开发及性能优化

1、规避javascript多人开发函数重名问题

命名空间

封闭空间

js模块化mvc（数据层、表现层、控制层）

seajs

变量转换成对象的属性

对象化

2、请说出三种减低页面加载时间的方法

压缩css、js文件

合并js、css文件，减少http请求

外部js、css文件放在最底下

减少dom操作，尽可能用变量替代不必要的dom操作

3、你所了解到的Web攻击技术

（1）XSS（Cross-Site Scripting，跨站脚本攻击）：指通过存在安全漏洞的Web网站注册用户的浏览器内运行非法的HTML标签或者JavaScript进行的一种攻击。  
 （2）SQL注入攻击  
 （3）CSRF（Cross-Site Request Forgeries，跨站点请求伪造）：指攻击者通过设置好的陷阱，强制对已完成的认证用户进行非预期的个人信息或设定信息等某些状态更新。

4、web前端开发，如何提高页面性能优化？

（1）内容方面：

1.减少 HTTP 请求 (Make Fewer HTTP Requests)

2.减少 DOM 元素数量 (Reduce the Number of DOM Elements)

3.使得 Ajax 可缓存 (Make Ajax Cacheable)

（2）针对CSS：

1.把 CSS 放到代码页上端 (Put Stylesheets at the Top)

2.从页面中剥离 JavaScript 与 CSS (Make JavaScript and CSS External)

3.精简 JavaScript 与 CSS (Minify JavaScript and CSS)

4.避免 CSS 表达式 (Avoid CSS Expressions)

（3）针对JavaScript ：

1. 脚本放到 HTML 代码页底部 (Put Scripts at the Bottom)

2. 从页面中剥离 JavaScript 与 CSS (Make JavaScript and CSS External)

3. 精简 JavaScript 与 CSS (Minify JavaScript and CSS)

4. 移除重复脚本 (Remove Duplicate Scripts)

（4）面向图片(Image)：

1.优化图片

2 不要在 HTML 中使用缩放图片

3 使用恰当的图片格式

4 使用 CSS Sprites 技巧对图片优化

5、前端开发中，如何优化图像？

1、不用图片，尽量用css3代替。 比如说要实现修饰效果，如半透明、边框、圆角、阴影、渐变等，在当前主流浏览器中都可以用CSS达成。

2、 使用矢量图SVG替代位图。对于绝大多数图案、图标等，矢量图更小，且可缩放而无需生成多套图。现在主流浏览器都支持SVG了，所以可放心使用！

3.、使用恰当的图片格式。我们常见的图片格式有JPEG、GIF、PNG。

基本上，内容图片多为照片之类的，适用于JPEG。

而修饰图片通常更适合用无损压缩的PNG。

GIF基本上除了GIF动画外不要使用。且动画的话，也更建议用video元素和视频格式，或用SVG动画取代。

4、按照HTTP协议设置合理的缓存。

5、使用字体图标webfont、CSS Sprites等。

6、用CSS或JavaScript实现预加载。

7、WebP图片格式能给前端带来的优化。WebP支持无损、有损压缩，动态、静态图片，压缩比率优于GIF、JPEG、JPEG2000、PG等格式，非常适合用于网络等图片传输。

6、图像格式的区别

矢量图：图标字体，如 font-awesome；svg

位图：gif, jpg(jpeg), png

1、gif:是是一种无损，8位图片格式。具有支持动画，索引透明，压缩等特性。适用于做色彩简单(色调少)的图片，如logo,各种小图标icons等。

2、JPEG格式是一种大小与质量相平衡的压缩图片格式。适用于允许轻微失真的色彩丰富的照片，不适合做色彩简单(色调少)的图片，如logo,各种小图标icons等。

3、png:PNG可以细分为三种格式:PNG8，PNG24，PNG32。后面的数字代表这种PNG格式最多可以索引和存储的颜色值。

关于透明：PNG8支持索引透明和alpha透明;PNG24不支持透明;而PNG32在24位的PNG基础上增加了8位（256阶）的alpha通道透明;

优缺点：

1、能在保证最不失真的情况下尽可能压缩图像文件的大小。

2、对于需要高保真的较复杂的图像，PNG虽然能无损压缩，但图片文件较大，不适合应用在Web页面上。

7、浏览器是如何渲染页面的？

1.解析HTML文件，创建DOM树。

   自上而下，遇到任何样式（link、style）与脚本（script）都会阻塞（外部样式不阻塞后续外部脚本的加载）。

2.解析CSS。优先级：浏览器默认设置<用户设置<外部样式<内联样式<HTML中的style样式；

3.将CSS与DOM合并，构建渲染树（Render Tree）

4.布局和绘制，重绘（repaint）和重排（reflow）