**HTML&CSS：**

**对Web标准的理解、浏览器内核差异、兼容性、hack、CSS基本功：布局、盒子模型、选择器优先级、**

**HTML5、CSS3、Flexbox**

**JavaScript：**

**数据类型、运算、对象、Function、继承、闭包、作用域、原型链、事件、RegExp、JSON、Ajax、**

**DOM、BOM、内存泄漏、跨域、异步装载、模板引擎、前端MVC、路由、模块化、Canvas、ECMAScript 6、Nodejs**

**其他：**

**移动端、响应式、自动化构建、HTTP、离线存储、WEB安全、优化、重构、团队协作、可维护、易用性、SEO、UED、架构、职业生涯、快速学习能力**

**From Web**

**原文参考链接：**

**<https://github.com/markyun/My-blog/tree/master/Front-end-Developer-Questions/Questions-and-Answers>**

**1、DOM结构 —— 两个节点之间可能存在哪些关系以及如何在节点之间任意移动。**

1. **DOM操作 —— 如何添加、移除、移动、复制、创建和查找节点等。**

Javascript

创建节点：document.createElement(); document.createTextNode()

添加节点: parentNode.appendChild(); element.innerHTML = ; element.textNode = ‘’, parentElement.insertBefore(newElement, referenceElement)

移除节点：ele.removeChild(),

替换节点: ele.replaceChild(ele1, ele2);

复制节点：node.cloneNode(true) // true指的是克隆时连其子元素一并复制。此方法不会拷贝事件处理程序。

查找节点：

jQuery:

clone(true) 连事件处理程序一并复制

remove(), detach(), empty(), remove和empty 都会移除事件，empty溢出的是后代元素，detach不会移除事件句柄，可能会造成内存泄漏。

replaceAll(), replaceWith(),

append, appendTo, prepend, prependTo, before, after,insertAfter,insertBefore

wrap, wrapAll, unwrap 包裹元素

3、事件 —— 如何使用事件，以及IE和标准DOM事件模型之间存在的差别。

IE事件模型：捕获阶段 document -> 目标元素 attachEvent() <IE9， this指向window

DOM事件：冒泡阶段 目标元素 -> document （文本节点也会触发事件）

addEventListener(‘Event’, function(){}, useCapture)

This 值指向触发事件的元素。

优点： 可以注册多个事件，支持自定义事件，可以选择在捕获还是冒泡阶段触发事件。

4、XMLHttpRequest —— 这是什么、怎样完整地执行一次GET请求、怎样检测错误。

Var req = new XMLHttpRequest(); // 对象

Req.open(string method,string url, [bool async], [bool user], [bool password]) //建立连接

Req.send() // 发送请求

Req.onreadystatechange = function(){

If(req.readyState = 4){ // loaded

If(req.status == 200){

// success.

}

}}

5、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。

6、盒模型 —— 外边距、内边距和边框之间的关系，及IE8以下版本的浏览器中的盒模型

7、块级元素与行内元素 —— 怎么用CSS控制它们、以及如何合理的使用它们

8、浮动元素 —— 怎么使用它们、它们有什么问题以及怎么解决这些问题。

9、HTML与XHTML —— 二者有什么区别，你觉得应该使用哪一个并说出理由。

1. JSON —— 作用、用途、设计结构。

**HTML**

1、Doctype作用？标准模式与兼容模式各有什么区别？

2、link与@import的区别？

作用不同: @import只能用于加载CSS

加载顺序差别，@import会先加在DOM然后再加载CSS。

版本控制：Link可以通过JavaScript控制版本号，而@import不行

1. 浏览器内核

渲染引擎和JS引擎。

Trident: IE

Gecko: Firefox

Presto: Opera

Webkit: Safari, Chrome

**Mine**

JSONP原理：通过<script> 标签发起一个get请求来取代XHR请求，JSONP生成一个<script>标签并插到DOM中，然后浏览器会接管并向src属性所指向的url地址发送请求。

当服务器返回请求时，响应结果会包装成一个JavaScript函数，并由该请求所对应的毁掉函数调用。

# 前端页面渲染机制

用户访问页面流程：