什么是响应式网页?

通过 CSS3 Media Query 实现响应式 Web 设计

响应式 Web 设计(Responsive Web design)的理念是，页面的设计与开发应当根据用户行为以及设备环境(系统平台、屏幕尺寸、屏幕定向等)进行相应的响应和调整。

具体的实践方式由多方面组成，包括弹性网格和布局、图片、CSS media query 的使用等。无论用户正在使用笔记本还是 iPad，我们的页面都应该能够自动切换分辨率、图片尺寸及相关脚本功能等，以适应不同设备；换句话说，页面应该有能力去自动响应用户的设备环境。这样，我们就可以不必为不断到来的新设备做专门的版本设计和开发了。

Doctype? 严格模式与混杂模式-如何触发这两种模式，区分它们有何意义?

声明位于文档中的最前面的位置，处于 标签之前。此标签可告知浏览器文档使用哪种 HTML 或 XHTML 规范。

标准模式和混杂模式（quirks mode）。在标准模式中，浏览器根据规范呈现页面；在混杂模式中，页面以一种比较宽松的向后兼容的方式显示。混杂模式通常模拟老式浏览器（比如 Microsoft IE 4 和 Netscape Navigator 4）的行为以防止老站点无法工作。

在 IE 6 出现时，在标准模式中使用的是正确的盒模型，在混杂模式中使用的则是老式的专有盒模型。为了维持对 IE 5 和更低版本的向后兼容性，Opera 7 和更高版本也在混杂模式中使用有缺点的 IE 盒模型。

前端页面有哪三层构成，分别是什么？作用是什么？

网页的结构层（structural layer）由 HTML 或 XHTML 之类的标记语言负责创建。标签，也就是那些出现在尖括号里的单词，对网页内容的语义含义做出了描述，但这些标签不包含任何关于如何显示有关内容的信息。例如，P 标签表达了这样一种语义：“这是一个文本段。”

网页的表示层（presentation layer） 由 CSS 负责创建。 CSS 对“如何显示有关内容”的问题做出了回答。

网页的行为层（behavior layer）负责回答“内容应该如何对事件做出反应”这一问题。这是 Javascript 语言和 DOM 主宰的领域。

使用 (X)HTML 去搭建文档的结构。

使用 CSS 去设置文档的呈现效果。

使用 DOM 脚本去实现文档的行为

如何居中一个浮动元素?

方法一：让最外面的层相对定位，left 等于 50%，然后内部嵌套层也使用相对定位且 left 设为-50%，这样的效果就是内层相对整行为水平居中；

方法二：使用 display: table;

方法三：直接使用 table 布局（使用太多 table 容易让结构看起来比较混乱，其实页面中使用少量的 table，只要不要嵌套使用，还是可以实现使用少量 CSS，达到最好的效果的），这种方法这里就不举例演示了。

如何让 ie6,7,8，兼容 html5 的标签？

我一直使用公司里的提供的 jqside 插件，里面就是把 html5 的标签放到字符串，用字符串的 split 的方法变为数组，用 while 的方法，变量减减，用 dom 的 createElement 方法进行再 ie678 里创建标签。

```
if( $.isIE678 ){
  Var
    html5="abbr,article,aside,audio,canvas,datalist,details,dialog,even
    tsource,figure,footer,header,hgroup,mark,menu,meter,nav,output,prog
    ress,section,time,vid eo".split(','),
    i = html5.length;
  while(i--) document.createElement(html5[i]);
}
```

在 Css 中那个属性会影响 dom 读取文档流的顺序？

Float

行内元素有哪些？块级元素有哪些？CSS 的盒模型？

Css 的盒模型：从外到里，margin，border，padding，content。

块元素在页面里，占一行，可以设定宽和高，可以容纳块元素和行内元素。常见的块元素有 div，p，h1-h6，ul 等。

行内元素没有宽和高的属性但可以与其他元素同一行，一般不可以包含块元素，行内元素的高度一般由元素内部的字体大小决定，宽度由内容的长度控制。常见的行内元素有 a，b，span，strong，em 等。

CSS 引入的方式有哪些? link 和@import 的区别是?

内联引用 CSS。可灵巧应用样式於各标签中。方便于编写代码时的使用。没有整篇文件的“统一性”，在需要修改某样式的时候也变的比较困难。

内部引用 CSS 将样式规则写在<STYLE>...</STYLE>标签之中。

外部引用 link 标签引用 CSS 将样式规则写在.css 的样式文件中，再以<link>标签引入。这样引入该 css 样式表文件以后，就可以直接套用该样式档案中所制定的样式了。

外部引用 @import 引用 CSS 跟 link 方法很像，但必须放在<STYLE>..</STYLE>中： <STYLE TYPE="text/css"><!--@import url(引入的样式表的位址、路径与档名);--> </STYLE>可以灵活的引入 css 文件对 xhtml 元素进行控制。有时候也用来编写某些 css hack。这种方法的缺点：在个别文件或元素的灵活度不足老祖宗的差别。link 属于 XHTML 标签，而@import 完全是 CSS 提供的一种方式。

加载顺序的差别。当一个页面被加载的时候（就是被浏览者浏览的时候），link 引用的 CSS 会同时被加 载，而@import 引用的 CSS 会等到页面全部被下载完再被加载。

兼容性的差别。由于@import 是 CSS2.1 提出的所以老的浏览器不支持，@import 只有在 IE5 以上的才能识别，而 link 标签无此问题。

使用 dom 控制样式时的差别。当使用 javascript 控制 dom 去改变样式的时候，只能使用 link 标签，因为@import 不是 dom 可以控制的。

CSS 选择符有哪些？哪些属性可以继承？优先级算法如何计算？内联和 important 哪个优先级高？

ID 和 CLASS
Class 可继承;伪类 A 标签可以继承;列表 UL,LI,DL,DD,DT 可继承
优先级就近原则，样式定义最近者为准载入样式以最后载入的定位为准
优先级为:!important>id>class>tag;Important 比内联优先级高

CSS 层叠是什么?介绍一下？

CSS 翻译过来叫做层叠样式表。运用到层叠的时候，主要就是靠 CSS 的组合与子选择器。去编辑样式。它的作用是定义网页的外观（例如字体，颜色等等），它也可以和 javascript 等浏览器端脚本语言合作做出许多动态的效果。

CSS 指层叠样式表 (Cascading Style Sheets)样式定义如何显示 HTML 元素样式通常存储在样式表中把样式添加到 HTML 4.0 中，是为了解决内容与表现分离的问题外部样式表可以极大提高工作效率外部样式表通常存储在 CSS 文件中多个样式定义可层叠为一

html 的意义？

HTML 指的是超文本标记语言 (Hyper Text Markup Language)
HTML 不是一种编程语言，而是一种标记语言 (markup language)
标记语言是一套标记标签 (markup tag)
HTML 使用标记标签来描述网页
HTML 标记标签通常被称为 HTML 标签 (HTML tag)
HTML 标签是由尖括号包围的关键词，比如 <html>
HTML 标签通常是成对出现的，比如 和
标签对中的第一个标签是开始标签，第二个标签是结束标签
开始和结束标签也被称为开放标签和闭合标签

介绍 HTML5 和 CSS3(对比)?

HTML 5 还包含了新的语义化的元素标签，比如：<nav>,<header>,<footer>,<aside>以及 <figure> 等等 。

拖放（Drag 和 drop）是 HTML5 标准的组成部分。canvas 元素用于在网页上绘制图形。

HTML5 支持内联 SVG（矢量图形）

Canvas 和 SVG 都允许您在浏览器中创建图形，但是它们在根本上是不同的。

HTML5 Geolocation（地理定位）用于定位用户的位置。在谷歌地图上显示您的位置。

HTML5 引入了应用程序缓存，这意味着 web 应用可进行缓存，并可在没有因特网连接时进行访问。

web worker 是运行在后台的 JavaScript，独立于其他脚本，不会影响页面的性能。您可以继续做任何愿意做的事情：点击、选取内容等 等，而此时 web worker 在后台运行。

在客户端存储数据

HTML5 提供了两种在客户端存储数据的新方法：

localStorage - 没有时间限制的数据存储

sessionStorage - 针对一个 session 的数据存储

HTML5 服务器发送事件（server-sent event）允许网页获得来自服务器的更新。

border-image 原理？

1.调用边框图片 border-image 的 url 属性，通过相对或绝对路径链接图片。

2，边框图片的建裁 border-image 的数值参数剪裁边框图片，形成九宫格。

3.剪裁图片的边框边框图片被切割成 9 部分，以一一对应的关系放到 div 边框的九宫格中，然后再压缩（或拉伸）至边框（border-width 或 border-image-width）的宽度大小。

4.执行重复属性被填充至边框九宫格四个角落的的边框图片是不执行重复属性的。上下的九宫格执行水平方向的重复属性（拉伸或平铺），左右的格子执行垂直方向的重复属性，而中间的那个格子则水平重复和垂直方向的重复都要执行。

5、完成绘制，实现效果

自定义手机 UI 组件？如何实现的？

Input,button,radio,checkbox.

把本身的 input 隐藏掉，给后面的 label 进行样式，并且用 label 的 for 属性，去指定 input 的 id。去点击 label 的时候，css3 的：chcked 和：disabled 去确定状态，样式用 css 精灵去排版。

图片切换的实现思路?

以全局监听的方式通过 a 标签的描点进行 view 动态切换页面，只要把 a 标签带有 id 的 href 属性的值指到锚点，用 CSS3 的动画进行切换页面.

HTML5 都有哪些新的 JS API?

二维绘图 API，可以用在一个新的画布（Canvas）元素上以呈现图像、游戏图形或者其他运行中的可视图形。

一个允许 web 应用程序将自身注册为某个协议或 MIME 类型的 API。

一个引入新的缓存机制以支持脱机 web 应用程序的 API。

一个能够播放视频和音频的 API，可以使用新的 video 和 audio 元素。

一个历史纪录 API，它可以公开正在浏览的历史纪录，从而允许页面更好地支持 AJAX 应用程序中实现对后退功能。

跨文档/跨域的消息传递，它提供了一种方式，使得文档可以互相通信而不用考虑它们的来源域，在某种程度上，这样的设计是为了防止跨站点的脚本攻击。

一个支持拖放操作的 API，用它可以与 draggable 特性相关联。

一个支持编辑操作的 API，用它可以与一个新的全局 contenteditable 特性相关联。

一个新的网络 API，它支持 web 应用程序在本地网络上互相通信，并在它们的源服务器上维持双向的通信。

使用 JavaScript API 的键/值对实现客户端的持久化存储，同时支持嵌入的 SQL 数据库。

服务器发送的事件，通过它可以与新的事件源（event-source）元素关联，新的事件源元素有利于与远程数据源的持久性连接，而且极大地消除了在 Web 应用程序中对轮询的需求。

AMD 和 CMD 是什么？它们的区别有哪些？

AMD 和 CMD 是二种模块定义规范。现在都使用模块化编程，AMD，异步模块定义；CMD，通用模块定义。AMD 依赖前置，CMD 依赖就近。CMD 的 API 职责单一，没有全局 require，AMD 的一个 API 可以多用。

MVC BFC

mvc 是模型(model)－视图(view)－控制器(controller)的缩写，一种软件设计典范使用 MVC 的目的是将 M 和 V 的实现代码分离，从而使同一个程序可以使用不同的表现形式。MVC 对应 Html，CSS，js。

BFC 全称”Block Formatting Context”，中文为“块级格式化上下文”。流体特性：块状水平元素，如 div 元素（下同），在默认情况下（非浮动、绝对定位等），水平方向会自动填满外部的容器；BFC 元素特性表现原则就是，内部子元素不会影响外部的元素。

HTML 5 增加了一项新功能是 自定义数据属性 ，也就是 data- 自定义属性。

在 HTML5 中我们可以使用以 data- 为前缀来设置我们需要的自定义属性，来进行一些数据的存放。

```
<div id = "user" data-uid = "12345" data-uname = "愚人码头"></div>
// 使用 getAttribute 获取 data- 属性
var user = document . getElementById ( 'user' ) ;
var userName =user . getAttribute ( 'data-uname' ) ; // userName = '愚人码头'
var userId = user . getAttribute ( 'data-uid' ) ; // userId = '12345'
使用 setAttribute 设置 data- 属性
user . setAttribute ( 'data-site' , 'http://www.css88.com' ) ;
```

Position 的四个属性详解

Position 的四个属性：static，fixed，absolute，relative

static，是 position 属性的默认值，也就是无特殊定位，遵循 html 定位规则。

fixed，是相对于浏览器窗口进行定位，不随页面的上下翻动而移动，比如固定在页面末端的二维码等。

absolute，它是相对于它的第一个父元素进行定位，如果你想让这个 div#demo 里的一个子 div#sub 相对于#demo 定位在右上角的某个地方，应该给#demo 相对定位，#sub 绝对定位。

此时，它的第一个父元素就是 id=demo 的 div；否则它的父元素就是 body，这样它的位置在页面中保持不变，但是随着页面移动会发生变化（区别 fixed）。

relative，relative 是相对于自己来定位的，相对于其正常位置进行定位。例如：#demo{position:relative;top:-50px;},这时#demo 会在相对于它原来的位置上移 50px。

P.S:采用左列 left 浮动，右列不浮动，采用 margin-left 定位的方式。

css 中 box 和 flex

首先'box'呐是比较早的语法，使用它时需要带上前缀，比如 display: -webkit-box; / Chrome 4+, Safari 3.1, iOS Safari 3.2+ /，而"flex"是 2012 年的语法，是 css3 新规定的，也将是以后标准的语法。将父元素的 display 属性设置为-webkit-box（box），然后子元素通过属性-webkit-box-flex 来指定一个框的子元素是否是灵活的或固定的大小，如上，定义两个灵活的 p 元素。如果父级 box 的总宽度为 300px，#P1 将有一个 100px 的宽度，#P2 将有一个 200px 的宽度，也就是呈固定比例划分。当然了，也可以这样写：

```
<div style="background-color: pink;width: 500px;height:500px;display:flex">
  <p style="background-color: orange;flex:1.0;border:1px solid red;">111</p>
  <p style="background-color: orangered;flex:2.0;border:1px solid
blue;">222</p>
</div>
```

当然了 css3 规定了，一系列的有关 box 的属性，比如 box-shadow。。。。。

JQ 常见选择器？

，号选择器，分组选择器。空格，祖父选择器。>大于号，父子选择器。+号选择器，紧接下一个兄弟选择器。~号，元素之后所有的 siblings 元素。

：first，：last，：not，：first-child，:last-child,:animated.：checked

jQuery 插件实现方式，分别介绍？

jQuery.fn.extend 封装直接在$下面的方法，就是根下面，

扩展 jQuery 元素集来提供新的方法（通常用来制作插件）。

$.extend 用来在 jQuery 命名空间上增加新函数。用一个或多个其他对象来扩展一个对象，返回被扩展的对象

批量的方法用 fn，静态的用$.extend（），不建议用扩展到根下面。

bind 和 live 的区别?

live 方法其实是 bind 方法的变种，其基本功能就同 bind 方法的功能是一样的，都是为一个元素绑定某个事件，但是 bind 方法只能给当前存在的元素绑定事件，对于事后采用 JS 等方式新生成的元素无效，而 live 方法则正好弥补了 bind 方法的这个缺陷，它可以对后 生成的元素也可以绑定相应的事件.

js 和 jq 如何转换？

jQuery 对象是通过 jQuery 包装 DOM 对象后产生的对象。jQuery 对象是 jQuery 独有的，其可以使用 jQuery 里的方法，但是不能使用 DOM 的方法；例如：

$("#img").attr("src","test.jpg"); 这里的 $("#img")就是 jQuery 对象。

DOM 对象就是 Javascript 固有的一些对象操作。DOM 对象能使用 Javascript 固有的方法，但是不能使用 jQuery 里的方法。例如：

document.getElementById("img").src = “test.jpg”; 这里的 document.getElementById("img") 就是 DOM 对象。

$("#img").attr("src","test.jpg"); 和 document.getElementById("img").src = "test.jpg"; 是等价的，是正确的，但是 $("#img").src = "test.jpg" ；或者 document.getElementById("img").attr("src","test.jpg"); 都是错误的。

DOM 对象转成 jQuery 对象

对于已经是一个 DOM 对象，只需要用 $() 把 DOM 对象包装起来，就可以获得一个

jQuery 对象了，$(DOM 对象)

如：var v = document.getElementById("v"); //DOM 对象

var $v = $(v); //jQuery 对象

转换后，就可以任意使用 jQuery 的方法。

jQuery 对象转成 DOM 对象

两种转换方式讲一个 jQuery 对象转换成 DOM 对象： [index] 和 .get(index);

(1) jQuery 对象是一个数据对象，可以通过 [index] 的方法，来得到相应的 DOM 对象。

如： var $v = $("#v");//jQuery 对象

var v = $v[0]; //DOM 对象

alert(v.checked); //检测这个 checkbox 是否被选中

(2) jQuery 本身提供，通过.get(index) 方法得到相应的 DOM 对象

如：var $v = $("#v");//jQuery 对象

var v = $v.get(0); //DOM 对象 ($v.get()[0] 也可以)

alert(v.checked); //检测这个 checkbox 是否被选中

通过以上方法，可以任意的相互转换 jQuery 对象和 DOM 对象，需要再强调的是：

DOM 对象才能使用 DOM 中的方法，jQuery 对象是不可以使用ＤＯＭ中的方法

给出一个数组如何去掉重复的项？

实现一个把数组里面的重复元素去除的方法：

主要的是 Array 的 prototype 的方法。

```
var arr=[1,3,5,3,6,9,1,2,2]
var arr=['a','b','a','c','c','ab','bc']
function removeRepeat(arr){
    var i,tmpArr=[];
    for(i in arr){
      if(tmpArr.join(',').indexOf(arr[i])==-1){
        tmpArr.push(arr[i]);
      }
    }
    return tmpArr;
}
var r=(arremoveRepeatr);
console.log(r);
```

二．方法：

```
Array.prototype.unique=function(){
    var i,tmpArr=[];
    for(i in this){
      if(typeof this[i]!='function'){
        if(tmpArr.join(',').indexOf(this[i])==-1){
          tmpArr.push(this[i]);
        }
      }
    }
  return tmpArr;
}
var arr=['a','b','a','c','c','ab','bc'];
var r=arr.unique();
console.log(r);
```

js 如何实现面向对象？

```
var name = 'Chen Hao';
var email = 'haoel(@)hotmail.com';
var website = 'http://coolshell.cn';
var chenhao = {
    name : 'Chen Hao',
    email : 'haoel(@)hotmail.com',
    website : 'http://coolshell.cn'
};
//以成员的方式 chenhao.name; chenhao.email; chenhao.website;
//以 hash map 的方式 chenhao["name"]; chenhao["email"];
chenhao["website"];
//我们可以看到， 其用 function 来做 class。
var Person = function(name, email, website){
  this.name = name; this.email = email; this.website = website;
  this.sayHello = function(){
    var hello = "Hello, I'm "+ this.name + ",  \n" + "my email is: " +
this.email + ",  \n" + "my website is: " + this.website;
    alert(hello);
  };
};
var chenhao = new Person("Chen Hao", "haoel@hotmail.com",
"http://coolshell.cn");
chenhao.sayHello();
```

Javascript 的数据和成员封装很简单。没有类完全是对象操作。纯动态！

Javascript function 中的 this 指针很关键，如果没有的话，那就是局部变量或局部函数。去找最紧跟的上一个 function。

Javascript 对象成员函数可以在使用时临时声明，并把一个全局函数直接赋过去就好了。

Javascript 的成员函数可以在实例上进行修改，也就是说不同实例相同函数名的行为不一定一样。

js 如何实现继承?

继承是指一个对象直接使用另一对象的属性和方法

实现方法：对象冒充，及 call（）Apply（）参见上述 call 和 apply 的用法。

原型链方式：js 中每个对象均有一个隐藏的__proto__属性，一个实例化对象的__proto__属性指向其类的 prototype 方法，而这个 prototype 方法又可以被赋值成另一个实例化对象，这个对象的__proto__又需要指向其类，由此形成一条链。

那么__proto__是什么？每个对象都会在其内部初始化一个属性，就是__proto__，当我们访问一个对象的属性 时，如果这个对象内部不存在这个属性，那么他就会去__proto__里找这个属性，这个__proto__又会有自己的__proto__，于是就这样 一直找下去，也就是我们平时所说的原型链的概念。

定义一个 Dog 对象，并增加一个 name 属性，该属性可以在新建对象时通过参数传入

```
function Dog( name ){ this.name = name;}
// 通过原型方式扩展 Dog 对象
Dog.prototype = {
  // 重新覆盖构造函数让其指向 Dog constructor:Dog,
  Wow:function(){
    console.group();
    console.info("I am: "+this.name );
    console.info("WangWang....");
    console.groupEnd();
  },
  yelp:function(){this.Wow();}
};
function MadDog(name){ Dog.apply( this, [name]);}
MadDog.prototype=new Dog();
// 重新覆盖构造函数，让其指向 MadDog
MadDog.prototype.constructor=MadDog;
MadDog.prototype.yelp=function(){
  self=this;
  setInterval(function(){self.Wow();},5000);
}
var xiaoXian=new Dog("xiaoXian"); xiaoXian.yelp();
var xiaoMang=new MadDog("xiaoMang"); xiaoMang.yelp();
console.log( xiaoXian.constructor == xiaoMang.constructor );
```

如果扩展 js 中原生的 String 对象?string 的方法？

String.prototype.name= function(){}

Slice 从字符串的第一个参数提取第二个参数，也可以截取数组。返回的结果类型：string/object

Substring 从字符串的第一个参数提取第二个参数，返回的结果类型，string。

Indexof 返回短字符串在长字符串出现的位置。

Lastindexof 返回最后一个短字符串出现的位置。

Replace 字符串的替换方法，

Split 字符串分割方法，能转换为数组，数组转换字符串，用 json（）方法。

document.ready()和 window.onload 的区别?

Document.ready()是 jQuery 中准备出发的事件，当加载到当前元素就执行了，

Window.onload 是整个页面加载之后才执行

闭包是什么？

闭包是有权访问另一个函数作用域中的变量的函数。

闭包是个函数，而它“记住了周围发生了什么”。表现为由“一个函数”体中定义了“另一个函数”

“闭包”是一个表达式（一般是函数），它具有自由变量以及绑定这些变量的环境（该环境“封闭了”这个表达式）。

1.闭包有权访问函数内部的所有变量。

2.当函数返回一个闭包时，这个函数的作用域将会一直在内存中保存到闭包不存在为止。

```
function f() {
  var rs = [];
  for (var i=0; i <10; i++) {
    rs[i] = function() {return i;};
  }
  return rs;
}
var fn = f();
for (var i = 0; i < fn.length; i++) {
  console.log('函数 fn[' + i + ']()返回值:' + fn[i]());
}
```

介绍 jQuery easyUI？jQuery easyUI 组件使用?

jQuery EasyUI 是一组基于 jQuery 的 UI 插件集合，而 jQuery EasyUI 的目标就是帮助 web 开发者更轻松的打造出功能丰富并且美观的 UI 界面。开发者不需要编写复杂的 javascript，也不需要对 css 样式有深入的了解，开发者需要了解的只有一些简单的 html 标签。

布局 layout，上南，下北，左西右东，中间内容，左边的组件是：tree，手风琴，中间的有 tab，tab 里有 datagrid 数据表格，还有数据表格的 toolbar 工具栏。对话框 dialog;

什么是 DOM，什么是 DOM？以及它们的用法？

BOM 即浏览器对象模型，浏览器对象模型 (BOM) 使 JavaScript 有能

力与浏览器“对话”， 由于现代浏览器已经（几乎）实现了 JavaScript 交互性方面的相同方法和

属性，因此常被认为是 BOM 的方法和属性。所有浏览器都支持 window 对象。它表示浏览器窗口。

所有 JavaScript 全局对象、函数以及变量均自动成为 window 对象的成员。

全局变量是 window 对象的属性。

全局函数是 window 对象的方法。

甚至 HTML DOM 的 document 也是 window 对象的属性之一：

Window 对象其实是 DOM 中所有的对象都源自 window 对象，有 location 对象， history 对象，方法有.Resize（）
alert( ) .confirm( ).prompt( ).open().close().setInterval().setTimeout()。

window.open() - 打开新窗口

window.close() - 关闭当前窗口

window.moveTo() - 移动当前窗口

window.resizeTo() - 调整当前窗口的尺寸

DOM 的文档对象模型，最顶级的对象是 document。可以 js 通过操作 DOM，就是一个接口，可以访问 html 的标准方法。要改变页面的某个东西，JavaScript 就需要获得对 HTML 文档中所有元素进行访问的入口。这个入口，连同对 HTML 元 素进行添加、移动、改变或移除的方法和属性，都是通过文档对象模型来获得的（DOM）。

JSON 和 JSONP 的区别？JSONP 的原理？

JSON(JavaScript Object Notation)和 JSONP(JSON with Padding)虽然只有一个字母的差别，但其实他们根本不是一回事儿：JSON 是一种数据交换格式，而 JSONP 是一种依靠开发人员的聪明才智创造出的一种非官方跨域数据交互协议。

1、一个众所周知的问题，Ajax 直接请求普通文件存在跨域无权限访问的问题，甭管你是静态页面、动态网页、web 服务、WCF，只要是跨域请求，一律不准；

2、不过我们又发现，Web 页面上调用 js 文件时则不受是否跨域的影响（不仅如此，我们还发现凡是拥有”src”这个属性的标签都拥有跨域的能力，比如<script>、<img>、<iframe>）；

3、于是可以判断，当前阶段如果想通过纯 web 端（ActiveX 控件、服务端代理、属于未来的 HTML5 之 Websocket 等方式不算）跨域访问数据就只有一种可能，那就是在远程服务器上设法把数据装进 js 格式的文件里，供客户端调用和进一步处理；

4、恰巧我们已经知道有一种叫做 JSON 的纯字符数据格式可以简洁的描述复杂数据，更妙的是 JSON 还被 js 原生支持，所以在客户端几乎可以随心所欲的处理这种格式的数据；

5、这样子解决方案就呼之欲出了，web 客户端通过与调用脚本一模一样的方式，来调用跨域服务器上动态生成的 js 格式文件（一般以 JSON 为后缀），显而易见，服务器之所以要动态生成 JSON 文件，目的就在于把客户端需要的数据装入进去。

6、客户端在对 JSON 文件调用成功之后，也就获得了自己所需的数据，剩下的就是按照自己需求进行处理和展现了，这种获取远程数据的方式看起来非常像 AJAX，但其实并不一样。

7、为了便于客户端使用数据，逐渐形成了一种非正式传输协议，人们把它称作 JSONP，该协议的一个要点就是允许用户传递一个 callback 参数给服务端，然后服务端返回数据时会将这个 callback 参数作为函数名来包裹住 JSON 数据，这样客户端就可以随意定制自己的函数来自动处理返回数据了。

如何解析 json？

用 ajax 去请求 json 数据，在回调函数里，把数据传过到函数里。通过用一个 for 循环，用 innerHTML 和 jQuery 的方法 html（）的方法，渲染到页面里。

手机浏览器独有的三个事件？

touchstart 事件：当手指触摸屏幕时候触发，即使已经有一个手指放在屏幕上也会触发。

touchmove 事件：当手指在屏幕上滑动的时候连续地触发。在这个事件发生期间，调用 preventDefault()事件可以阻止滚动。

touchend 事件：当手指从屏幕上离开的时候触发。

touchcancel 事件：当系统停止跟踪触摸的时候触发。关于这个事件的确切出发时间，文档中并没有具体说明，咱们只能去猜测了。

为什么要用 Zepto?

jquery 适用于 PC 端桌面环境，桌面环境更加复杂，jquery 需要考虑的因素非常多，尤其表现在兼容性上面，相对于 PC 端，移动端的发杂都远不及 PC 端,手机上

的带宽永远比不上 pc 端。pc 端下载 jquery 到本地只需要 1~3 秒（90+K），但是移动端就慢了很多，2G 网络下你会看到一大片空白网页在加载，相信用户第二次就没打开的欲望了。zepto 解决了这个问题，只有不到 10K 的大小，2G 网络环境下也毫无压力，表现不逊色于 jquery。所以移动端开发首选框架，个人推荐 zepto.js。

jq mobi，zepoto 手机跨平台的手机框架。

如何区分多个终端（zepto）？

detece 来判断：

```
//general device type
$.os.phone $.os.tablet
// specific OS $.os.ios $.os.android $.os.webos $.os.blackberry
$.os.bb10 $.os.rimtabletos
// specific device type $.os.iphone
$.os.ipad $.os.touchpad $.os.kindle
// specific browser $.browser.chrome $.browser.firefox $.browser.silk
$.browser.playbook
// Additionally,version information is available as well.
// Here's what's returned for an iPhone running
iOS 6.1. !!$.os.phone//=>true
!!$.os.iphone//=> true !!$.os.ios//=>true !!$.os.version//=>"6.1"
!!$.browser.version//=>"536.26"
```

简述下 cookie 的操作，还有 cookie 的属性都知道哪些？

Session 是由应用服务器维持的一个服务器端的存储空间，用户在连接服务器时，会由服务器生成一个唯一的 SessionID,用该 SessionID 为标识符来存取服务器端的 Session 存储空间。

Cookie 是客户端的存储空间，由浏览器来维持。

如果不设置过期时间，则表示这个 cookie 生命周期为浏览器会话期间，只要关闭浏览器窗口，cookie 就消失了。

AJAX 是什么?AJAX 的交互模型(流程)?AJAX 跨域的解决办法?同步和异步的区别?

Asynchronous JavaScript and XML（异步 JavaScript 和 XML），是一种创建交互式网页应用的网页开发技术。简单的说它是多种技术的组合，目的就是让前台的数据交互变得更快捷，不用刷新页面就可以完成数据的更新。

优点很明显，利于用户体验，不会打断用户的操作，在不刷新页面的情况下更新内容，减小服务器压力也是它很硬性的一个优点；而缺点，除了倍受追捧的 SEO 问题，安全问题、前进后退（这个虽然可以用其他方法解决，但 AJAX 本身的机制还是没变）、破坏程序的异常机制以及对新兴手持设备支持不完美的问题都是它现存的一些缺点。

readyState 五种状态：

请求未初始化，还没有调用 open()。

请求已经建立，但是还没有发送，还没有调用 send()。

请求已发送，正在处理中（通常现在可以从响应中获取内容头）。

请求在处理中；通常响应中已有部分数据可用了，没有全部完成。

响应已完成；您可以获取并使用服务器的响应了。

status 常见的几种状态

100——客户必须继续发出请求

101——客户要求服务器根据请求转换 HTTP 协议版本

200——成功

201——提示知道新文件的 URL

300——请求的资源可在多处得到
301——删除请求数据
302------缓存问题
404——没有发现文件、查询或 URl
500——服务器产生内部错误

正则表达式有系统学习过吗(看书或网上教程)?有的话就问问简单点的邮箱验证、URL 验证，或者问问 贪婪匹配与懒惰匹配 的理论知识？

验证邮箱

```
function isEmail(str){
  var reg = /^([a-zA-Z0-9_-])+@([a-zA-Z0-9_-])+(\.[a-zA-Z0-9_-])+/;
 return reg.test(str);
}
```

验证日期格式

```
function testReg(reg,str){
  return reg.test(str);
}
var reg = /^\d{4}-\d{1,2}-\d{1,2}$/;
```

字母和数字的组合

```
function istrue(str){
  var reg=/^(([a-z]+[0-9]+)|([0-9]+[a-z]+))[a-z0-9]*$/i;
  return reg.test(str);
}
```

正则匹配价格

```
function checkPrice( me ){
  if( !( /^(?:\d+|\d+\.\d{0,2})$/.test(me.value) ) ){
    me.value = me.value.replace(/^(\d*\.\d{0,2}|\d+).*$/,'$1');
  }
}
```

电话号码正则

```
telReg = /^\d{3,4}-\d{7,8}(-\d{3,4})?$/
```

当正则表达式中包含能接受重复的限定符时，通常的行为是（在使整个表达式能得到匹配的前提下）匹配尽可能多的字符。我们更需要懒惰匹配，也就是匹配尽可能少的字符。前面给出的限定符都可以被转化为懒惰匹配模式，只要在它后面加上一个问号?。这样.*?就意味着匹配任意数量的重复，但是在能使整个匹配成功的前提下使用最少的重复。

AJAX 同步和异步的区别？ajax 的交互流程？

同步的时候，当加载页面的时候，它会等待后台服务器响应，会打断用户的操作，电脑也会变白一会，而异步，则不打断用户操作，自行加载数据。

区别 onmouseover 和 mouseover

onmouseover 是 Event 对象的一个属性，Mouseover 是 jQuery 中的一个事件。

推荐使用 jQuery，直接执行方法$("#id").mouseover(function(){});完全使用 js 则是如下写法：document.getElementById("id").onmouseover=function(){};
document.getElementsByTagName("body")[0].style.backgroundColor="pink" ;
//注意不要忘了 style，深刻理解 DOM 的本质。

Javascript 的 addEventListener()及 attachEvent()区别分析

addEventListener()和 attachEvent()是一个侦听事件并处理相应的函数，可以动态的为网页内的元素添加一个或多个事件。可以将事件和元素分离，这样可编辑性就高了。

addEventListener 的使用方式：

target.addEventListener(type, listener, useCapture);

target：文档节点、document、window 或 XMLHttpRequest。

type：字符串，事件名称，不含“on”，比如“click”、“mouseover”、keydown”等。

listener ：实现了 EventListener 接口或者是 JavaScript 中的函数。

useCapture ：是否使用捕捉，一般用 false 。

例如：document.getElementById("testText").addEventListener("keydown", function (event) { alert(event.keyCode); }, false);

attachEvent()是

target.attachEvent(type, listener);

注意：attachEvent（）中的 type： 字符串，事件名称，含“on”，比如“onclick”、“onmouseover”、“onkeydown”等。

事件监听机制（冒泡和捕获）

事件捕获：事件从最上一级标签开始往下查找，直到捕获到事件目标(target)。

事件冒泡：事件从事件目标(target)开始，往上冒泡直到页面的最上一级标签。

假设一个元素 div，它有一个下级元素 p。 元素 这两个元素都绑定了 click 事件，如果用户点击了 p，它在 div 和 p 上都触发了 click 事件，那这两个事件处理程序哪个先执行呢？如 div 先触发，这就叫做事件捕获。如 p 先触发，这就叫做事件冒泡。IE 只支持事件冒泡，其他主流浏览器两种都支持。程序员可以自己选择绑定事件时采用事件捕获还是事件冒泡，方法就是绑定事件时通过 addEventListener 函数，它有三个参数，第三个参数若是 true，则表示采用事件捕获，若是 false，则表示采用事件冒泡。事件的传播是可以阻止的：在 W3c 中，使用 stopPropagation（）方法在捕获的过程中 stopPropagation（）；后，后面的冒泡过程也不会发生了 propagation 【传播，蔓延】阻止事件的默认行为，例如 click a 标签后的跳转在 W3c 中，使用 preventDefault（）方法；在 IE 下设置 window.event.returnValue = false;

如果给一个元素同时绑定两个事件，会怎么样？

Dom 0 级和 Dom 2 级都可以给一个元素添加多个事件，Dom 0 级的每个事件只支持一个事件处理程序，如果绑定同一个事件，那么后边的那个事件，函数会覆盖掉前边的那个事件函数。Dom2 级可以添加多个事件处理程序，他们会按照添加的顺序触发。

prototype 属性

每一个构造函数都有一个属性叫做原型(prototype)。这个属性非常有用：为一个特定类声明通用的变量或者函数。prototype 是一个对象，因此，你能够给它添加属性。你添加给 prototype 的属性将会成为使用这个构造函数创建的对象的通用属性。

js 事件委托

“事件处理程序过多”问题的解决方案就是事件委托。

事件委托利用的是事件冒泡机制，只制定一事件处理程序，就可以管理某一类型的所有事件（使用事件委托，只需在 DOM 树中尽量最高的层次上添加一个事件处理程序）。这里要用到事件源：event 对象，需要用到 target 属性，其 事件属性可返回事件的目标节点（触发该事件的节点）

```
oUl.onmouseover = function(ev){
  var target = ev.target
  if(target.nodeName.toLowerCase() == "li"){
    target.style.background = "red";
  }
}
```

回调函数

函数 a 有一个参数，这个参数是个函数 b，当函数 a 执行完以后执行函数 b。那么这个过程就叫回调。函数 b 是你以参数形式传给函数 a 的，那么函数 b 就叫回调函数。回调函数可以继续扩展一个函数的功能，可以是程序非常灵活。

```
function zy(callback){alert("开始");callback();}
function zygg(){alert("我是回调函数");}
function test(){zy(zygg)}
```

JavaScript 内置对象有以下几种。

String 对象：处理所有的字符串操作
Math 对象：处理所有的数学运算
Date 对象：处理日期和时间的存储、转化和表达
Array 对象：提供一个数组的模型、存储大量有序的数据
Event 对象：提供 JavaScript 事件的各种处理信息

JavaScript 内置函数

①escape() 函数可对字符串进行编码，这样就可以在所有的计算机上读取该字符串。eg:?=%3

②eval() 函数可计算某个字符串，并执行其中的的 JavaScript 代码。eg:eval("x=10;y=20;document.write(x*y)")

③isFinite() 函数用于检查其参数是否是无穷大。返回 true 或者 false。

④isNaN( )函数可用于判断其参数是否是 NaN

⑤parseFloat() 函数可解析一个字符串，并返回一个浮点数。

⑥parseInt() 函数可解析一个字符串，并返回一个整数。

⑦unescape() 函数可对通过 escape() 编码的字符串进行解码。

Unicode 和 ASCII 的区别是什么回答

计算机发明后，为了在计算机中表示字符，人们制定了一种编码，叫 ASCII 码。中国人利用连续 2 个扩展 ASCII 码的扩展区域（0xA0 以后）来表示一个汉字，该方法的标准叫 GB-2312。因为各个国家地区定义的字符集有交集，因此使用 GB-2312 的软件，就不能在 BIG-5 的环境下运行（显示乱码），反之亦然。

为了把全世界人民所有的所有的文字符号都统一进行编码，于是制定了 UNICODE 标准字符集。UNICODE 使用 2 个字节表示一个字符(unsigned shor int、WCHAR、_wchar_t、OLECHAR)。

嵌套路由怎么定义？

在实际项目中我们会碰到多层嵌套的组件组合而成，但是我们如何实现嵌套路由呢？因此我们需要在 VueRouter 的参数中使用 children 配置，这样就可以很好的实现路由嵌套。

index.html，只有一个路由出口

```
<div id="app">
      <!-- router-view 路由出口，路由匹配到的组件将渲染在这里 -->
      <router-view></router-view>
</div>
```

main.js，路由的重定向，就会在页面一加载的时候，就会将 home 组件显示出来，因为重定向指向了 home 组件，redirect 的指向与 path 的必须一致。children 里面是子路由，当然子路由里面还可以继续嵌套子路由。

怎么定义 vue-router 的动态路由？怎么获取传过来的动态参数？

在 router 目录下的 index.js 文件中，对 path 属性加上/:id。使用 router 对象的 params.id。

vue-router 有哪几种导航钩子？

第一种：是全局导航钩子：router.beforeEach(to,from,next)，作用：跳转前进行判断拦截。

第二种：组件内的钩子

第三种：单独路由独享组件

scss 是什么？在 vue.cli 中的安装使用步骤是？有哪几大特性？

css 的预编译。

使用步骤：

第一步：用 npm 下三个 loader（sass-loader、css-loader、node-sass）

第二步：在 build 目录找到 webpack.base.config.js，在那个 extends 属性中加一个拓展.scss

第三步：还是在同一个文件，配置一个 module 属性

第四步：然后在组件的 style 标签加上 lang 属性 ，例如：lang=”scss”

有哪几大特性：

1、可以用变量，例如（$变量名称=值）；

2、可以用混合器，例如（）

3、可以嵌套

mint-ui 是什么？怎么使用？说出至少三个组件使用方法？

基于 vue 的前端组件库。

npm 安装，然后 import 样式和 js，vue.use（mintUi）全局引入。在单个组件局部引入：import {Toast} from ‘mint-ui’。

组件一：Toast(‘登录成功’);

组件二：mint-header；
组件三：mint-swiper

v-model 是什么？怎么使用？ vue 中标签怎么绑定事件？

可以实现双向绑定，指令（v-class、v-for、v-if、v-show、v-on）。vue 的 model 层的 data 属性。绑定事件：<input @click=doLog()/>

iframe 的优缺点？

iframe 也称作嵌入式框架，嵌入式框架和框架网页类似，它可以把一个网页的框架和内容嵌入在现有的网页中。

优点：

解决加载缓慢的第三方内容如图标和广告等的加载问题
Security sandbox
并行加载脚本
方便制作导航栏

缺点：

iframe 会阻塞主页面的 Onload 事件
即时内容为空，加载也需要时间
没有语意

简述一下 Sass、Less，且说明区别？

他们是动态的样式语言，是 CSS 预处理器,CSS 上的一种抽象层。他们是一种特殊的语法/语言而编译成 CSS。

变量符不一样，less 是@，而 Sass 是$;

Sass 支持条件语句，可以使用 if{}else{},for{}循环等等。而 Less 不支持;

Sass 是基于 Ruby 的，是在服务端处理的，而 Less 是需要引入 less.js 来处理 Less 代码输出 Css 到浏览器

axios 是什么？怎么使用？描述使用它实现登录功能的流程？

请求后台资源的模块。

npm install axios -S 装好，然后发送的是跨域，需在配置文件中 config/index.js 进行设置。后台如果是 Tp5 则定义一个资源路由。

js 中使用 import 进来，然后.get 或.post。返回在.then 函数中如果成功，失败则是在.catch 函数中

axios+tp5 进阶中，调用 axios.post（‘api/user’）是进行的什么操作？

axios.put（‘api/user/8’)呢？

跨域，添加用户操作，更新操作。

vuex 是什么？怎么使用？哪种功能场景使用它？

vue 框架中状态管理。

在 main.js 引入 store，注入。新建了一个目录 store，….. export 。

场景有：单页应用中，组件之间的状态。音乐播放、登录状态、加入购物车

mvvm 框架是什么？它和其它框架（jquery）的区别是什么？哪些场景适合？

一个 model+view+viewModel 框架，数据模型 model，viewModel 连接两个
区别：vue 数据驱动，通过数据来显示视图层而不是节点操作。
场景：数据操作比较多的场景，更加便捷

自定义指令（v-check、v-focus）的方法有哪些？它有哪些钩子函数？还有哪些钩子函数参数？

全局定义指令：在 vue 对象的 directive 方法里面有两个参数，一个是指令名称，另外一个是函数。组件内定义指令：directives
钩子函数：bind（绑定事件触发）、inserted(节点插入的时候触发)、update（组件内相关更新）
钩子函数参数：el、binding

说出至少 4 种 vue 当中的指令和它的用法？

v-if：判断是否隐藏；
v-for：数据循环出来；
v-bind:class：绑定一个属性；
v-model：实现双向绑定

vue-router 是什么？它有哪些组件？

vue 用来写路由一个插件。router-link、router-view

导航钩子有哪些？它们有哪些参数？

导航钩子有：
全局钩子和组件内独享的钩子。
beforeRouteEnter、afterEnter、beforeRouterUpdate、beforeRouteLeave
参数：
有 to（去的那个路由）、from（离开的路由）、next（一定要用这个函数才能去到下一个路由，如果不用就拦截）最常用就这几种

Vue 的双向数据绑定原理是什么？

vue.js 是采用数据劫持结合发布者-订阅者模式的方式，通过 Object.defineProperty()来劫持各个属性的 setter，getter，在数据变动时发布消息给订阅者，触发相应的监听回调。
具体步骤：
第一步：需要 observe 的数据对象进行递归遍历，包括子属性对象的属性，都加上 setter 和 getter
这样的话，给这个对象的某个值赋值，就会触发 setter，那么就能监听到了数据变化
第二步：compile 解析模板指令，将模板中的变量替换成数据，然后初始化渲染页面视图，并将每个指令对应的节点绑定更新函数，添加监听数据的订阅者，一旦数据有变动，收到通知，更新视图

第三步：Watcher 订阅者是 Observer 和 Compile 之间通信的桥梁，主要做的事情是:
1、在自身实例化时往属性订阅器(dep)里面添加自己
2、自身必须有一个 update()方法
3、待属性变动 dep.notice()通知时，能调用自身的 update()方法，并触发 Compile 中绑定的回调，则功成身退。

第四步：MVVM 作为数据绑定的入口，整合 Observer、Compile 和 Watcher 三者，通过 Observer 来监听自己的 model 数据变化，通过 Compile 来解析编译模板指令，最终利用 Watcher 搭起 Observer 和 Compile 之间的通信桥梁，达到数据变化 -> 视图更新；视图交互变化(input) -> 数据 model 变更的双向绑定效果。

请说下封装 vue 组件的过程？

首先，组件可以提升整个项目的开发效率。能够把页面抽象成多个相对独立的模块，解决了我们传统项目开发：效率低、难维护、复用性等问题。

然后，使用 Vue.extend 方法创建一个组件，然后使用 Vue.component 方法注册组件。子组件需要数据，可以在 props 中接受定义。而子组件修改好数据后，想把数据传递给父组件。可以采用 emit 方法。

你是怎么认识 vuex 的？

vuex 可以理解为一种开发模式或框架。比如 PHP 有 thinkphp，java 有 spring 等。通过状态（数据源）集中管理驱动组件的变化（好比 spring 的 IOC 容器对 bean 进行集中管理）。

应用级的状态集中放在 store 中； 改变状态的方式是提交 mutations，这是个同步的事物； 异步逻辑应该封装在 action 中。

vue-loader 是什么？使用它的用途有哪些？

解析.vue 文件的一个加载器，跟 template/js/style 转换成 js 模块。

用途：js 可以写 es6、style 样式可以 scss 或 less、template 可以加 jade 等

请说出 vue.cli 项目中 src 目录每个文件夹和文件的用法？

assets 文件夹是放静态资源；components 是放组件；router 是定义路由相关的配置;view 视图；app.vue 是一个应用主组件；main.js 是入口文件

vue.cli 中怎样使用自定义的组件？有遇到过哪些问题吗？

第一步：在 components 目录新建你的组件文件（smithButton.vue），script 一定要 export default {

第二步：在需要用的页面（组件）中导入：import smithButton from '../components/smithButton.vue'

第三步：注入到 vue 的子组件的 components 属性上面,components:{smithButton}

第四步：在 template 视图 view 中使用，<smith-button> </smith-button>

问题有：smithButton 命名，使用的时候则 smith-button。

聊聊你对 Vue.js 的 template 编译的理解？

简而言之，就是先转化成 AST 树，再得到的 render 函数返回 VNode（Vue 的虚拟 DOM 节点）

详情步骤：

首先，通过 compile 编译器把 template 编译成 AST 语法树（abstract syntax tree 即 源代码的抽象语法结构的树状表现形式），compile 是 createCompiler 的返回值，createCompiler 是用以创建编译器的。另外 compile 还负责合并 option。

然后，AST 会经过 generate（将 AST 语法树转化成 render funtion 字符串的过程）得到 render 函数，render 的返回值是 VNode，VNode 是 Vue 的虚拟 DOM 节点，里面有（标签名、子节点、文本等等）

vue 的历史记录

history 记录中向前或者后退多少步

vuejs 与 angularjs 以及 react 的区别？

1.与 AngularJS 的区别

相同点：

都支持指令：内置指令和自定义指令。

都支持过滤器：内置过滤器和自定义过滤器。

都支持双向数据绑定。

都不支持低端浏览器。

不同点：

1.AngularJS 的学习成本高，比如增加了 Dependency Injection 特性，而 Vue.js 本身提供的 API 都比较简单、直观。

2.在性能上，AngularJS 依赖对数据做脏检查，所以 Watcher 越多越慢。Vue.js 使用基于依赖追踪的观察并且使用异步队列更新。所有的数据都是独立触发的。对于庞大的应用来说，这个优化差异还是比较明显的。

2.与 React 的区别

相同点：

React 采用特殊的 JSX 语法，Vue.js 在组件开发中也推崇编写.vue 特殊文件格式，对文件内容都有一些约定，两者都需要编译后使用。

中心思想相同：一切都是组件，组件实例之间可以嵌套。

都提供合理的钩子函数，可以让开发者定制化地去处理需求。

都不内置列数 AJAX，Route 等功能到核心包，而是以插件的方式加载。

在组件开发中都支持 mixins 的特性。

不同点：

React 依赖 Virtual DOM,而 Vue.js 使用的是 DOM 模板。React 采用的 Virtual DOM 会对渲染出来的结果做脏检查。

Vue.js 在模板中提供了指令，过滤器等，可以非常方便，快捷地操作 DOM。

什么是 vue 生命周期？

Vue 实例从创建到销毁的过程，就是生命周期。也就是从开始创建、初始化数据、编译模板、挂载 Dom→渲染、更新→渲染、卸载等一系列过程，我们称这是 Vue 的生命周期。

vue 生命周期的作用是什么？

它的生命周期中有多个事件钩子，让我们在控制整个 Vue 实例的过程时更容易形成好的逻辑。

请详细说下你对 vue 生命周期的理解？

总共分为 8 个阶段创建前/后，载入前/后，更新前/后，销毁前/后

创建前/后： 在 beforeCreated 阶段，vue 实例的挂载元素$el 和数据对象 data 都为 undefined，还未初始化。在 created 阶段，vue 实例的数据对象 data 有了，$el 还没有。

载入前/后：在 beforeMount 阶段，vue 实例的$el 和 data 都初始化了，但还是挂载之前为虚拟的 dom 节点，data.message 还未替换。在 mounted 阶段，vue 实例挂载完成，data.message 成功渲染。

更新前/后：当 data 变化时，会触发 beforeUpdate 和 updated 方法。

销毁前/后：在执行 destroy 方法后，对 data 的改变不会再触发周期函数，说明此时 vue 实例已经解除了事件监听以及和 dom 的绑定，但是 dom 结构依然存在

第一次页面加载会触发哪几个钩子？

第一次页面加载时会触发 beforeCreate, created, beforeMount, mounted 这几个钩子

DOM 渲染在 哪个周期中就已经完成？

DOM 渲染在 mounted 中就已经完成了

简单描述每个周期具体适合哪些场景？

生命周期钩子的一些使用方法： beforecreate ： 可以在这加个 loading 事件，在加载实例时触发 created ： 初始化完成时的事件写在这里，如在这结束 loading 事件，异步请求也适宜在这里调用 mounted ： 挂载元素，获取到 DOM 节点 updated ： 如果对数据统一处理，在这里写上相应函数 beforeDestroy ： 可以做一个确认停止事件的确认框 nextTick ： 更新数据后立即操作 dom

arguments 是一个伪数组，没有遍历接口，不能遍历

cancas 和 SVG 的是什么以及区别

SVG

SVG 是一种使用 XML 描述 2D 图形的语言。

SVG 基于 XML，这意味着 SVG DOM 中的每个元素都是可用的。您可以为某个元素附加 JavaScript 事件处理器。

在 SVG 中，每个被绘制的图形均被视为对象。如果 SVG 对象的属性发生变化，那么浏览器能够自动重现图形。

Canvas

Canvas 通过 JavaScript 来绘制 2D 图形。

Canvas 是逐像素进行渲染的。

在 canvas 中，一旦图形被绘制完成，它就不会继续得到浏览器的关注。如果其位置发生变化，那么整个场景也需要重新绘制，包括任何或许已被图形覆盖的对象。

Canvas 与 SVG 的比较

Canvas
依赖分辨率
不支持事件处理器
弱的文本渲染能力
能够以 .png 或 .jpg 格式保存结果图像
最适合图像密集型的游戏，其中的许多对象会被频繁重绘
SVG
不依赖分辨率
支持事件处理器
最适合带有大型渲染区域的应用程序（比如谷歌地图）
复杂度高会减慢渲染速度（任何过度使用 DOM 的应用都不快）
不适合游戏应用

PHP 中 include 和 include_once 的区别？

include_once()语句的语法和 include()语句类似，主要区别也是避免多次包含一个文件而引起函数或变量的重复定义。include_once() 语句在脚本执行期间包含并运行指定文件。include_once 语句把 include 的功能扩展了。在程序执行期间，将指定的文件包含进来，如果从文件引用进来的程序先前已经包含过的时候，include_once()就不会把它再包含进来。也就是仅仅可以引用同一个文件一次！

雅虎的 14 条优化规则？

1. 尽可能的减少 HTTP 的请求数 content
2. 使用 CDN（Content Delivery Network） server
3. 添加 Expires 头(或者 Cache-control) server
4. Gzip 组件 server
5. 将 CSS 样式放在页面的上方 css
6. 将脚本移动到底部（包括内联的） javascript
7. 避免使用 CSS 中的 Expressions css
8. 将 JavaScript 和 CSS 独立成外部文件
9. 减少 DNS 查询 content
10. 压缩 JavaScript 和 CSS (包括内联的)
11. 避免重定向 server
12. 移除重复的脚本 javascript
13. 配置实体标签（ETags） css
14. 使 AJAX 缓存

让你来制作一个访问量很高的大网站，你会如何来管理所有 CSS 文件、JS 与图片？

1、css 文件，以及 js 文件尽量分别都放在一个文件里，因为客户端请求服务器的次数就会减少。

2、背景图尽量采用聚合技术，就是放在一个图片里，用 background-position 来定位；

3、css 文件里尽量都精简一些，比如说#sidebarcontent{}啥的，我们直接可以用#s-c{}因为这样整个文件的容量就会减少 ，同样的原理，在线也可以压缩 js 文件。容量变小些嘛；

HTTP 协议的状态消息都有哪些?(如 200、302 对应的描述)国内外的 JS 牛人都知道哪些?

协议是指计算机通信网络中两台计算机之间进行通信所必须共同遵守的规定或规则，超文本传输协议(HTTP)是一种通信协议，它允许将超文本标记语言(HTML)文档从 Web 服务器传送到客户端的浏览器，

“100” ：Continue（继续） 初始的请求已经接受，客户应当继续发送请求的其余部分。（HTTP 1.1 新）

“101” ：Switching Protocols（切换协议） 请求者已要求服务器切换协议，服务器已确认并准备进行切换。（HTTP 1.1 新）

“200” ：OK（成功） 一切正常，对 GET 和 POST 请求的应答文档跟在后面。

“201” ：Created（已创建）服务器已经创建了文档，Location 头给出了它的 URL。

“202” ：Accepted（已接受）服务器已接受了请求，但尚未对其进行处理。

“203” ：Non-Authoritative Information（非授权信息） 文档已经正常地返回，但一些应答头可能不正确，可能来自另一来源 。（HTTP 1.1 新）。

“204” ：No Content（无内容）未返回任何内容，浏览器应该继续显示原来的文档。

“205” ：Reset Content（重置内容）没有新的内容，但浏览器应该重置它所显示的内容。用来强制浏览器清除表单输入内容（HTTP 1.1 新）。

“206” ：Partial Content（部分内容）服务器成功处理了部分 GET 请求。（HTTP 1.1 新）

“300” ：Multiple Choices（多种选择）客户请求的文档可以在多个位置找到，这些位置已经在返回的文档内列出。如果服务器要提出优先选择，则应该在 Location 应答头指明。

“301” ：Moved Permanently（永久移动）请求的网页已被永久移动到新位置。服务器返回此响应（作为对 GET 或 HEAD 请求的响应）时，会自动将请求者转到新位置。

“302” ：Found（临时移动）类似于 301，但新的 URL 应该被视为临时性的替代，而不是永久性的。注意，在 HTTP1.0 中对应的状态信息是“Moved Temporatily”，出现该状态代码时，浏览器能够自动访问新的 URL，因此它是一个很有用的状态代码。注意这个状态代码有时候可以和 301 替换使用。例如，如果浏览器错误地请求 http://host/~user（缺少了后面的斜杠），有的服务器返回 301，有的则返回 302。严格地说，我们只能假定只有当原来的请求是 GET 时浏览器才会自动重定向。请参见 307。

“303” ：See Other（查看其他位置）类似于 301/302，不同之处在于，如果原来的请求是 POST，Location 头指定的重定向目标文档应该通过 GET 提取（HTTP 1.1 新）。

“304” ：Not Modified（未修改）自从上次请求后，请求的网页未被修改过。原来缓冲的文档还可以继续使用，不会返回网页内容。

“305” ：Use Proxy（使用代理）只能使用代理访问请求的网页。如果服务器返回此响应，那么，服务器还会指明请求者应当使用的代理。（HTTP 1.1 新）

“307” ：Temporary Redirect（临时重定向）和 302（Found）相同。许多浏览器会错误地响应 302 应答进行重定向，即使原来的请求是 POST，即使它实际上只能在 POST 请求的应答是 303 时才能重定向。由于这个原因，HTTP 1.1 新增了 307，以便更加清除地区分几个状态代码：当出现 303 应答时，浏览器可以跟随重定向的 GET 和 POST 请求；如果是 307 应答，则浏览器只能跟随对 GET 请求的重定向。（HTTP 1.1 新）

“400” ：Bad Request（错误请求）请求出现语法错误。

“401” ：Unauthorized（未授权）客户试图未经授权访问受密码保护的页面。应答中会包含一个 WWW-Authenticate 头，浏览器据此显示用户名字/密码对话框，然后在填写合适的 Authorization 头后再次发出请求。

“403” ：Forbidden（已禁止） 资源不可用。服务器理解客户的请求，但拒绝处理它。通常由于服务器上文件或目录的权限设置导致。

“404” ：Not Found（未找到）无法找到指定位置的资源。

“405” ：Method Not Allowed（方法禁用）请求方法（GET、POST、HEAD、DELETE、PUT、TRACE 等）禁用。（HTTP 1.1 新）

“406” ：Not Acceptable（不接受）指定的资源已经找到，但它的 MIME 类型和客户在 Accpet 头中所指定的不兼容（HTTP 1.1 新）。

“407” ：Proxy Authentication Required（需要代理授权）类似于 401，表示客户必须先经过代理服务器的授权。（HTTP 1.1 新）

“408” ：Request Time-out（请求超时）服务器等候请求时超时。（HTTP 1.1 新）

“409” ：Conflict（冲突）通常和 PUT 请求有关。由于请求和资源的当前状态相冲突，因此请求不能成功。（HTTP 1.1 新）

“410” ：Gone（已删除）如果请求的资源已被永久删除，那么，服务器会返回此响应。该代码与 404（未找到）代码类似，但在资源以前有但现在已经不复存在的情况下，有时会替代 404 代码出现。如果资源已被永久删除，那么，您应当使用 301 代码指定该资源的新位置。（HTTP 1.1 新）

“411” ：Length Required（需要有效长度）不会接受包含无效内容长度标头字段的请求。（HTTP 1.1 新）

“412” ：Precondition Failed（未满足前提条件）服务器未满足请求者在请求中设置的其中一个前提条件。（HTTP 1.1 新）

“413” ：Request Entity Too Large（请求实体过大）请求实体过大，已超出服务器的处理能力。如果服务器认为自己能够稍后再处理该请求，则应该提供一个 Retry-After 头。（HTTP 1.1 新）

“414” ：Request-URI Too Large（请求的 URI 过长）请求的 URI（通常为网址）过长，服务器无法进行处理。

“415” ：Unsupported Media Type（不支持的媒体类型）请求的格式不受请求页面的支持。

“416” ：Requested range not satisfiable（请求范围不符合要求）服务器不能满足客户在请求中指定的 Range 头。（HTTP 1.1 新）

“417” ：Expectation Failed（未满足期望值）服务器未满足”期望”请求标头字段的要求。

“500” ：Internal Server Error（服务器内部错误）服务器遇到错误，无法完成请求。

“501” ：Not Implemented（尚未实施） 服务器不具备完成请求的功能。例如，当服务器无法识别请求方法时，服务器可能会返回此代码。

“502” ：Bad Gateway（错误网关）服务器作为网关或者代理时，为了完成请求访问下一个服务器，但该服务器返回了非法的应答。

“503” ：Service Unavailable（服务不可用）服务器由于维护或者负载过重未能应答。通常，这只是一种暂时的状态。

“504” ：Gateway Time-out（网关超时） 由作为代理或网关的服务器使用，表示不能及时地从远程服务器获得应答。（HTTP 1.1 新）

“505” ：HTTP Version not supported（HTTP 版本不受支持）不支持请求中所使用的 HTTP 协议版本。

WEB 前端的开发工具？

Yslow，雅虎开发工具，判断网页里哪个运行的慢，基于网页查找。
Firebug：火狐开发工具，也用过一段时间，挺好用的，只是不习惯。
Chrome：谷歌开发工具，比较适用于移动端和网页的调试。

常见 ie6 的浏览器兼容 bug（3-5 个）？

1.文字本身的大小不兼容。同样是 font-size:14px 的宋体文字，在不同浏览器下占的空间是不一样的，ie 下实际占高 16px，下留白 3px，ff 下实际占高 17px，上留白 1px，下留白 3px，opera 下就更不一样了。解决方案：给文字设定 line-height 。确保所有文字都有默认的 line-height 值。

2.ff 下容器高度限定，即容器定义了 height 之后，容器边框的外形就确定了，不会被内容撑大，而 ie6 下是会被内容撑大，高度限定失效,ie7,8,9 都不会撑大。所以不要轻易给容器定义 height。解决方案用：height! important；min-height：100px；max-height： 200px

3.横向上的撑破容器问题。如果 float 容器未定义宽度，ff 下内容会尽可能撑开容器宽度，ie 下则会优先考虑内容折行。故内容可能撑破的浮动容器需要定义 width。

4.最被痛恨的，double-margin bug。ie6 下给浮动容器定义 margin-left 或者 margin-right 实际效果是数值的 2 倍。解决方案，给浮动容器定义 display:inline。

5. 吞吃现象。还是 ie6，上下两个 div，上面的 div 设置背景，却发现下面没有设置背景的 div 也有了背景，这就是吞吃现象。对应上面的背景吞吃现象，还有滚动下边框缺失的现象。解决方案：使用 zoom:1。这个 zoom 好象是专门为解决 ie6 bug 而生的。

6.注释也能产生 bug“多出来的一只。”这是前人总结这个 bug 使用的文案，ie6 的这个 bug 下，大家会在页面看到字出现两遍，重复的内容量因注释的多少而变。解决方案：用“<!-[if !IE]> picRotate start <![endif]->”方法写注释。

7.img 下的留白，如下代码： 把 div 的 border 打开，你发现图片底部不是紧贴着容器底部的，是 img 后面的空白字符造成，要消除必须这样写 后面两个标签要紧挨着。ie7 下这个 bug 依然存在。解决方案：给 img 设定 display:block。

8. 失去 line-height。 <img />文字 ，很遗憾，在 ie6 下单行文字 line-height 效果消失了，原因是<img />这个 inline-block 元素和 inline 元素写在一起了。解决方案：让 img 和文字都 float 起来。

引申：大家知道 img 的 align 有 text-top，middle，absmiddle 啊什么的，你可以尝试去调整 img 和文字让他们在 ie 和 ff 下能一致，你会发现怎么调都不会让你满意。索性让 img 和文字都 float 起来，用 margin 调整。

9.clear 层应该单独使用。也许你为了节省代码把 clear 属性直接放到下面的一个内容层，这样有问题，不仅仅是 ff 和 op 下失去 margin 效果，ie 下某些 margin 值也会失效 dd ff

10.ie 下 overflow:hidden 对其下的绝对层 position:absolute 或者相对层 position:relative 无效。解决方案：给 overflow:hidden 加 position:relative 或者 position: absolute。另，ie6 支持 overflow-x 或者 overflow-y 的特性，ie7、ff 不支持。

11.ie6 下严重的 bug，float 元素如没定义宽度，内部如有 div 定义了 height 或 zoom:1，这个 div 就会占满一整行，即使你给了宽度。float 元素如果作为布局用或复杂的容器，都要给个宽度的

12.浮动元素之后跟着一个元素之间的有 3 像素的差距？解决方案：浮动的元素：overflow：hidden；后面的元素设置_margin-left：-3px，

SQL 是什么？

SQL 是用于访问和处理数据库的标准的计算机语言。
SQL 指结构化查询语言
SQL 使我们有能力访问数据库
SQL 是一种 ANSI 的标准计算机语言
SQL 是一门 ANSI 的标准计算机语言，用来访问和操作数据库系统。SQL 语句用于取回和更新数据库中的数据。

SQL 能做什么？

SQL 面向数据库执行查询
SQL 可从数据库取回数据
SQL 可在数据库中插入新的记录
SQL 可更新数据库中的数据
SQL 可从数据库删除记录
SQL 可创建新数据库
SQL 可在数据库中创建新表
SQL 可在数据库中创建存储过程
SQL 可在数据库中创建视图
SQL 可以设置表、存储过程和视图的权限

PHP 的意义？

PHP 是一种创建动态交互性站点的强有力的服务器端脚本语言。
PHP 是免费的，并且使用非常广泛。同时，对于像微软 ASP 这样的竞争者来说，PHP 无疑是另
一种高效率的选项。PHP 极其适合网站开发，其代码可以直接嵌入 HTML 代码。
PHP 语法非常类似于 Perl 和 C。PHP 常常搭配 Apache (web 服务器) 一起使用。不过它也
支持 ISAPI，并且可以运行于 Windows 的微软 IIS 平台。

什么是 PHP？

PHP 指 PHP:超文本预处理器（译者注：PHP: Hypertext Preprocessor，递归命名）
PHP 是一种服务器端的脚本语言，类似 ASP
PHP 脚本在服务器上执行
PHP 支持很多数据库（MySQL、Informix、Oracle、Sybase、Solid、PostgreSQL、Generic ODBC 等等）
PHP 是一个开源的软件（open source software，OSS）
PHP 可免费下载使用

什么是 MySQL？

MySQL 是一种数据库服务器
MySQL 支持标准的 SQL
MySQL 可在许多平台上编译
MySQL 可免费下载使用

PHP + MySQL

PHP 与 MySQL 的组合是跨平台的（意思是您可以在 Windows 环境进行开发，而在 Unix 平台上提供服务）

为什么要使用 PHP？

PHP 可在不同的平台上运行（Windows、Linux、Unix）
PHP 与目前几乎所有的正在被使用的服务器相兼容（Apache、IIS 等）
PHP 可从官方的 PHP 资源免费下载：www.php.net
PHP 易于学习，并可高效地运行在服务器端

DNS 的工作原理（递归和迭代）（应用层）

DNS 的工作原理及过程分下面几个步骤：
第一步：客户机提出域名解析请求，并将该请求发送给本地的域名服务器。
第二步：当本地的域名服务器收到请求后，就先查询本地的缓存，如果有该纪录项，则本地的域名服务器就直接把查询的结果返回。
第三步：如果本地的缓存中没有该纪录，则本地域名服务器就直接把请求发给根域名服务器，然后根域名服务器再返回给本地域名服务器一个所查询域(根的子域) 的主域名服务器的地址。
第四步：本地服务器再向上一步返回的域名服务器发送请求，然后接受请求的服务器查询自己的缓存，如果没有该纪录，则返回相关的下级的域名服务器的地址。
第五步：重复第四步，直到找到正确的纪录。
第六步：本地域名服务器把返回的结果保存到缓存，以备下一次使用，同时还将结果返回给客户机。

从输入 URL 到页面加载完的过程中都发生了什么事情？

①首先如果我们如果输入的不是 ip 地址，而是域名的话，就需要 IP 解析，DNS 域名解析（具体见 DNS 工作机制）。
②解析出来对应的 IP 后，如不包含端口号，http 协议默认端口号是 80；https（http+ssl（传输层））是 430！然后向 IP 发起网络连接，根据 http 协议要求，组织一个请求的数据包，里面包含大量请求信息。
③服务器响应请求，将数据返回给浏览器。数据可能是根据 HTML 协议组织的网页，里面包含页面的布局、文字。数据也可能是图片、脚本程序等。
④开始根据资源的类型，将资源组织成屏幕上显示的图像，这个过程叫渲染，网页渲染是浏览器最复杂、最核心的功能。
⑤将渲染好的页面图像显示出来，并开始响应用户的操作。