什么是模块化开发？模块化开发的规范？

模块化开发：是指按照特定的规范划分为各个独立的个体。

模块化开发的规范:

AMD : 异步模块规范，需要的时候才回去加载模块，是一个异步过程，常用于客户端

CMD : 通用模块规范，初始化的时候就加载需要的模块，是一个同步的过程，常用于服务器端

http 数据请求的同步和异步，跨域问题？怎么决绝跨域问题？

跨域：js出于安全考虑是js是不能访问其他服务器上的数据，即同源策略，而跨域就是通过某些手段绕过同源策略，实现不同域之间的通信。

什么情况下会产生跨域：不用域之间有数据

需要调用其他域的数据或者内容，就会产生跨域

决绝跨域问题：（1）document.domain+iframe： 解决主域相同而子域不同的跨域问题

（2）location.hash＋iframe：

（3）跨域资源共享（CORS）就是使用自定义的HTTP让浏览器与服务器沟通，决定请求是否成功

（4）window.name实现的跨域数据传输

（5）jsonp

swig执行的原理：首先是把HTML文件读取出来，然后用 swig 的引擎去解析，把 {{}} 中的内容用 程序上下文中的变量去替换，{%%}中的内容调用 node 的执行语句去执行，把渲染好的文本放入 HTTP body中，并发到客户端

git原理：

mysql : 是一个关系型数据库管理系统，关联数据库将数据放到不同的数据表中

mongoDB ：

(1).mongodb 的弱一致性，更能保证数据库的访问速度，mysql中count会对数据集进行锁定，这样可以的到精确的数据，但是现在数据是不断更新和变化的，这种精确变得没什么意义了，但是mongoDB在某些情况下会锁定数据库，

(2).mongoDB的文档结构的存储方式，可以更便捷的获取数据

(3).内置GridFS(分布式文件管理系统)支持海量的数据存储

(4)mongoDB性能优越

缺点：（1）不支持事务操作 （2）占用空间较大

tcp/ip：

回调函数的原理

自适应网站布局：

就是使用一套css样式来适应所有屏幕的大小，而不是给每个屏幕尺寸的设备制作一个站点

<meta name=’viewport’ content=’width:divice-width,initial-scale:1.0’>

Css @Viewport 和media query 配合使用

@viewport {

width: device-width;

zoom: 2;

}

Initial-scale:1.0 网站初始大小占屏幕面积的100%

Media-query IE10/11 , opera, webkit, moz 都已支持，，且需要厂商前缀。

@-webkit-viewport {

width: device-width;

}

@-moz-viewport {

width: device-width;

}

@-ms-viewport {

width: device-width;

}

@-o-viewport {

width: device-width;

}

@viewport {

width: device-width;

}

**call()和apply()**

运用call（）或者apply()方法强制转换上下文环境

原型和原型链：

原生ajax原理：

就是XMLHttpRequest通过向服务器发起异步请求，从服务器获取数据，然后js动态修改html 来操作dom,局部刷新页面

同步和异步的区别：

同步就是一个请求发送完毕之后，在没有得到结果之前是不会继续执行后边的程序

异步就是一个请求发送完成后，不管有没有收到回复，后边的程序都会继续执行

grunt gulp :都可以实现代码的合并和压缩，图片的压缩，gulp 代码更短更简洁

css属性在ie下的差异

ie7下识别不出来displsy:inline-block

解决的方案是：设置display:inline-block;  \*zoom: 1;\*display: inline;

Zoom : 触发ie的hasLayout属性，清除浮动、清除margin的重叠

为什么html5中只用声明<!DOCTYPE html>

因为html 4.01里边需要对DTD（规范定义集）进行引用，html4.01是基于SGML的，它定义了三种声明<!DOCTYPE> ，而html5是不基于SGML的

层叠上下文布局原理：

BFC:块级格式化上下文

IFC：行内格式化上下文

元素的display属性会决定盒是行内级还是块级：

block, table, flex, grid, list-item 为块级

inline, inline-block, inline-table, inline-flex, inline-grid 为行内级

css全局属性（很多）：目前经常使用的draggable，id，style，title等等

web 语义化：包括html语义化和css语义化，

http 数据请求的方法：get:向服务器发送资源 post,head,

闭包的概念：闭包就是定义在函数内部的函数，可以用来访问其他内部函数的变量，闭包会将这些变量的值存到内存当中,并且会保留局部的作用域

如何进行网站的优化：服务器：使用cdn,Ajax使用GET进行请求,避免空的src的img标签 content：将资源放到不同的域下，减少iframe的使用，避免重定向

cookies:减小cookie的规模 css:要将样式表放在顶部，不使用css表达式

js: 对js文件进行压缩

图片：使用延迟加载，优化图片进行压缩