单详情表 WHERE 订单编号=4;

**SQL问题3：如何查询4号订单所购买的产品信息？**

SELECT \* FROM 商品表 WHERE 商品编号 IN (

SELECT 产品编号 FROM 订单详情表 WHERE 订单编号=4

)；

4.页面中的数据分页显示——难点&重点



前端和后台在传输分页数据时，一般使用如下格式的JSON字符串：

**{**

**"record\_count": 22, //总记录数**

**"page\_size": 5, //页面大小，即每页的记录数——固定的值**

**"page\_count": 5, //总页数**

**"cur\_page": 5, //当前显示的数据属于哪一页**

**"data": [ {},{},{},{},{} ] //当前页中的数据**

**}**

**PHP中如何获取总记录数：**

SELECT **COUNT(order\_id)** FROM jd\_orders WHERE ....;

**PHP中如何计算出总页数：**

**ceil( record\_count / page\_size )**

**PHP中如何得到当前页号：**

cur\_page = **$\_REQUEST['pno']**

**MySQL中如何实现分页查询：**

SELECT \* FROM jd\_orders WHERE ... **LIMIT $start, $count**;

说明： $start 指定从哪一行记录开始读取，第一行为0

$count 指定最多读取几行记录

第1页 0/1/2/3/4 LIMIT 0, 5

第2页 5/6/7/8/9 LIMIT 5, 5

第3页 10/11/12/13/14 LIMIT 10,5

第4页 15/16/17/18/19 LIMIT 15,5

第5页 20/21 LIMIT 20,5

公式： **LIMIT (cur\_page-1)\*page\_size, page\_size**

|  |
| --- |
| SQL语句中的分组计算函数：  SELECT MAX(price) FROM orders WHERE ...;  SELECT MIN(price) FROM orders WHERE ...;  SELECT AVG(price) FROM orders WHERE ...; //求平均值  SELECT SUM(price) FROM orders WHERE ...; //求总和  SELECT COUNT(price) FROM orders WHERE ...; //求数量 |

# HTML5

# 6.16 HTML5 (表单)input

复习：

分页查询：

前后端交互的数据格式：

{

"record\_count": 123,

"page\_size": 10,

"page\_count": 13,

"cur\_page": 1,

"data": [{},{},{}...{}]

}

前端页面：

<ul class="pager">

</ul>

function loadDataByPage ( pno ){

$.ajax( {

url: 'data/data\_bypage.php',

data: {pno: pno},

success: function(pager){

//1把数据显示在DOM树

//循环遍历 pager.data 数组

//2输出一个分页条

$('#pager').empty();

$('#pager').append('<li><a href="">当前页-2</a></li>');

$('#pager').append('<li><a href="">当前页-1</a></li>');

$('#pager').append('<li class="active">当前页</li>');

$('#pager').append('<li><a href="">当前页+1</a></li>');

$('#pager').append('<li><a href="">当前页+2</a></li>');

}

} );

}

1 2 3

1 2 3 4

1 2 3 4 5

3 4 5 6 7

10 11 12 13

11 12 13

SELECT \* FROM 13亿人口表;

第三阶段知识点汇总：

1. jQuery

概述 查找 操作 遍历 添加 删除 替换 克隆 事件 动画

1. AJAX

数据库服务器 Web服务器 PHP HTTP 原生AJAX jQuery-AJAX

1. HTML5新技术

1)表单新特性——再再次重要

2)视频和音频——再次重要

3)Canvas绘图——重要

4)SVG绘图——次重要

5)地理定位

6)拖放API

7)WebWorker

8)WebStorage——次重要

9)WebSocket

1.HTML5新特性：表单新特性——新的input type

**<input type="?">**

已经学过的type： text、password、radio、checkbox、hidden、file、image、button、submit、reset

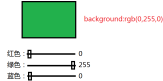
HTML5新添type：email、search、url、tel、number、range、color、date、week、month

提示：

1. HTML5新添的input type存在严重的兼容性问题！
2. 新特性弹出的提示消息本质是“窗口”，不是HTML元素，无法使用JS、CSS进行控制！

1. email: 要求输入必需符合邮箱格式 xx@xx
2. url: 要求输入必需符合URL格式 xx:yy
3. tel: 无输入格式验证，在手机中弹出的数字键盘，而不是字母键盘
4. number: 要求输入一个整型数字，可以配合max/min/step新属性使用
5. search: Chrome中会显示一个x号
6. range: 范围选择器、滑块，可以配合max/min/step新属性使用

练习：使用滑块组件实现一个调色板：



1. color：颜色选择器，Chrome/FF偷懒了，都直接使用操作系统自带的颜色选择器
2. date：日期选择器，FF不支持
3. month：月份选择器，FF不支持
4. week：星期选择器，FF不支持

2.HTML5新特性：表单新特性——新的表单元素

**已经学过的表单输入元素：**

input、textarea、select/option

**HTML5新增的表单元素：**

1. **<datalist></datalist>**  数据列表，本身是不可见元素，为某个input提供可选输入的列表，使用方法：

<datalist id="LISTID"><option>选项名</option></datalist>

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

1. **<progress></progress>** 进度条，仅用于进度的提示，不能被用户操作，也不会被提交。

<progress value="0.5"></progress>

练习：使用定时器，修改progress的value属性

1. **<meter></meter>** 刻度尺，仅用于水平/数量的提示，不能被用户操作，也不会被提交。需要指定如下属性：

<meter min="最小值" max="最大值" low="下限值" high="上限值" optimum="最优值" value="当前实际值"></meter>

如果最优值小于下限值，刻度尺颜色分配：

min (绿色) low (黄色) high (红色) max

如果最优值大于下限值且小于上限值，刻度尺颜色分配：

min (黄色) low (绿色) high (黄色) max

如果最优值大于上限值，刻度尺颜色分配：

min (红色) low (黄色) high (绿色) max

1. **<output></output>** 语义标签，表示一个计算的输出值，仅用于计算结果的提示，不能被用户操作，也不会被提交。

提示：上述四个新元素存在兼容性问题。

3.HTML5新特性：表单新特性——input的新属性

**已经学过的input的属性：**

<input id="" class="" style="" title="" name="" value="" type="" readonly checked selected diabled >

**HTML5新增的input属性：**

<input type="number" max="最大值" min="最小值" step="步长">

<input type="text" list="datalist元素的ID" autocomplete="on/off">

<meter min="最小值" max="最大值" low="下限值" high="上限值" optimum="最优值" value="当前实际值"></meter>

1. placeholder：占位符，用于在input中显示提示性文字，注意：不是value
2. autocomplete：可赋值为on或off，是否记录之前提交的内容，以实现下次输入时的自动提示
3. autofocus：自动获得输入焦点
4. multiple：用于email/url/tel等，允许使用逗号指定多个输入
5. form：为input指定form属性，可以将input从form内部提到外部，且不影响输入域的提交

//练习：把submit按钮放到表单外部会如何？

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

与表单输入验证有关的属性：

1. required：必填属性，若用户未填写内容，则无法提交
2. minlength：限制最少的字符个数，非标准，FF(firefox)不支持 （chrome自定的）
3. maxlength：限制最大的字符个数
4. max：一般用于number、range、<meter>、<progress>，表示允许的数字最大值
5. min：一般用于number、range、<meter>、<progress>，表示允许的数字最小值
6. step：一般用于number、range，表示步长
7. pattern：使用正则表达式进行验证

<input type="text" name="cellphone" placeholder="请输入手机号" **pattern="^1[358]\d{9}$"**  **required** >

4.如何修改表单的提示消息——难点&了解

HTML5表单可能在如下情形下弹出提示消息：

1. 特定的input type不被满足，如email、url
2. 特定的验证属性不被满足，如required、pattern、step、minlength

不同的浏览器会弹出不同的提示消息。如何自定义弹出的提示消息：

HTML5为所有的表单输入域都添加了一个validity对象属性，该对象中包含如下的属性：

1. badInput:false 无效的输入，如
2. customError:false 是否存在自定义错误
3. patternMismatch:false 正则表达式不匹配，违反pattern验证
4. rangeOverflow:false 数字范围过大，违反了max验证
5. rangeUnderflow:false 数字范围过小，违反了min验证
6. stepMismatch:false 步长不匹配，违反step验证
7. tooLong:false 字符个数太长，违反maxlength验证
8. tooShort:false 字符个数太短，违反minlength验证
9. typeMismatch:false 类型不匹配，如email/url格式错误
10. valid:true **当前输入值是否有效**
11. valueMissing:false 缺少值，违反required验证

上述属性会随着用户的输入即时的发生改变，只要有一项验证失败，对应的boolean值就会变成true，导致valid属性变为false；只有当所有其他属性都变为false，valid属性才会变为true，该输入域是有效的，可以被提交了。

通过检查上述属性是否为true，可以通过“设置自定义的有效性消息setCustomValidity”设置自定义的错误消息——validity.customError就会变为true，自定义错误消息的优先级高于浏览器默认的消息优先级。

1603A，练习内容：自定义HTML5表单验证提示消息

创建一个注册新用户表单，要求用户输入如下内容，注意需要的验证，每个验证的失败消息必须自定义。

用户名：必填

密码：必填，长度不能小于6，不能大于9

邮箱：必须符合email格式

微博：必须符合url格式

年龄：数值不能小于18，不能大于60

手机号码：必须符合手机号码格式

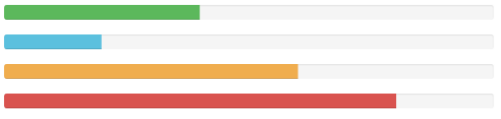
**Custom:自定义**

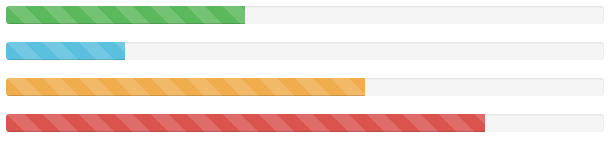
|  |
| --- |
| btSubmit.onclick = function(){ //提交按钮被点击时，设置自定义验证消息  if(userName.validity.valueMissing===true){  //设置自定义错误消息——会导致validity.customError变为true  userName.setCustomValidity('用户名不能为空');  }else {  //取消自定义错误消息——会导致validity.customError变为false  userName.setCustomValidity('');  }  } |

午间练习：









夜间练习：

(1)使用DIV实现一个圆形进度条



# 6.17 MP4 / 3 Canvas(重点) 绘图

复习：

HTML5新特性：

1. 表单新特性
2. 视频和音频
3. **Canvas绘图——重点**
4. **SVG绘图**
5. 地理定位
6. 拖放API
7. WebWorker
8. WebStorage
9. WebSocket

表单新特性：

1. 新的input **type**

<input type="?">

email、search、url、tel、number、range、color、date、month、week

1. 新的**表单元素**

<input><textarea><select>

<datalist><progress><meter><output>

1. 表单元素的**新属性**

<input ?>

placeholder、autocomplete、autofocus、multiple、form

-------------------------

required、minlength、maxlength、max、min、step、pattern

自定义输入域的提示消息

if( input.validity.xxx === true ){

input.setCustomValidity("自定义消息");

}else {

input.setCustomValidity("");

}

1.HTML5新特性——视频

HTML5之前在页面播放需要使用flash播放器；HTML5添加两个标签：

<video></video>

<audio></audio>

这两个标签有一些特殊属性、事件、方法。

<video>标签可用的属性：

1. src：待播放的资源的URL
2. width：播放区域的宽度
3. height：播放区域的高度
4. controls：是否显示播放控件，默认不显示
5. autoplay：是否自动播放，默认不自动播放
6. loop：是否循环播放，默认不循环
7. preload：如何预加载视频，可以赋值为

“none”：不预加载

“metadata”：只预加载元数据，如影片宽、高、大小、时长

“auto”：（默认值）自动预加载元数据以及部分影片数据

1. poster：影片播放之前显示的“电影海报”

元数据(Meta data)：描述数据的数据，称为元数据

元数据： 员工姓名 年龄 是否党员

数据： Tom 20 true

--------------------------------------------------------

元数据： charset="UTF-8"

数据： <p>这是一个新闻内容...</p>

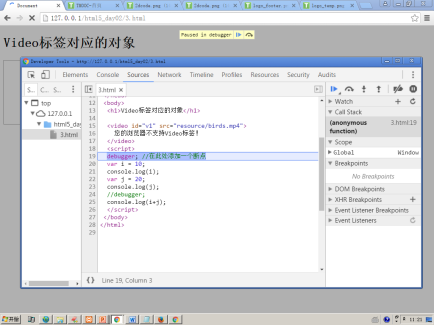
--------------------------------------------------------

元数据： width="856" height="480"

数据：

2.Video对象的成员

提示：学会使用Chrome中的断点调试



1. autoplay:false
2. controls:false
3. currentTime:0 当前播放到哪一秒
4. duration:NaN 总时长
5. ended:false
6. loop:false 是否循环
7. muted:false 是否静音
8. oncanplay:null 当能够播放时触发
9. oncanplaythrough:null 当能够顺利播放一段时间
10. onended:null 当播放的结尾时触发
11. onerror:null 当播放错误时触发
12. onloadeddata:null 当加载了影片数据时触发
13. onloadedmetadata:null 当加载了影片元数据时触发
14. onpause:null 当影片暂停时触发
15. onplay:null 当影片开始播放时触发
16. paused:true 当前是否处于暂停状态
17. poster:"" 电影海报
18. preload:"auto" 如何预加载
19. src:"http://127.0.0.1/html5\_day02/resource/birds.mp4"
20. videoHeight:0
21. videoWidth:0
22. volume:1
23. play(): function(){} 使用JS控制影片播放
24. pause(): function(){} 使用JS控制影片暂停

演示：自定义播放暂停和开始按钮

3.HTML5新特性——音频播放

HTML5之前，播放音频可以使用flash播放器，或者使用<bgsound>（IE专有，只能网页的背景音乐，不能使用JS控制播放、音量）。

HTML5中，使用<audio>标签来播放音频。

该标签的常用属性与<video>几乎一样，如：

1. src
2. autoplay：是否自动播放，默认为false
3. controls：是否显示播放控制条，默认为false
4. loop：是否循环播放，默认为false

练习：页面中看不见音频播放控制条，但存在一个复选框“播放背景音乐”，勾选则播放，取消勾选则暂停。

提示：可能使用audio.paused属性，还可能使用audio.play()、audio.pause()等方法。

**4.HTML5新特性——Canvas(画布) 绘图技术——重点&小难点**

提示：Canvas绘图的难点在**方法**和**属性**的记忆上！

Canvas技术用于在网页上实现绘图，主要应用领域：

1. 绘制各种统计图表，柱状图、饼图、曲线图、分布图...
2. 动画和游戏

使用Canvas的方法：

<canvas width="500" height="400">

您的浏览器不支持Canvas标签！

</canvas>

提示：

(1)Canvas的宽和高只能用属性方式声明！若使用样式来声明则无效。

(2)Canvas的本意是“画布/画板”，可以盛放绘制的内容；真正执行绘图任务的是“画笔/绘图上下文对象”——Context

(3)每一个画布，有且只有一个画笔对象：

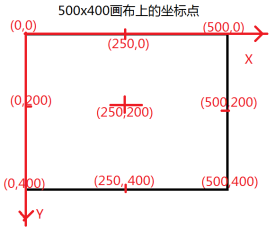
var ctx = canvas.getContext( '2d' );

(4)画笔/绘图上下文对象的成员——小难点： （fill 充满/填充，rect矩形 ， gradient 倾斜度..）

1. fillStyle:"#000000" 填充颜色
2. **font:"10px sans-serif" 字体**
3. globalAlpha:1 全局透明度
4. lineCap:"butt" 线的端点样式
5. lineJoin:"miter" 线的连接处样式
6. **lineWidth:1**   **线条的宽度**
7. shadowBlur:0 阴影部分模糊距离
8. shadowColor:"rgba(0, 0, 0, 0)" 阴影部分颜色
9. shadowOffsetX:0 阴影水平偏移量
10. shadowOffsetY:0 阴影竖直偏移量
11. **strokeStyle:"#000000" 轮廓/描边颜色**

---------------------------------------------------

1. arc:*arc() 绘制一个弧线*
2. beginPath:*beginPath() 开始绘制路径*
3. clearRect:*clearRect() 清除一个矩形范围*
4. clip:*clip() 裁切*
5. closePath:*closePath() 闭合一条路径*
6. createLinearGradient:*createLinearGradient() 创建一个线性渐变色*
7. createRadialGradient:*createRadialGradient() 创建一个径向渐变色*
8. drawImage:*drawImage() 绘制一幅图像*
9. ellipse:*ellipse() 绘制一个椭圆*
10. fill:*fill() 填充一条路径*
11. **fillRect:*fillRect() 填充一个矩形区域***
12. **fillText:*fillText() 填充一段文本***
13. lineTo:*lineTo() 绘制一条直线*
14. measureText:*measureText() 测量一段文本，得到宽度*
15. moveTo:*moveTo() 移动画笔到指定点*
16. rect:*rect() 绘制一个矩形路径*
17. rotate:*rotate() 旋转*
18. scale:*scale() 缩放*
19. stroke:*stroke() 对一条路径描边*
20. strokeRect:*strokeRect() 对一个矩形进行描边*
21. strokeText:*strokeText() 对一段文本进行描边*
22. translate:*translate() 进行位移*



5.使用画笔在画布上绘制图形

**(1)绘制矩形(长方形)——矩形以自己的左上角作定位点**

**ctx.lineWidth = 10; 修改描边的宽度**

**ctx.strokeStyle = '颜色'/渐变色; 修改描边的颜色**

**ctx.strokeRect(x, y, w, h) 描边一个矩形**

**ctx.fillStyle = '颜色'/渐变色对象; 填充样式**

**ctx.fillRect(x, y, w, h) 填充一个矩形**

**ctx.clearRect(x,y, w, h); 清除一个矩形范围内所有内容**

**(2)绘制文本——文字以自己的坐下角作定位点**

ctx.font = '20px SimHei'; 设置字体大小和样式

**var w = ctx.measureText(txt).width; 获取一段文本的宽度**

ctx.strokeStyle = "颜色"/渐变;

ctx.strokeText(txt, x, y) 对文字进行描边

ctx.fillStyle = "颜色"/渐变;

ctx.fillText(txt, x, y) 对文字进行填充

(3)绘制路径

(4)绘制图像

(1)绘制一个矩形，颜色在随机的改变，在画布中来回的碰撞移动

(2)使用从服务器端获取的如下的JSON数据：

[

{name:'六月', value: 3500},

{name:'七月', value: 5500},

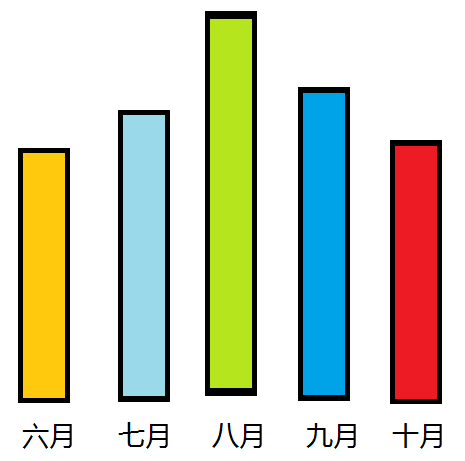
{name:'八月', value: 8500},

{name:'九月', value: 6500},

{name:'十月', value: 4000}

]

绘制出类似下面的统计柱状图：



**css:**

**pointer-events: none; //滚动的时候设置当前元素不可被点击**

例子：<http://evanyou.me/> canvas元素中的样式用到以上样式

# 6.20 绘圆弧 绘img

复习：

HTML5新特性：

1. 表单新特性
2. 视频和音频
3. Canvas绘图
4. SVG绘图

Canvas绘图技术使用步骤：

<canvas id="c1" width="500" height="400">

您的浏览器不支持Canvas标签！不能使用样式设置画布的宽和高！

</canvas>

<script>

var ctx = document.getElementById('c1').getContext('2d');

//var ctx = $('#c1')[0].getContext('2d');

</script>

1.使用画笔(绘图上下文对象)绘制各种图形和图像

提示：Canvas绘图的难点： (1)坐标的确定 (2)函数名的记忆

**(1)绘制矩形——矩形的左上角为定位点 （stroke：笔画）**

ctx.lineWidth=20;

ctx.strokeStyle='颜色'/渐变对象;

ctx.strokeRect(x,y, w,h);

ctx.fillStyle='颜色'/渐变对象;

ctx.fillRect(x,y, w,h);

ctx.clearRect(x, y, w, h);

**(2)绘制文本——文本的定位在左下角**

ctx.font='30px SimHei';

var txtWidth = ctx.measureText(txt).width;

ctx.strokeText(txt, x, y);

ctx.fillText(txt, x, y);

**(3)绘制路径**

ctx.beginPath( ); //开始一条新路径

...在路径上添加定位点...

ctx.closePath( ); //闭合当前路径



常见的添加定位点的方法：

ctx.moveTo(x,y); //把画笔移动到指定点

ctx.lineTo(x,y); //从上一个点开始到当前点绘制一条直线

ctx.arc(x,y,r,sAngle,eAngle); //绘制一条圆弧线

( 默认从3点位置开始: 0\*Math.PI/180 )

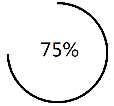
ctx.ellipse(x,y,rx,ry,sAngle,eAngle); //绘制一条椭圆弧线

路径的两个用途： (路径必须描绘)

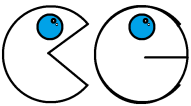
1. ctx.stroke(); //描边刚刚绘制的路径
2. ctx.fill(); //填充刚刚绘制的路径

练习：

1)绘制一个圆形进度条



2)绘制一个张嘴闭嘴的吃豆人



**(4)绘制图像——以图像的左上角为定位点**

ctx.rotate( angle ); //旋转画笔，此后绘制的图像/图像都会旋转——旋转围绕着画布的坐标原点为轴

ctx.translate(x, y); //平移坐标轴的原点

ctx.drawImage(img, x, y); //原宽原高的绘制图像

ctx.drawImage(img, x, y, width, height); //使用指定的宽和高绘制图像

提示：

1)创建图片： var img =new Image(); img.src="xx.jpg";

2)使用图片必须等待加载完成！ img.onload = function(){ ... }

练习：

1)绘制一个小飞机，从左飞到右，再从右飞到左...

2)绘制一个在原地打转的飞机

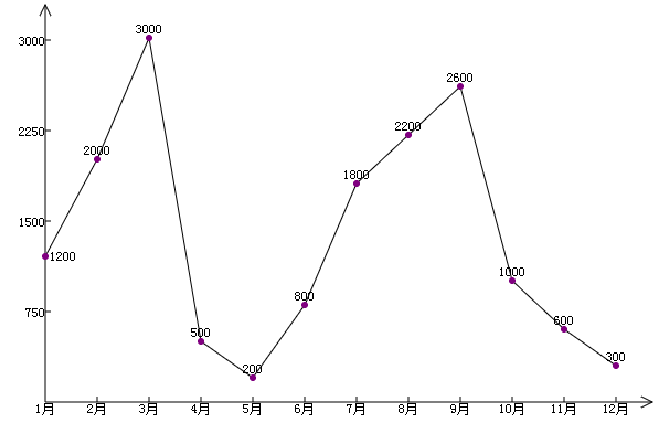
午间练习：



var data = 'ABCDEFGHJKMNPQRSTWXY3456789';

var c1 = data[随机下标];

1603A，练习内容： “京东用户中心”项目“消费统计”和“幸运抽奖”功能点

# 6.21 jd--图表框架--SVG

复习：

Canvas绘图方法：

1. 绘制矩形

ctx.strokeStyle = '';

ctx.fillStyle = '';

ctx.lineWidth = 20;

ctx.strokeRect(x,y,w,h)

ctx.fillRect(x,y,w,h)

ctx.clearRect(x,y,w,h)

1. 绘制文本

ctx.measureText(txt).width

ctx.strokeText(txt, x, y)

ctx.fillText(txt,x,y)

1. 绘制路径

ctx.beginPath();

ctx.moveTo(x,y)

ctx.lineTo(x,y)

ctx.arc(x,y,r,sAngle, eAngle)

ctx.ellipse(x,y,rx,ry,sAngle, eAngle)

1. 绘制图像

ctx.drawImage(img, x, y)

ctx.drawImage(img, x, y, w, h)

1.第三方的绘制图表的工具库

百度“JS绘图库”......

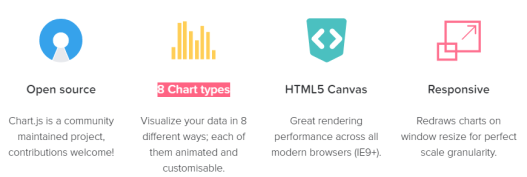
1. FusionCharts——收费 —— 80+种图表
2. HighChart——收费
3. JFreeChart——免费
4. Chart.js——免费 —— 8种图表

使用步骤：

1. 百度搜索名称，找到工具的官网
2. 到官网找API Document，边看手册边写代码

2.Chart.js的使用

官网：http://www.chartjs.org/



<canvas id="c1" width="700" height="500"></canvas>

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

<script>

//var chart = new Chart(c1, {图表的数据});

var chart = new Chart(c1, {

type: 'bar', //line/bar/pie/radar/doughnut/.....

options: {

responsive: false /\*取消响应式特性\*/

},

data: {

labels: ['1月','2月','3月','4月','5月'],

datasets: [{

label: '我的消费记录',

data:[3500, 5000, 8000, 10000, 6000]

}]

}

});

</script>

午间练习：给“京东用户中心”添加一个菜单项：“消费统计（Char.js版）”

HTML5新特性：

1. 表单新特性
2. 音频和视频
3. Canvas绘图——统计图、小游戏
4. SVG绘图
5. 地理定位
6. 拖放API
7. WebWorker
8. WebStorage
9. WebSocket

3.实现“幸运抽奖”小游戏

**多个异步请求间的同步问题：**

只有两张图片都加载完成，才能开始绘制游戏——但两张图片是异步加载的，无法事先预测哪个先加载完成。

var img1 = ...;

var img1Loaded = false;

img1.onload = function(){

img1Loaded = true;

if(img2Loaded){

//开始绘制

}

}

var img2 = ...;

var img2Loaded = false;

img2.onload = function(){

img2Loaded = true;

if(img1Loaded){

//开始绘制

}

}

扩展功能：

1. 在旋转停止时，判断所获奖项
2. 旋转开始后，转速逐渐提高，到达峰值后再逐渐降低
3. 抽奖完成后，把抽奖结果异步提交给服务器，保存入数据库

Canvas绘图技术的致命缺陷：



如何为某个图形(如矩形)添加事件监听？

因为每个图形图像都不是HTML元素，不能直接绑定事件监听函数！

所有的事件监听只能绑定给Canvas元素，再进一步的判定鼠标所在的位置，是否进入了特定的图像图形，设计响应的操作：

canvas.onmousemove = function(event){

var ex = event.offsetX;

var ey = event.offsetY;

if(ex>=150 && ex<=350 && ey>=150 && ey<=250){

console.log('鼠标在矩形上方');

}

}

浏览器中支持的绘图技术有三种：

1. Canvas绘图——HTML5新特性
2. SVG绘图——HTML5新特性
3. WebGL绘图——传统的绘图技术，尤其适合大型游戏！

|  |
| --- |
| **位图（bit）：**用三个整数描述一个颜色点，500px\*400px的一张位图就有200000个点，常见的位图：jpg、gif、png、psd.... **放大后会失真出现马赛克现象！**但善于描述颜色的细节/过渡。  **矢量图：**每条线/色块都用向量表示(有大小和方向的值)，常见的矢量图：.ai、swf.... 可以无限缩放而不会失真！但不善于描述丰富的颜色细节。 |

4.**SVG**绘图技术

Scalable Vector Graphic：可缩放的**矢量图**。

**Canvas绘图技术：**

属于**位图**绘图；

不能无限缩放；

只有一个Canavs元素，图形图像不是HTML元素；

适用于图像密集型游戏、统计图；

**SVG绘图技术：**

属于矢量图绘图；

可以不失真的缩放；

每一个图形/图像都是一个HTML元素，可以为任意图形绑定事件监听；

适合于统计图、地图应用，不适合于游戏；

提示：SVG技术早在HTML5之前就已经存在！

**在HTML5之前使用SVG的方法：**

1. 先创建一个XML文档，命名为xx.svg
2. HTML中使用<img><iframe><object><embed>标签显示SVG图形

**在HTML5之后使用SVG的方法：**

SVG绘图的所有标签都已经纳入到HTML5标签库，可以在网页中使用这些标签了。

**<body>**

**<h1>HTML5之后使用SVG的方法</h1>**

**<svg width="500" height="400">**

**<rect x="175" y="175" width="150" height="50"></rect>**

**<rect x="225" y="125" width="50" height="150"></rect>**

**</svg>**

**</body>**

5.使用SVG绘制各种图形图像——重点&非难点

**注意事项：**1)一般不使用内部样式设置图形的背景色等属性。 2)设置SVG图形的样式可以使用属性(都不是HTMLDOM属性)、伪行内样式(style="属性:值;")。 3)若想使用JS来读写SVG图形的属性，不能使用rect1.x形式！！只能使用setAttribute()和getAttribute(); 4)使用JS动态的添加新的SVG图形，必须使用document.createElementNS( 'http://www.w3.org/2000/svg', '标签名');

(1)绘制矩形

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

(2)绘制圆形

(3)绘制椭圆

(4)绘制直线

(5)绘制折线

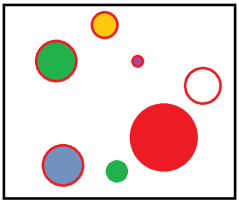
(6)绘制多边形

(7)绘制文字 <text>文字内容</text>

(8)绘制图像

练习： SVG绘图

(1)点击一个矩形/圆形，它慢慢的变宽、高、变淡



(2)使用SVG绘制“消费统计”图

服务器端返回如下JSON数据：

[ {"label":"1\u6708","value":3000},

{"label":"2\u6708","value":2000},

{"label":"3\u6708","value":6000},

{"label":"4\u6708","value":8000},

{"label":"5\u6708","value":4000} ]

# 6.22 SVG---two.js---地理定位

复习：

1. 表单新特性
2. 视频和音频
3. Canvas绘图——重点
4. SVG绘图——重点
5. 地理定位
6. 拖放API
7. WebWorker
8. WebStorage
9. WebSocket

Canvas绘图：位图，适用于图像密集型应用

SVG绘图：矢量图，适用于绘图、地图、图标，便于事件监听，有利于搜索引擎的检索

HTML5之前使用SVG:

1. 创建xx.svg XML文档
2. HTML页面中利用<img src="xx.svg">使用

HTML5之后使用SVG:

HTML5标签库中已经包含SVG图形图像，直接在HTML中使用SVG标签

1.使用SVG绘制各种图形——重点

**通用属性：**

fill="填充颜色"

stroke="描边颜色"

stroke-width="描边宽度"

style=" fill:''; stroke:'' "

opacity="透明度"

提示：可以把多个图形放在一个小组中，统一操作每个组员的样式

**(1)矩形**

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

**(2)圆形**

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

练习：在SVG中随机创建5个圆形，鼠标悬停某个圆上则出现一个边框，鼠标移出则边框消失 9:31~9:40

document.createElementNS('', 'circle')

**(3)椭圆**

<ellipse cx="" cy="" rx="横向半径" ry="纵向半径"></ellipse>

**(4)直线**

<line x1="" y1="" x2="" y2="" stroke="线条颜色"></line>

练习：绘制如下的小图标：



**(5)折线**

<polyline points="x1,y1 x2,y2 x3,y3 ..." fill="transparent" stroke="线条颜色"></polyline>

练习：绘制如下的小图标：

**(6)多边形**

<polygon points="x1,y1 x2,y2 x3,y3 ..."></polygon>

提示：多边形与折线的使用方法完全一样！只是多边形最后必须闭合！

练习： 绘制实心、空心五角星

**(7)文字**

<text x="" y="" font-size="20">文本内容</text>

提示：SVG中的文字直接出现在DOM树中，可以被搜索引擎的爬虫机器人检索到！

**(8)图像**

<image x="" y="" xlink:href=""></image>

提示：SVG中若绘制的了位图，无限缩放的特性就没有了！

2.为SVG图形图像添加特效——了解

所有的特效必须声明在<defs></defs>标签内，必须有ID，供某个图形/图像所引用。

**(1)渐变色特效**

<defs>

<linearGradient id="g1" x1="0%" y1="0%" x2="100%" y2="0%">

<stop offset="0%" stop-color="red"></stop>

<stop offset="100%" stop-color="green"></stop>

</linearGradient>

</defs>

<rect x="10" y="100" width="480" height="10" fill="url(#g1)"></rect>

**(2)滤镜特效——高斯模糊**

<defs>

<filter id="f5"> <!--定义滤镜元素-->

<feGaussianBlur stdDeviation="5"></feGaussianBlur>

</filter>

</defs>

<text x="0" y="100" font-size="100" filter="url(#f5)">达内科技</text>

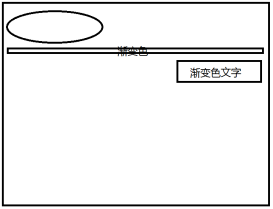
提示：可以到<http://www.w3school.com.cn/svg/svg_examples.asp>查看更多的滤镜特效

午间游戏：

1)使用SVG实现“蜻蜓点水”

2)使用多边形绘制“主页（小房子）”、“打分（星号）”

3)绘制如下的页面效果



**Canvas绘制矩形：**

ctx.fillStyle = 'red';

ctx.fillRect(x, y, w, h);

**SVG绘制矩形：**

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

**WebGL绘制矩形：**

......；

DOM => jQuery

Canvas/SVG => Two.js

3.第三方的绘图函数库——Two.js——扩展小知识

官网：http://jonobr1.github.io/two.js/

是一个2D绘图函数库，提供了一个完全一样的调用接口，用于在Canvas/SVG/WebGL上绘制图形。但该工具本身没有为Canvas/SVG添加新的功能。

1. 绘制基本图形：圆形、矩形、线条
2. 元素分组，统一修改样式、旋转和平移
3. 定时器动画

原理：每秒钟触发60次two.update事件，只需要在update事件的监听函数中修改元素的属性

4.Three.js——了解

官网：http://threejs.org/

是一个3D绘图函数库

LBS：Location Based Service，基于定位的服务

5.HTML5新特性之五——地理定位

Geolocation：地理定位，用于返回客户端当前所在的地理定位坐标（经度（longitude：-180~180）、纬度（latitude：0~90）、速度（speed）），用于LBS应用，如百度外卖、导航、基于定位的服务推介。

**实现原理：**

(1)手机中的浏览器：自带有GPS模块，浏览器可以读取此传感器的数据，精度可以在几米到几十米。

(2)PC中的浏览器：基于客户端的IP地址进行反向解析，精度大约在几公里到几十公里。

HTML5中新增了一个geolocation对象，专用于返回当前浏览器的地理定位数据： window.navigator.geolocation ，该对象包含三个成员方法：

1. getCurrentPosition() 返回当前的定位数据
2. watchPosition() 不间断的监视定位数据
3. clearWatch() 清除监视

使用方法：

var g = window.navigator.geolocation;

g.getCurrentPosition(succCallback,errCallback);

//定位成功后的回调函数

function succCallback(position){

console.log('定位成功!')

console.log('所在经度：'+position.coords.longitude);

console.log('所在纬度：'+position.coords.latitude);

console.log('当前速度：'+position.coords.speed);

console.log('海拔高度：'+position.coords.altitude);

}

//定位失败后的回调函数

function errCallback(err){

console.log('定位失败！');

console.log('失败编码：'+err.code);

console.log('失败信息：'+err.message);

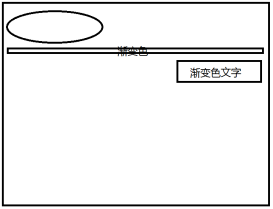
}

午间游戏：

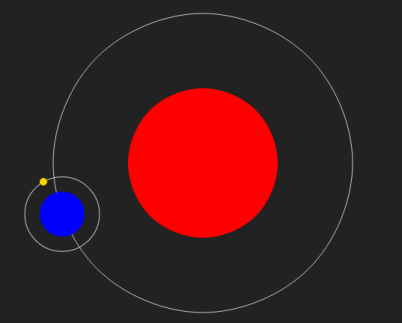
1)使用SVG实现“蜻蜓点水”

2)使用SVG多边形绘制“主页（小房子）”、“打分（星号）”

3)使用SVG绘制如下的页面效果



1603A 练习内容： Two.js实现“太阳系”



使用Two.js实现后，再使用Canvas技术实现。

# 6.23-百度地图API--拖放API

复习：

jQuery

AJAX

HTML5新特性

1. 表单新特性
2. 视频和音频
3. Canvas绘图 Chart.js
4. SVG绘图 Two.js
5. 地理定位 BMap(百度地图)
6. 拖放API jQueryUI Draggable
7. WebWorker
8. WebStorage
9. WebSocket

HTML5新特性之五——地理定位

浏览器获得地理定位数据的原理：

手机： GPS、基站、IP反向解析

PC： IP反向解析

window.navigator.geolocation.getCurrentPosition(

function(pos){ //定位成功的回调函数

//pos.coords.longitude

//pos.coords.latitude

//pos.coords.altitude

//pos.coords.speed

},

function(err){ //定位失败的回调函数

//err.id err.message

}

)

API：Application Programming Interface，应用程序编程接口，一个框架/组件/系统暴露给外界的可供调用的URL/对象/函数/方法，该接口可以返回特定格式的数据（如XML、JSON格式）。

2.百度地图API

官网：http://lbsyun.baidu.com/

使用步骤：

(1)注册新用户，或者使用QQ、微博、微信账号登陆

https://passport.baidu.com/v2/?reg&regType=1&tpl=mn&u=http://lbsyun.baidu.com/



(2)登录

https://passport.baidu.com/v2/?login&fr=old&login&u=http://lbsyun.baidu.com/

(3)查看开发者手册

http://lbsyun.baidu.com/index.php?title=jspopular



(4)每个使用百度地图API的应用必须申请一个唯一的AccessKey(AK)

http://lbsyun.baidu.com/apiconsole/key



(5)调用百度地图API

1)在地图上以某点为中心进行显示

2)添加地图控件

3)在地图上标记某点

DEMO地址： http://developer.baidu.com/map/jsdemo.htm#a1\_2



4.HTML5新特性之六——拖放API（Drag & Drop）——小重点

拖：Drag

放：Drop

拖放动作实际涉及到两个对象：

**源对象：**SourceObject，被拖动的对象

**目标对象：**TargetObject，被拖动的对象可以移动到目标对象上方，实施移出/释放操作

HTML5为拖放对象添加了7个新的事件：

**(1)源对象可以触发的事件：**

ondragstart：源对象开始被拖动

ondrag：源对象被拖动中

ondragend：源对象被拖动结束

**(2)目标对象可以触发的事件：**

ondragenter：源对象被拖动着进入目标对象上方

ondragover：源对象被拖动着处于目标对象上方(悬停静止/移动)

ondrop：源对象被拖动着在目标对象上方松开鼠标

ondragleave：源对象被拖动着离开目标对象上方

注意：ondragover事件结束后的默认行为就是要触发ondragleave，即默认情况下，ondrop永远无法触发。可能需要阻止ondragover事件的默认行为：**event.preventDefault()**;

练习：当图片被拖动着移动到DIV上方时，DIV边框变深；移出则又变淡；在DIV上方松开则图片移到DIV内部

如何在拖放源对象和目标对象间传递数据？

HTML5中所有拖放相关的事件对象都有一个属性：dataTransfer

(1)源对象上触发的事件对象与目标对象上触发的事件对象是不同的对象！

(2)源对象拖放事件的dataTransfer对于与目标上是不同的对象！

(3)每个dataTransfer对象都有setData()、getData()、clearData()等方法，用于在源对象和目标对象间传递数据

**源对象上：保存数据——数据装上拖拉机**

event.dataTransfer.setData('text/html', '数据');

**目标对象上：读取数据——从拖拉机上取出数据**

var data = event.dataTransfer.getData('text/html');

午间练习：实现一个可以随着鼠标拖动而移动位置的图片

提示：

（1）图片必须设置定位方式 absolute

（2）ondrag事件必须读取event.pageX和pageY，修改图片的left和top.......

（3）注意：松开鼠标前的一刹那，ondrag事件坐标的问题

5.补充知识点：如何在远程下载的网页中显示客户端本地的一张图片

**拖放事件的源对象：**

客户端操作系统中的图片文件

**拖放事件的目标对象：**

document.ondrop = function(){ 阻止默认行为：在新窗口中打开图片 }

div.ondrop = function(event){

//读取图片文件数据，创建<img src="">追加为div的子元素

var file = event.dataTransfer.files[0];

var reader = new FileReader();

reader.readAsDataURL( file );

reader.onload = function(){

var img = new Image();

img.src = reader.result;

div.appendChild( img );

}

}

|  |
| --- |
| XML DOM 中与文件操作相关的对象：   1. File 表示一个文件，用于描述文件的信息 2. FileList 类数组对象，表示一个文件列表，其中可能存在多个文件 .length 3. FileReader 文件读取器对象，用于读取一个File对象中的数据 |

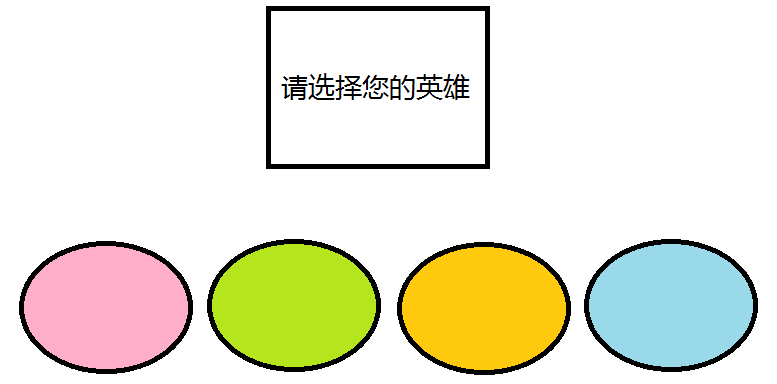
**普通URL：**

<img src="http://baidu.com/1.jpg">

**DataURL：**

<img src="data:image/jpeg;base64,/9j/4AAQSkZJRgABAQEAYABg">

1603A 练习内容： 选择您的游戏英雄



要求：

拖放选择英雄；

英雄选择区域最多出现一个英雄；

某个英雄被选中后，下方的英雄列表中该英雄以“不能再选”形式显示；

# 6.24 Webworker Storage

复习：

HTML5新特性之六——拖放API

(1)源对象可以触发的事件

ondragstart

ondrag

ondragend

(2)目标对象可以触发的事件

ondragenter

ondragover event.preventDefault();

ondrop

ondragleave

源对象和目标对象之间传递数据：

源对象：

ondragstart = function(){

event.dataTransfer.setData('text/html', '数据')

}

目标对象：

ondrop = function(){

var data = event.dataTransfer.getData('text/html')

}

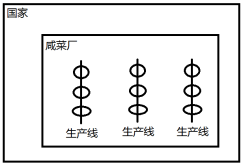
1. 表单新特性 5
2. 视频和音频 6
3. Canvas绘图 9
4. SVG绘图 8
5. 地理定位 1
6. 拖放API 4
7. Web Worker 3
8. Web Storage 7
9. Web Socket 2

1.Web Worker

**操作系统(国家)**：提供程序的运行环境，管理磁盘、内存；

**进程(咸菜厂)**：一个程序被操作系统从硬盘中调入内存开始执行，是分配内存的基本单位

**线程(生产线)**：线程是进程内部执行代码(生成任务)的基本单位



面试题：进程和线程的关系怎样？

进程(Process)是管理内存的基本单位；线程(Thread)处于进程内部，是进程内执行代码的基本单位。每个进程内部至少有一个线程！每个进程内部可以同时存在多个线程，并发执行。

**浏览器进程中发起HTTP请求时，可以使用多个线程（如Chrome中有6个请求线程）；**

**浏览器进程中负责渲染DOM树(执行所有的HTML/CSS/JS代码)的只有一个线程——称为UI主线程。 （UI：user Inerface , 用户界面）**

pan.onload = function(){

console.log('圆盘加载完成');

}

pin.onload = function(){

console.log('指针加载完成');

}

上述代码中，两张图片由于是不同的线程进行加载，理论上有可能同时到达客户端浏览器，但浏览器把它们渲染出来，只能一个一个的绘制。

若页面中需要加载一个很耗时（算法复杂）的JS代码，会长时间占用渲染主线程，影响到后续的HTML元素的渲染，以及已经存在的元素的事件监听、定时器都无法获得执行！

**解决方案：**创建一个新的线程，让新线程执行耗时的JS代码去！

**在浏览器中创建一个新的线程：**new Worker( 'xx.js' )

**Worker线程的致命缺陷(特意的设计)**——**不能执行任何DOM操作！**不能使用window/document....等对象——就是为了防止多个线程同时修改DOM树。

（如要真正优化，就最后用C语言）

2.扩展知识——了解

主渲染线程如何与Worker线程间传递数据？

**主渲染线程给Worker线程传递数据：**

主线程： var w1 = new Worker('x.js');

w1.postMessage('字符串数据');

Worker线程：

onmessage = function(event){ event.data; }

Worker线程给主渲染线程传递数据：

主线程： var w2 = new Worker('x.js');

w2.onmessage = function(event){ event.data; }

Worker线程：

postMessage( '字符串数据' );

午间作业：

1)主线程给Worker线程发送一个整数消息；Worker线程算出该数字是否为质数，再把判定结果传递给主线程。

2)绘制思维导图：

jQuery

AJAX

HTML5新特性：.........

3.HTML5新特性之八——Web Storage

**服务器端存储：** 存储项目的核心数据；

1. 数据库
2. IO技术
3. Session技术

**客户端存储：**存储当前客户端的访问记录、内容偏好

1. Cookie：所有浏览器都支持，但数据量不能超过4KB
2. Flash：必须依赖于Flash播放器
3. Web Storage：Web存储，数据量不能超过8MB
4. Indexed DB：索引数据库
5. FileAPI： File、FileList、FileReader...

客户端存储的应用示例：

记录登录的账户信息、根据浏览历史进行推荐、主题选择、内容/板块定制

4.Web Storage的两种类型——重点 (window.可以省略)

(1)sessionStorage：会话级存储，可以在一个网站打开的多个页面之间传递数据；把数据保存在客户端浏览器内存中，只能在当前浏览器窗口中使用！窗口关闭或新建窗口都无法使用其它窗口中的sessionStorage数据

window.sessionStorage.setItem(key, value); //保存条目

var value = window.sessionStorage.getItem(key); //读取条目

window.sessionStorage.removeItem(key); //删除条目

window.sessionStorage.clear( ); //清空所有条目

演示：登录后的用户名可以在其它页面中显示。

(2)localStorage：本地/跨会话级存储，可以在一个网站打开的多个页面之间传递数据；把数据保存在客户端磁盘中，是永久存储，可以同一个网站的任意一个页面所使用！窗口关闭或新建窗口都可以继续使用其它窗口中的localStorage数据。

window.localStorage.setItem(key, value); //保存条目

var value = window. localStorage.getItem(key); //读取条目

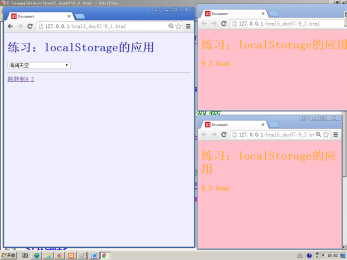
window. localStorage.removeItem(key); //删除条目

window. localStorage.clear( ); //清空所有条目

练习：在1.html选择(下拉菜单)一个用户喜欢的主题(每个主题对应不同的body背景色和字体颜色)，到了2.html仍然可以看到该主题的样式；重启浏览器仍然可以看到之前选择的主题样式

每当当前窗口中网页的localStorage被修改(是其它页面做的修改)，当前页面会触发：

window.onstorage = function(){ localStrorage存储的内容发生改变 }

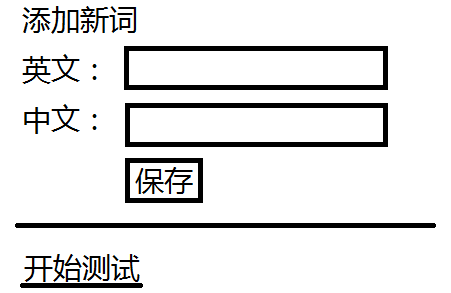


练习：实现登录页面中“□一周之内不再登录”，在其它页面中可以显示出欢迎回来：XX

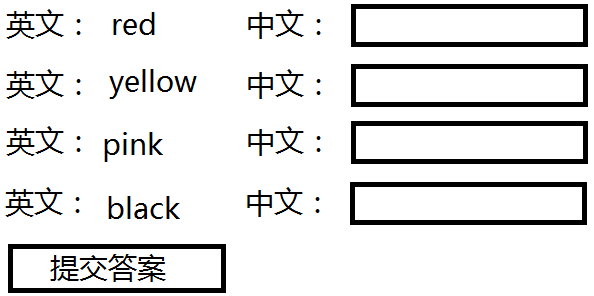
若用户勾选了，则此次登录成功后，会在客户端永久存储其用户名和密码；当再次访问登录页面时，已经填充好用户名和密码。

1603A 练习内容： 单词本

创建一个word\_save.html，多次点击“保存”，可以录入多个单词



再创建一个word\_test.html，不停的弹出英文，让用户输入中文，提示正确和错误的单词数。



提示：遍历出本地存储的所有键值对

for(var i=0; i<localStorage.length; i++){

var k = localStorage.key(i); //得到第i个键

var v = localStorage.getItem( k ); //得到第i个键对应的值

}

# 6.25.Web Storage 飞机大战1

复习：

(1)Web Worker

用于在主渲染线程之外创建一个新的线程，执行一个耗时的JS任务。

**主线程：**

var w1 = new Worker('x.js');

var inputData = ...;

w1.postMessage(inputData);

w1.onmessage = function( event ){

//event.data

}

**Worker线程(x.js)：**

//不能操作DOM，不能使用window、document、元素、样式等

onmessage = function(event){

//event.data

////处理数据....

var result = ....;

postMessage( result );

}

(2)Web Storage

用于在客户端存储数据，如访问记录、登录信息、主题定制、板块定制。

1. Cookie：字符数据，不能超4KB，操作复杂，兼容性好
2. Web Storage：字符数据，不能超过8MB，操作简单，H5新特性

window.sessionStorage：会话级存储，数据存于浏览器进程内存

window.localStorage：跨会话级存储，数据存于磁盘文件中

存储数据：setItem(k, v)

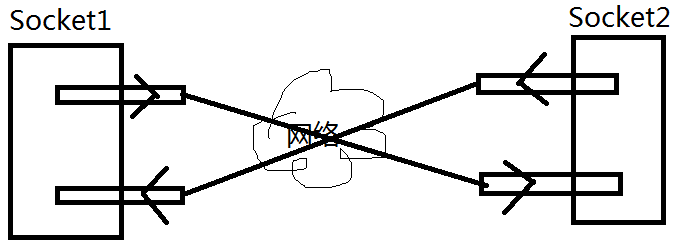
读取数据：getItem(k)

数据个数：length

第i个键：key(i)

1.HTML5新特性之九——Web Socket

Socket：套接字，插座，用于向网络上发送数据，从网络上接收数据。



**HTTP协议**：基于“请求-响应”模型，客户端发一个请求，服务器返回一个响应，先有请求才能有响应，一个请求只能获得一个响应。缺陷：网络聊天室、实时走势图使用HTTP协议力不从心！

**WS协议**：WebSocket协议，客户端一旦连接到服务器就不再断开(永久连接)，某一方可以给对方连续发送多条消息，对方只接收而不发送消息。缺陷：服务器同一时间可以保持的总连接数是有限的。

**使用JS创建WebSocket协议的客户端：**

var client = new WebSocket( 'ws://服务器地址:端口' );

client.onopen = function(){ //客户端连接到ws服务器

//每次接收到服务器的消息都会触发

client.onmessage = function(){

//客户端接收到服务器的消息

}

client.send( '发给服务器的消息1' );

client.send( '发给服务器的消息2' );

client.send( '发给服务器的消息3' );

}

**创建使用WebSocket协议的服务器端：**

1. 可以使用独立的PHP文件——脱离Apache直接运行
2. 可以使用支持WS协议的Java程序
3. 可以使用支持WS协议NodeJS程序

使用NodeJS作为服务器的WebSocket聊天室：

https://segmentfault.com/a/1190000000479518

2.HTML5新特性阶段项目——飞机大战

所用技术： Canvas技术 + 音频播放 + WebStorage

实现功能： 打敌机，赚积分！

难度提示： 属于碰撞类游戏，前期会比较简单，后期代码逻辑结构复杂。

游戏阶段：

1. DOWNLOADING
2. READY
3. STARTING
4. PLAY
5. PAUSE
6. GAMEOVER

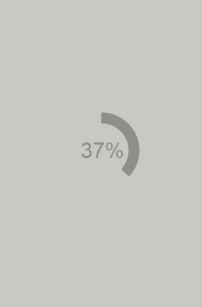
项目目录：

img/

index.html

app.js

午间练习：加载游戏需要的所有图片，绘制一个圆形进度条



var imgBullet1 = new Image();

imgBullet1.src = 'img/bullet1.png';

imgBullet1.onload = function(){

progress += 3;

drawProgress();

}

3.飞机大战阶段1——下载阶段

绘制圆形进度条

var progress = 0;

img.src = 'xx.png';

img.onload = function(){

progress += 权重值;

//绘制进度

ctx.arc( 圆心X, 圆心Y, 半径, 起始角度, 终止角度 )

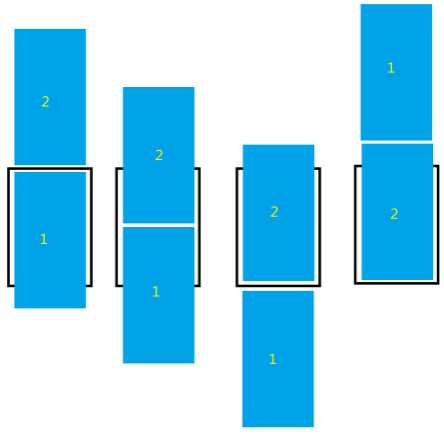
}

4.飞机大战阶段2——就绪阶段

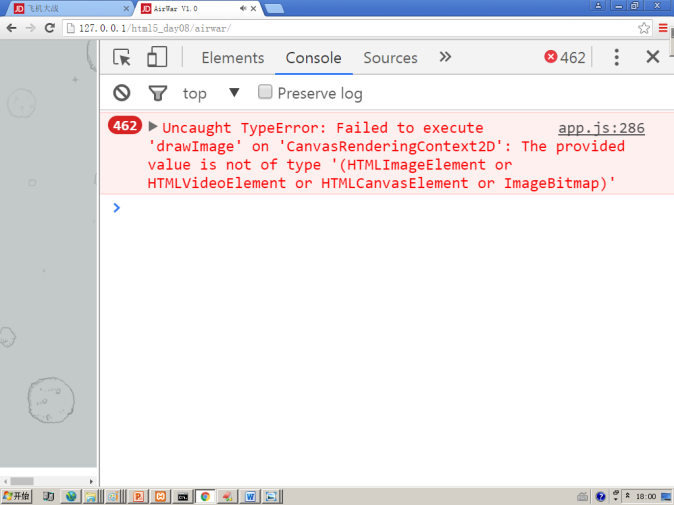
静态图片呈现出“动画”效果——使用定时器，每隔一段时间重新绘制。

游戏引擎(Engine)：指游戏中的定时器，游戏中最好只有一个主定时器，每次的绘制时间间隔固定。

如何使用有限长度的图片实现无限延伸的背景效果：



项目中的常见错误：



# 6.27 新浪云(重点) 飞机2

复习：

HTML5阶段项目——飞机大战

1. PHASE\_DOWNLOAD

var progress = 0;

function drawProgress(){ }

img.onload = function(){

progress += 3;

drawProgress();

}

1. PHASE\_READY

startEngine(); //定时器

var sky = new Sky(...);

function Sky(){ }

drawLogo(){}

1. PHASE\_STARTING

var runningPlane = new RunningPlane()

function RunningPlane(){ }

1. PHASE\_PLAY

var hero = new Hero();

function Hero(){ } //我方英雄

function Bullet(){ } //一个子弹

function BulletList(){} //我方子弹列表

--------------------------------

function Enemy1(){ } //小号敌机

function Enemy2(){ } //中号敌机

function Enemy3(){ } //大号敌机

function EnemyList(){ } //敌机列表

----------------------------------

碰撞检验 //难点&重点

1. PHASE\_PAUSE

canvas.onmouseout = ....

canvas.onmouseover = ....

1. PHASE\_GAMEOVER

hero.lives === 0

1.调试游戏程序中的错误

注意 debugger 断点的使用！！！

2.如何随机的生成不同种类的敌人

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

var num = Math.floor(Math.random()\*200);

if(num<6){ //创建小号敌机 6

this.add( new Enemy1(imgsEnemy1) );

}else if(num<9){ //创建中号敌机 3

this.add( new Enemy2(imgsEnemy2) );

}else if(num<10){ //创建大号敌机 1

this.add( new Enemy3(imgsEnemy3) );

}

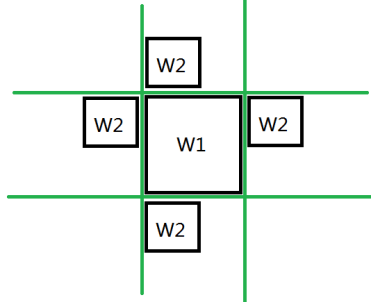
3.如何实现碰撞检验

两种碰撞检验方法：

1. 坐标相交
2. 颜色相交

坐标相交碰撞，必须满足4个条件：

1. w2.x + w2.width >= w1.x
2. w1.x + w1.width >= w2.x
3. w2.y + w2.height >= w1.y
4. w1.y + w1.height >= w2.y



每个子弹 与 每个敌机 碰撞检验

我方英雄 与 每个敌机 碰撞检验

**4.云服务器的使用**

AmazonCloud

阿里云

百度云

腾讯云

新浪云 ￥1=100云豆

新浪云服务器（Web服务器+DB服务器）的使用步骤：

1. 注册微博账号

官方地址： http://www.weibo.com



1. 注册新浪云账号

邀请码地址：http://t.cn/R4WcToI

注意“安全邮箱”和“安全密码”，是以后上传代码、管理数据库的用户名和密码！

1. 登录新浪云

登录地址：http://sae.sina.com.cn

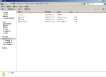
 

1. 创建一个Web应用，获得一个免费的二级域名

PHP解释器必须选择5.6版本！

1. 创建代码仓库，上传自己的项目代码

1. 使用PC/手机浏览器访问自己的项目



-------------------------------------------------------

(6)（可选）启用云数据库服务器，提交SQL语句

注意：修改下列配置信息

$conn = mysqli\_connect('w.rdc.sae.sina.com.cn','用户名','密码','app\_andemaxiya','3307');

找自学视频： SVN、Git 代码管理工具的使用

# 插件：

弹出框插件：jquery.fancybox-1.3.4.pack.js

鼠标论滚：jquery.mousewheel-3.0.4.pack.js

样式模仿例子：

https://www.fundebug.com/

# 备注：

获得当前页面的地址：

**Window.document.location.href;**

Console.log(location.href);

直接获得请求结果：

function queryReplyMsg(num){  
 var \_url="/mer/queryMessageReply.htm";  
 var cont=$.ajax({  
 url:\_url,  
 data : {messageType:num},  
 type : "POST",  
 cache : false,  
 global : false,  
 dataType : "json",  
  **async : false** //async : false,表示同步，同步会等待执行完成，知道success 执行完成！如果是true，倒是有可能先执行alert("谁先"); }).responseText;  
 return eval("("+cont+")"); //返回请求结果data  
}

问： [jQuery获取未选中的checkbox](http://blog.csdn.net/zh921112/article/details/41803187)

答：var unCheckedBoxs = $("input[name='myCheckbox']").not("input:checked");

**问：定位方法：**

var \_top = obj.offsetTop;  
var \_left = obj.offsetLeft;  
var \_height = obj.offsetHeight;  
  
while ((obj = obj.offsetParent) != null) {  
 \_top += obj.offsetTop;  
 \_left += obj.offsetLeft;  
}

**问：点击空白处，隐藏弹出框**

$(document).click(function(e){  
 if($**(e.target).parent()[0].className!**='show\_tips\_L'  
 && **e.target.className!**='show\_tips\_L'

&& **e.target.id....** && $(e.target).parent()[0].className!='bind\_LPho'){  
 $('#licence\_msg .show\_tips\_L').fadeOut();  
 }  
});

**问：H5判断网络类型**

navigator对象中的connection对象有一个type属性 是判断网络类型的

**问：释放jquery的$关键字**

jQuery.noConflict();  
或者  
var jq=$.noConflict();  
具体可以看下[http://www.w3school.com.cn/jquery/core\_noconflict.asp](http://www.w3school.com.cn/jquery/core_noconflict.asp" \t "http://bbs.blueidea.com/_blank)