

前端开发那些事儿



张百鸽

美小技术部 - 前端组

- 前端发展史
- 前端框架
- 开发技能
- 面试中，必考察的点

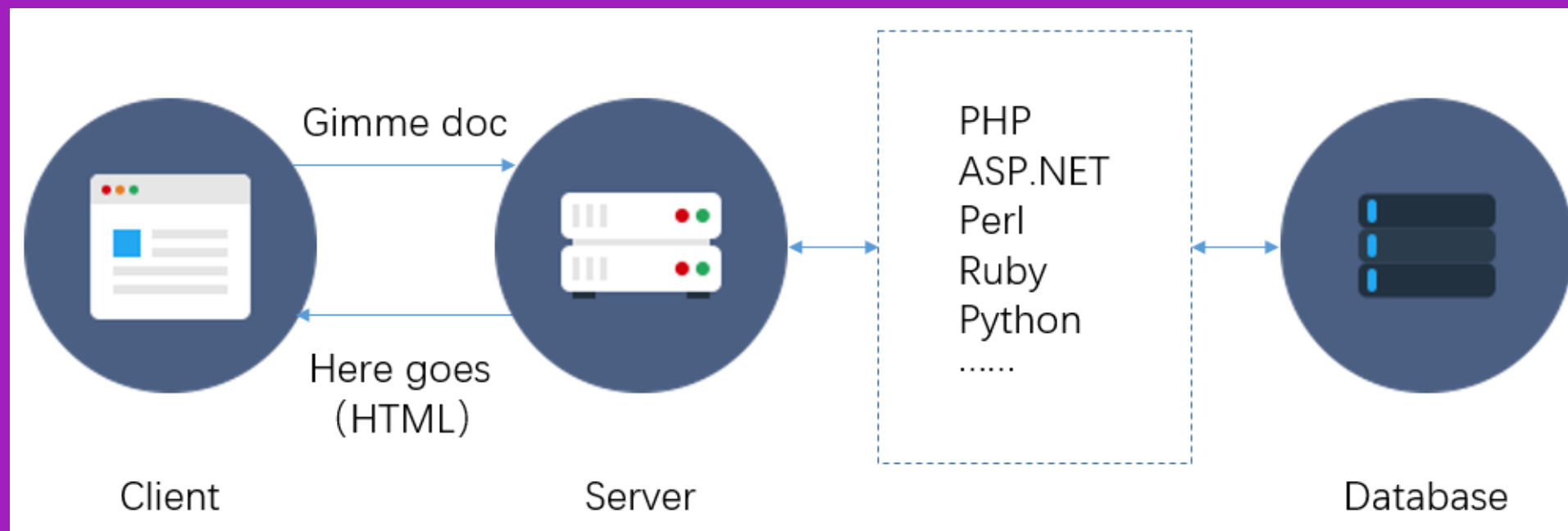
前端发展史-刀耕火种

从静态走向动态

年份	事件
1994	Tim Berners - Lee(万维网之父)创建W3C，Tim基友发布CSS 网景推出了第一版Navigator 前端起点(Web 1.0时代)
1995	Rasmus Lerdorf 动态Web网页设计的服务端脚本PHP(Personal Home Page)诞生 网景推出JavaScript - 实现客户端计算任务
1996	微软推出了iframe标签 - 实现异步的局部加载
1999	W3C发布第四代HTML标准 微软推出用于异步数据传输的ActiveX 各大浏览器厂商模拟实现了 - XMLHttpRequest - Ajax诞生
2005	谷歌利用Ajax技术打造Gmail和谷歌地图之后 - Ajax大受关注 前端的第一次飞跃（进入Web2.0时代）
2006	XMLHttpRequest被W3C正式纳入标准 新浪、搜狐、网易、腾讯、淘宝

(2005年之前，后端主导，程序员不区分前端和后端)

前端发展史



前端发展史 - JQuery一统天下

从后端走向前端

年份	事件
2006	John Resig发布了JQuery 灵活的DOM操作 优雅的语法、符合直觉的事件驱动 极易上手，基于jQuery插件构成了庞大的生态系统
2008之前	IE的市场份额巨大（版本太多，问题太多，ugly，buggy），加上浏览器性能弱，标准化程度低，造成Web发展受到制约
2008	谷歌V8引擎 - 现代浏览器的崛起终结了微软的垄断时代 前端计算能力过剩

（2008之前的前端不叫前端，岗位定位网页设计师 / 美工）

前端发展史-移动互联

从前端走向全端

年份	事件
2007	第一代iphone发布
2008	第一台安卓手机发布（移动端的发展进程和PC的历史如出一辙）
	<p>Native App的天下(原生)</p> <p>浏览器试图取代操作系统的篡位之心从未消减</p> <p>Web App太多好处（无需开发两套版本，无须安装，无须手动升级，无须审核，最大的是降低成本）</p> <p>Web App太多问题（H5几大API: Geolocation API、Vibration API、Luminosity API、Camera API）</p> <p>Web App性能差，是Native App还能作为主流存在一段时间的重要原因</p> <p>Hybrid, nw, Electron, react native, weex, 以及半路杀出来的微信小程序 - 都是在想能方面向Native靠近的尝试</p>

前端发展史-发现新大陆

年份	事件
2009	2009 Ryan Dahl发布了node (node基于V8引擎的服务端JavaScript运行环境，相当于一个虚拟机，js在服务端语言中有了一席之地) gitHub 2008年上线

(前端的第二次飞跃) 意味着JavaScript走出浏览器的藩篱，迈出了全端化的第一步

前端发展史 – 百花齐放，迈向工程化

年份	事件
2009	标准组织发布第五代JavaScript (ES5) 前端装备整体提升，如同改革开放进入一个目不暇接的新时代 AngularJS诞生
2010	NPM、BackboneJS和RequireJS,标志着JavaScript进入模块化开发的时代
2011	微软公司发布Windows 8操作系统，将JavaScript作为应用程序的开发语言之一，直接提供系统支持
2012	微软发布TypeScript语言
2013	Facebook发布UI框架库React ECMA正式推出JSON的国际标准
2014	Vue.js诞生 – 前后端分离大势所趋 ，第五代HTML标准发布 H5 – 浏览器厂商主导，与W3C合作制订的一整套Web应用规范（新的岗位H5前端开发工程师）
2015	Angular框架宣布，2.0版将基于微软公司的TypeScript语言开发 ECMA标准化组织正式批准了ECMAScript 6语言标准
2016	《ECMAScript 2016 标准》发布
2017	《ECMAScript 2017 标准》发布，正式引入了 async 函数

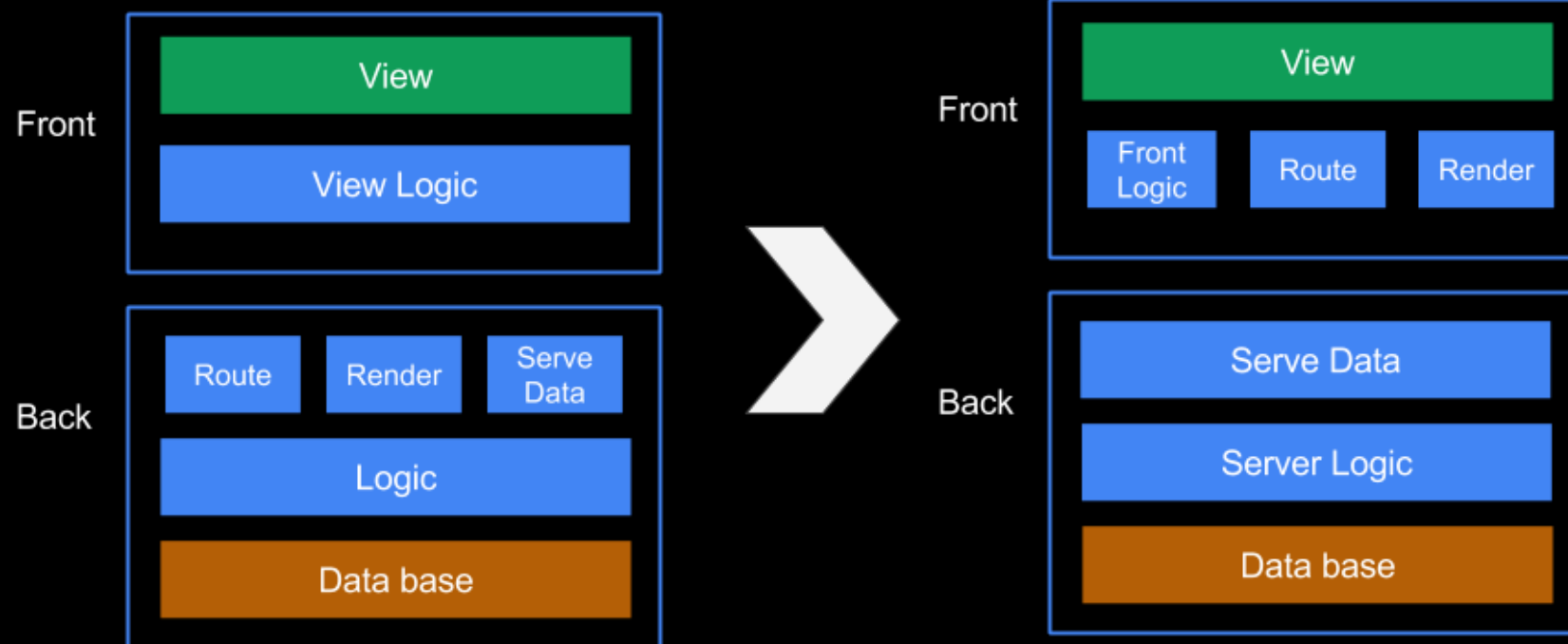
（2008年之后，前端工程师的岗位开始被大公司引进，直至现在，细分领域：H5前端开发，移动端前端开发，）

前端发展史

Web的演化-从后端到前端



前后端的分工明确



国内领先的实时后端云

前端框架

#	库 / 框架
模版引擎	Ejs、Pug、Underscore.template、NodeTpl、artTemplate、juicer(node),Django(python)等
UI库	Ant Design、Material UI、Bootstrap、We UI、Lay UI、Amaze UI、JQuery UI、自造UI等
CSS库	SASS、LESS、CSS Modules等
JS库(统一接口封装)	JQuery、Zepto、requirejs、backbone.js、d3.js, Sea.js/KISSY (阿里系), Pomelo (网易系), fis/esl (百度系), JX/MT/FrozenUI (腾讯系) 等
前端框架(具有特定的业务场景)	AngularJS、React、Vue、React-Native、Ionic,Walle(阿里系)等
构建工具	Webpack、Gulp、Grunt等
IDE工具	VS Code、chrome浏览器、chrome开发者工具、翻墙工具等

只有永远的类库， 没有永远的框架

开发技能

中级前端开发工程师

#	#
HTML	HTML5新规范
CSS	CSS3，以及SASS，或者LESS-页面布局，盒模型等
原生JS	原型链，DOM事件，面向对象，this关键字等
熟练使用第三方框架	理解框架实现的原理
网络层面	HTTP协议
能够进行插件开发	组件开发

开发技能

高级前端开发工程师

#

能够自己开发JS库

精通框架底层原理

能够实现高复用性，高性能的项目

实现自动化框架测试

开发技能

全栈开发工程师

#	#
测试类	实现前端的自动化测试和集成测试
通信类	跨域通信，接口通讯
前端安全类	XSS，CSRF
效能类	页面性能，项目优化
服务端	熟悉使用node作为后端服务（node，python，php）

面试中，必考察的点

#	#
HTML	H5规范，HTML页面渲染原理
CSS	CSS3属性，Flex,Grid,SASS和LESS区别，布局，盒子
js基础	数据类型，this，闭包，继承，XMLHttpRequest
js进阶	技术栈、第三方框架实现原理，使用的注意事项，React Diff算法,ES6和ES5对比优劣
js高级	实现一个高难度的demo，算法，MVVM和MVC
网络类	HTTP协议，HTTP状态码，HTTP请求头,Ajax，Fetch，Promise，异步同步，Event Loop
通信类	跨域，跨域的解决方案
效能类	性能优化，函数优化，代码优化，浏览器
项目经验	做过的最复杂的模块或功能
workflow	了解个人开发习惯，git的使用，构建工具的使用

面试中，必考察的点

问题	考察点
假设高度已知，写出三栏布局，其中左栏，右栏宽度格为300px，中间自适应	HTML语义化，CSS布局，CSS3新属性
谈谈你对CSS和模型的认识	标准和IE，如何设置，BFC
DOM事件级别，DOM事件流	对于DOM0,DOM2,DOM3的理解
什么是HTTP协议	考察对于HTTP的特点，组成，方法，状态码的理解
使用js实现5种方式定义对象	new关键字，类继承，原型继承，混合继承
实现一个数组的操作	对于Array的使用
实现一个treeSelect	递归算法
描述一下url输入到页面的渲染的整个过程	HTML树渲染，Style Tree渲染，js脚本单线程执行，网络层面的认知，能够注意到的优化点
React和Redux的关系，和使用	原理理解
做过的最复杂的模块或功能	个人实际开发的场景复杂度，分析问题，解决问题的能力
描述一下团队工作流	个人在团队中是一个什么角色，标准化如何，开发的模块起到什么作用

从前端走向全栈

- 优秀的前端工程师招聘难度越来越大
- 全球架构师峰会-阿里菜鸟高级前端技术专家-银鹏提出 **全栈计划: Let's Full Stack**全栈是一种“创业精神”

HTML



CSS



JS



谢谢!