search\_engine主要考虑以下功能：

1，后台由php实现，用php调用webapi,并结构化显示返回xml内容（可转换成json？）

2，前端页面设计部分copy，嵌入php文件中

3，适配pc和移动端

实施细节：（参考用）

1，form使用“method='get'”

理由如下：

1）由于本设计没有安全性的考虑（密码，定单等）

2）提交数据少，get时url length<2083

3) 浏览器可缓存,直白来讲即是说,在回退/前进时,不用再次提交,且能SEO

4) W3C的官方建议是：当且仅当form是幂等（idempotent）的时候，使用get。一般来说，如

果提交这个请求纯粹只是从服务端获取数据而不进行其他操作，并且多次提交不会有明显的副

作用，应该使用get。(学习\*)

2, span内联元素的应用:

span标签主要是靠多种多样的id和class属性增加语义。

3，role属性：role属性是用来强调语义的，

1. font face property:

为字体显示提供更强大的支持。以前只能调用用户机上已有的字体进行显示，现在可以合理地从服务器下载想要用户看到的字体。

1) Firefox、Chrome、Safari 以及 Opera 支持 .ttf (True Type Fonts) 和 .otf (OpenType Fonts) 类型的字体。

2) Internet Explorer 9+ 支持新的 @font-face 规则，但是仅支持 .eot 类型的字体 (Embedded OpenType)。Internet Explorer 8 以及更早的版本不支持新的 @font-face 规则。

格式：

@font-face {

Font-family: //user define;

Font-style://for different use like normal, italic, oblique;

Font-weight://bold or specific number;

Src:local(), url(), format();

Font-display:;

}

两点需要注意：

一是src属性：

**<url> [ format( <string># ) ]?**

Specifies an external reference consisting of a [<url>](https://developer.mozilla.org/zh-CN/docs/Web/CSS/url" \o "CSS 数据类型 <url> 指向一个资源。它没有独有的表达形式，只能通过 url() 函数定义。), followed by an optional hint using the format() function to describe the format of the font resource referenced by that URL. The format hint contains a comma-separated list of format strings that denote well-known font formats. If a user agent doesn't support the specified formats, it skips downloading the font resource. If no format hints are supplied, the font resource is always downloaded.

二是font-display:我们在@font-face指令中使用font-display来加载自定义字体。

这个属性可以添加以下的值：

auto:默认值。典型的浏览器字体加载的行为会发生，也就是使用自定义字体的文本会先被隐藏，直到字体加载结束才会显示。

swap:后备文本立即显示直到自定义字体加载完成后再使用自定义字体渲染文本。在大多数情况下，这就是我们所追求的效果。之前提及到的JavaScript脚本实现的功能就基本和这个是一致的。

fallback:这个可以说是auto和swap的一种折中方式。需要使用自定义字体渲染的文本会在较短的时间（100ms according to Google ）不可见，如果自定义字体还没有加载结束，那么就先加载无样式的文本。一旦自定义字体加载结束，那么文本就会被正确赋予样式。

optional:效果和fallback几乎一样，都是先在极短的时间内文本不可见，然后再加载无样式的文本。不过optional选项可以让浏览器自由决定是否使用自定义字体，而这个决定很大程度上取决于浏览器的连接速度。如果速度很慢，那你的自定义字体可能就不会被使用。

(https://www.w3cplus.com/css/font-display-masses.html)

注：

TrueType(.ttf)

　　Windows和Mac系统最常用的字体格式，其最大的特点就是它是由一种数学模式来进行定义的基于轮廓技术的字体，这使得它们比基于矢量的字体更容易处理，保证了屏幕与打印输出的一致性。同时，这类字体和矢量字体一样可以随意缩放、旋转而不必担心会出现锯齿。

EOT – Embedded Open Type (.eot)

　　EOT是嵌入式字体，是微软开发的技术。允许OpenType字体用@font-face嵌入到网页并下载至浏览器渲染，存储在临时安装文件夹下。

OpenType (.otf)

　　OpenType是微软和Adobe共同开发的字体，微软的IE浏览器全部采用这种字体。致力于替代TrueType字体。

WOFF – Web Open Font Format (.woff)

　　WOFF（Web开发字体格式）是一种专门为了Web而设计的字体格式标准，实际上是对于TrueType/OpenType等字体格式的封装，每个字体文件中含有字体以及针对字体的元数据（Metadata），字体文件被压缩，以便于网络传输。

SVG (Scalable Vector Graphics) Fonts (.svg)

SVG是由W3C制定的开放标准的图形格式。SVG字体就是使用SVG技术来呈现字体，还有一种gzip压缩格式的SVG字体。

<!-- 适配移动端

viewport,是移动端特有。这是一个用于承载网页的虚拟的区域，。

承载关系：浏览器>viewport>网页

适配要求：

1. 网页宽度必须和浏览器保持一致

2. 默认显示的缩放比例和PC端保持（缩放比例1.0）

3. 不允许用户自行缩放网页

满足这些要求达到了适配，国际上通用的适配方案，标准的移动端适配方案。

-->

C、robots(机器人向导)

说明：robots用来告诉搜索机器人哪些页面需要索引，哪些页面不需要索引。

content的参数有all,none,index,noindex,follow,nofollow。默认是all。

举例：

1

<meta name="robots" content="none">

具体参数如下：

信息参数为all：文件将被检索，且页面上的链接可以被查询；

信息参数为none：文件将不被检索，且页面上的链接不可以被查询；

信息参数为index：文件将被检索；

信息参数为follow：页面上的链接可以被查询；

信息参数为noindex：文件将不被检索，但页面上的链接可以被查询；

信息参数为nofollow：文件将被检索，但页面上的链接不可以被查询；

script提供了两种方式来解决上述问题，async和defer，这两个属性使得script都不会阻塞DOM的渲染。

如果script标签设置了该属性，则浏览器会异步的下载该文件并且不会影响到后续DOM的渲染；  
如果有多个设置了defer的script标签存在，则会按照顺序执行所有的script；  
defer脚本会在文档渲染完毕后，DOMContentLoaded事件调用前执行。

async的设置，会使得script脚本异步的加载并在允许的情况下执行  
async的执行，并不会按着script在页面中的顺序来执行，而是谁先加载完谁执行。

defer

如果你的脚本代码依赖于页面中的DOM元素（文档是否解析完毕），或者被其他脚本文件依赖。

例：

评论框

代码语法高亮

polyfill.js

async

如果你的脚本并不关心页面中的DOM元素（文档是否解析完毕），并且也不会产生其他脚本需要的数据。

例：

百度统计

如果不太能确定的话，用defer总是会比async稳定

甘特图（Gantt chart）又称为横道图、条状图(Bar chart)。其通过条状图来显示项目，进度，和其他时间相关的系统进展的内在关系随着时间进展的情况。以提出者亨利·L·甘特（Henrry L. Ganntt）先生的名字命名。

（注：<https://www.cnblogs.com/jiasm/p/7683930.html）>

!important 的用法：

CSS写在不同的地方有不同的优先级，css文件中的定义 < 元素style中的属性，但是如果使用!important，就会提高指定样式规则的应用优先权，语法格式{ sRule!important }，即写在定义的最后面

**一、!important语法**

选择器{样式:值!important;}

!important为开发者提供了一个增加样式权重的方法。应当注意的是!important是对整条样式的声明，包括这个样式的属性和属性值

**三、!important应该注意的一些地方**

1、如果!important被用于一个简写的样式属性，那么这条简写的样式属性所代表的子属性都会被作用上!important。

2、关键字!important必须放在一行样式的末尾并且要放在该行分号前，否则就没有效果。 （不过分号前的空格不会影响它）

3、如果因为一些特殊原因，你不得不在一个代码块中声明两个同样的属性，那么请把!important加在第一个属性后面，因为这样做会让所有浏览器中第一个属性的权重更大，而IE6除外（这是一个只有IE6才有的hack，但是不会影响你的CSS）

**四、何时该使用!important**

**1、帮助测试可访问性**

正如前面提到过的，用户样式中可以包含!important声明，允许用户根据自己的特殊需求来给具体的CSS样式增加权重以帮助他们阅读和访问内容。

有特殊需求的用户可以把!important加到像font-size这样的打印属性上来加大字体，或者为了提高网页的对比度而把它加到与颜色相关的样式上。

**2、暂时解决紧急问题**

总会有这样的情况发生——某个客户的线上网站出现了CSS的bug，你必须快速修复。在大多数情况下你应该会用Firebug或者其他的开发者工具来调试你的CSS代码并且最终修复它。但是，如果这个bug发生在IE6或者是其他没有提供调试工具的浏览器上的时候，你可能需要使用!important来完成快速修复。

当你用这个临时修复的办法让网站继续上线以后（这样可以先摆平客户），你再花些时间用不破坏层叠且可维护性高的方法来修复这个bug也为时未晚。当你找到更好的解决方案你就可以替换掉线上的!important部分，而且客户对此毫不知情。

**3、用Firebug或者其他开发者工具覆写样式**

我们可以使用Firebug或者Chrome开发者工具（译注：两者均可用F12唤出）查看页面元素，在不影响真正的CSS样式的情况下可以自由地编辑样式，测试效果，调试bug等等。

!important可以被加在单行样式的后面从而让已经被覆盖的样式重新作用。这样的话，你不用大量修改你真正的样式就可以调试你的CSS，直到你找到问题所在。

**4、覆写用户生成内容中的行内样式**

CSS开发中一个让人头疼的问题，就是当我们遇到用户生成内容中有行内样式的情况，这种情况多发生在一些使用了所见即所得网页编辑器的CMS(content management system)系统中。在CSS层叠中，行内样式会覆盖常规样式。然而，我们并不想看到那些本该被我们定义好的CSS样式作用的元素却被用户生成内容中的行内样式改变。此时你就可以用!important来避免这种问题，因为，一条被开发者加了!important的样式会覆盖行内样式。

**关于单独设计的博客页面**

如果你曾经接触过设计个人博客页面的技术,有一种需求要求每个单独的文章都要有它自己独立的样式，需要使用行内样式。你可以用!important代码为一个个人页面编写它自己的样式。

!important在这个时候可以派上用场，为了在你的站点里创建一个独一无二的页面，你可以轻松覆盖默认的样式而不必担心原本的CSS权重。

**五、!important总结**

如果你遇到了特殊的需求，或者你想覆盖用户代理（译注：即浏览器）和开发者的默认样式，提高页面可访问性，你最好慎重使用!important。你要对你的CSS在考虑周全的同时尽可能地避免使用!important。甚至在上面提到的许多能用到!important的地方它也不总是必要的。

尽管如此，!important在CSS中还是有用的。如果你接手一个项目，而这个项目之前的开发人员正好使用了!important，那么你在修复一些问题的时候我们今天所讨论的将会派上用场。

Meta中og标签对SEO的作用及用法

meta property=og标签对SEO的作用及用法，如果你仔细观察会发现本站点<head>代码中有一段："property="og:image"这段代码是什么意思呢？property="og:image可以让网页成为一个“富媒体对象”。

用了Meta Property=og标签，就是你同意了网页内容可以被其他社会化网站引用等，目前这种协议被SNS网站如Fackbook、renren采用。

SNS已经成为网络上的一大热门应用，优质的内容通过分享在好友间迅速传播。为了提高站外内容的传播效率，2010年F8会议上Facebook公布 了一套开放内容协议(Open Graph Protocol)，任何网页只要遵守该协议，SNS就能从页面上提取最有效的信息并呈现给用户。

富媒体标签SEO

Open Graph Protocol应用指南

参与到Open Graph Protocol的好处

● 能够正确的分享您的内容到SNS网站

● 帮助您的内容更有效的在SNS网络中传播Meta Property标签的应用

● 按照您网页的类型，在<head>中添加入meta标签,并填上相应的内容

● 如果一个页面上有多个需要标识出的内容怎么办？

您可以重复meta标签，将认为og:type 标签是每一段内容的起始处，

● 安装SNS网站插件如人人网插件，同样会出现Meta Property标签，卸载后可以去除。

使用Open Graph Protocol应注意

meta property=og代码的功能并不等同于网页的meta name标签，两者针对的对象不一致，功能不同。

如果网站上要使用Open Graph Protocol，那么，meta property=og和meta name、Title标签应同时赋值。

文章开头举例的B2C网站，<head>如下：

<head>

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

<title>Cool Gadgets at the Right Price – Worldwide Free Shipping – \*\*\*\*\*\*</title>

<meta property=”og:title” content=”\*\*’sFacebook” />

<meta property=”og:type” content=”product”/>

<meta property=”og:url” content=”http://www.\*\*\*\*\*\*.com” />

<meta property=”og:image” content=”http://www.\*\*\*\*\*\*.com/logo.gif”/>

<meta property=”og:site\_name” content=”\*\*\*\*\*\*.COM” />

<meta property=”og:description” content=”CoolGadgets at the RightPrice – Worldwide Free Shipping – \*\*\*\*\*\*”/>

<meta property=”fb:admins” content=”100001422224225″ />

<meta property=”fb:page\_id” content=”241333394220″/>

<link href=”/css/reset.css” rel=”stylesheet”type=”text/css” />

<link href=”/css/default/all.16666.css” rel=”stylesheet” type=”text/css” />

<script type=”text/javascript” src=”/shop.js”></script>

<script type=”text/javascript” src=”/scripts/all.16666.build.js”></script> )

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

<meta name=”description” content=”Online shopping for cool gadgets at the right price. ” />

<meta name=”keywords” content=”cool gadgets, cheap gadgets, free shipping” />

</head>

如果你用上了Meta Property标签，你得在后台设置好，不然，就是空白.

Open Graph Protocol对SEO的影响

有些人使用所谓的网站质量在线检测，检测后结果提示Meta Property=og这段代码有问题，特别是一些SEO的检测，更是提示“特别错误”！然后删除Meta Property=og代码，检测正常。

因此担心Meta Property=og这段代码会对网站照成不良影响。

其实Open Graph Protocol并不会对SEO照成不良影响，相反，应用的合理，非常有利于网站的推广。

Protocol Element 协议元素

　　说明:

1. Null 表示是否允许为空

2. Mult 表示是否允许重复

3. Y 允许

4. N 禁止

Base Element基本类型

og:type 网页资源类型标识

Null Mult

N N

content enum:

● video 视频

● audio 音频

● link 链接

● photo 图片

● product 产品

og:title 标题描述

Null Mult

N N

og:image 缩略图

og:url 当前内容链

Null Mult

Y N

rr:appid 如果您的网站是CONNECT到renren.com的，请提供该ID

Null Mult

Y N

Video Element 视频

og:videosrc 视频资源链接，例如可是播放视频的flash地址

Null Mult

N Y

og:width 视频的宽度

Null Mult

Y Y

og:height 视频的高度

Null Mult

Y Y

Audio Element 音频

　　类似http://domain/some.mp3 直接从mp3获取信息，以下指从音乐播放网页获取

og:audiosrc 音乐资源链接，例如可是播放歌曲的flash地址

og:artist 音乐家

例如：

<meta property=”og:type” content=”audio” />

<meta property=”og:title” content=”小酒窝” />

<meta property=”og:image” content=”http://…./xx.jpg” />

<meta property=”og:url” content=”http://somedomain.com/aaamp3.html” />

<meta property=”og:audiosrc” content=”http://somedomain.com/audioplayer.swf?s=xxx.mp3″ />

<meta property=”og:artist” content=”林俊杰A,林俊杰B” />

Commen Page Element 普通网页

og:abstract 内容摘要

Null Mult

N N

og:contentid 内容主体的ID，用来标识当前页面主要内容所处的HTML标签的ID

Null Mult

Y N

例如：

<meta property=”og:type” content=”link” />

<meta property=”og:title” content=”文艺青年摄影装逼完全指南v2.4\*冲片+针孔教程” />

<meta property=”og:image”content=”/yingxiao/UploadFiles\_4687/201603/20160308141820615.png” />

<meta property=”og:url”content=”http://www.douban.com/group/topic/10560486/” />

<meta property=”og:abstract” content=”前言：” />

<meta property=”og:contentid” content=”topic” />

Graphic Element 图片

og:photo 图片列表

og:width 图片宽度

og:height 图片高度

例如：

<meta property=”og:type” content=”photo” />

<meta property=”og:title” content=”发国·招贴(2)” />

<meta property=”og:image”content=”/yingxiao/UploadFiles\_4687/201603/20160308141820588.jpg”/>

<meta property=”og:url” content=”http://www.douban.com/photos/photo/487762954/” />

<meta property=”og:photo”content=”/yingxiao/UploadFiles\_4687/201603/20160308141820502.jpg”/>

<meta property=”og:photo”content=”/yingxiao/UploadFiles\_4687/201603/20160308141820407.jpg”/>

Product Element 商品

og:price 产品价格

Null Mult

N N

og:description 产品描述

Null Mult

Y N

og:nick 店铺名

Null Mult

Y N

og:postfee 运

Null Mult

Y N

-webkit-/-moz-/-o-/-ms- 这些私有前缀是为了兼容不同的前端浏览器的。

Keyframes:

@keyframes *animationname* {*keyframes-selector* {*css-styles*;}}

|  |  |
| --- | --- |
| *animationname* | 必需。定义动画的名称。 |
| *keyframes-selector* | 必需。动画时长的百分比。  合法的值：   * 0-100% * from（与 0% 相同） * to（与 100% 相同） |
| *css-styles* | 必需。一个或多个合法的 CSS 样式属性。 |