

前端是什么

我们每天都在上网，我们上网上的是什么，目的是什么？最简单的就是看图片，视频，文字，这些在计算机中统称为叫超文本。所以，我们访问一个网页，到底做了些什么？可以分为两个部分，这个先讲第一个部分，叫做前端，可以理解是一个交互的前台。

浏览网页做了什么

浏览器内核（是用来渲染网页内容的，把你的网页代码转化为可见的页面8）

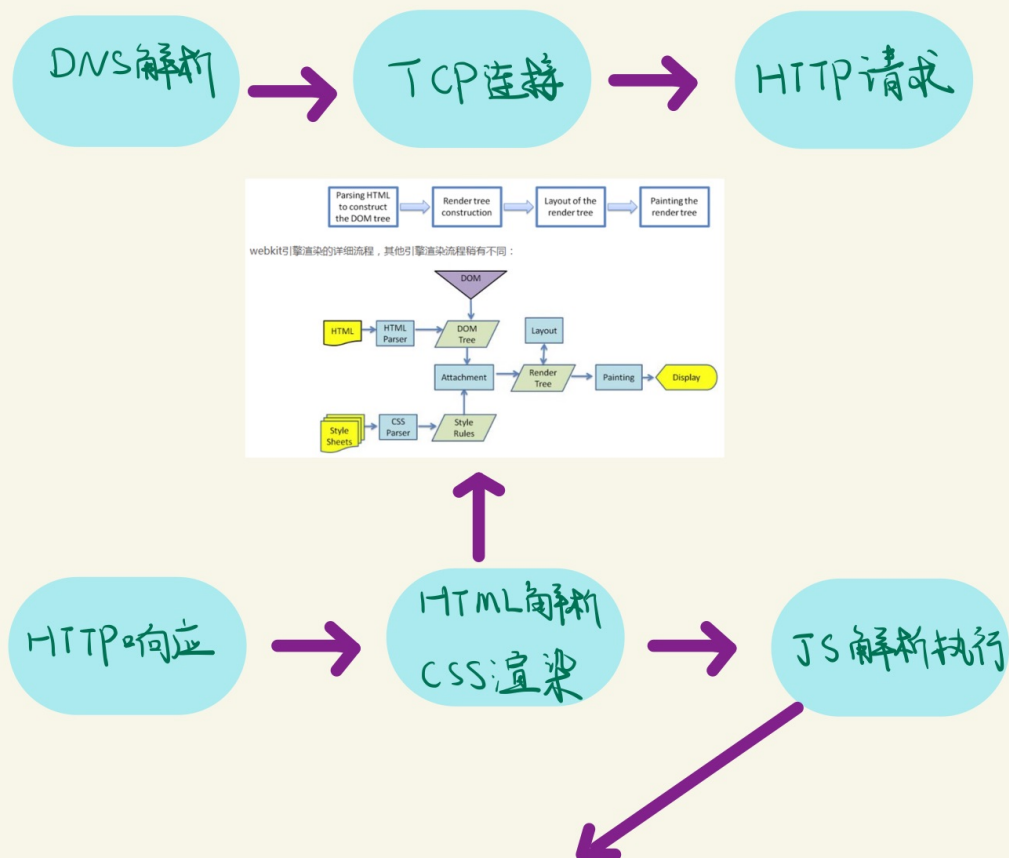
各大主要浏览器使用内核也是有差别的，大致可以分为以下几类：

- Trident 内核(IE) -> EdgeHTML 内核（旧版） -> Chromium内核（新版）：
IE -> edge(2022年6月15日，IE最终退出了大舞台，从此前端工程师松了一口气)
- Gecko 内核： FireFox
- Webkit 内核 -> Blink引擎： Chrome
- Webkit 内核： Safari
- Firefo内核： Gecko
- Presto-> Blink： Opera

浏览器js引擎（js引擎用来解释执行js代码）

比如：Chrome V8（nodejs就是基于V8引擎的后端开发）

浏览器渲染的流程：



1. 语义分析，即进行分词
2. 语法分析，即生成AST树
3. 预解析，即对某些变量与函数进行声明提前等的处理等
4. 运行，即根据AST树，或其它中间表示形式来执行JS代码

- **DOMTree**：浏览器将 HTML 解析成树形的数据结构。
- **CSSRuleTree**：浏览器将 CSS 解析成树形的数据结构。
- **RenderTree**：DOM 和 CSSOM 合并后生成 Render Tree。
- **layout**：有了 Render Tree，浏览器已经能知道网页中有哪些节点、各个节点的 CSS 定义以及他们的从属关系，从而去计算出每个节点在屏幕中的位置。
- **painting**：按照算出来的规则，通过显卡，把内容画到屏幕上。
- **reflow**（回流）：当浏览器发现某个部分发生了点变化影响了布局，需要倒回去重新渲染，内行称这个回退的过程叫 reflow。reflow 会从这个 root frame 开始递归往下，依次计算所有的结点几何尺寸和位置。reflow 几乎是无法避免的。现在界面上流行的一些效果，比如树状目录的折叠、展开（实质上是元素的显示与隐藏）等，都将引起浏览器的

reflow。鼠标滑过、点击.....只要这些行为引起了页面上某些元素的占位面积、定位方式、边距等属性的变化，都会引起它内部、周围甚至整个页面的重新渲染。通常我们都无法预估浏览器到底会 reflow 哪一部分的代码，它们都彼此相互影响着。

- `repaint`（重绘）：改变某个元素的背景色、文字颜色、边框颜色等等不影响它周围或内部布局的属性时，屏幕的一部分要重画，但是元素的几何尺寸没有变。

前端发展简史

1.html

HTML称为超文本标记语言，是一种标识性的语言。它包括一系列标签。通过这些标签可以将网络上的文档格式统一，使分散的Internet资源连接为一个逻辑整体。HTML文本是由HTML命令组成的描述性文本，HTML命令可以说明文字、图形、动画、声音、表格、链接等。

2.显示html

超文本传输协议，w3c。

3.浏览器出现

Netscape，mosic浏览器;微软，IE浏览器;Netscape被收购，Firefox浏览器（开源）;Google，Chrome浏览器。

Netscape技术：

http cookie

JavaScript

ssl协议

jar格式文件

4.浏览器技术

浏览器两个重要部分：

- 外壳
 - 外壳提供用户交互界面

- 内核
 - 提供html, css, 图像的渲染引擎, 提供dom编程接口
 - 提供JavaScript引擎

5.JavaScript

JavaScript和Java没有任何关系。

jsx

- React使用JSX来替代常规的JavaScript。JSX是一个看起来很像XML的JavaScript语法扩展。我们不需要一定使用JSX, 但它有以下优点:
 - JSX执行更快, 因为它在编译为 JavaScript代码后进行了优化。
 - 它是类型安全的, 在编译过程中就能发现错误。
 - 使用JSX编写模板更加简单快速。
 - jsx是在js代码里直接写XML的语法, 每一个XML标签都会被JSX转换工具转换成纯JS代码, 使用JSX可以使组件的结构和组件之间的关系看上去更加清晰。

TypeScript

TypeScript 是一种由微软开发的自由和开源的编程语言, 它是 JavaScript 的一个超集, 扩展了 JavaScript 的语法。

TypeScript 是 JavaScript 的超集, 扩展了 JavaScript 的语法, 因此现有的 JavaScript 代码可与 TypeScript 一起工作无需任何修改, TypeScript 通过类型注解提供编译时的静态类型检查。

TypeScript 可处理已有的 JavaScript 代码, 并只对其中的 TypeScript 代码进行编译。

6.ECMAScript

ECMAScript是一种由Ecma国际(前身为欧洲计算机制造商协会,英文名称是European Computer Manufacturers Association)通过ECMA-262标准化的脚本程序设计语言

7.v8引擎

基于v8引擎，诞生了nodejs,可以在服务端运行js

Nodejs

<https://nodejs.org/zh-cn/>

Node.js® 是一个开源、跨平台的 JavaScript 运行时环境

Node.js 可以做什么

Node.js 作为一个 JavaScript 的运行环境，仅仅提供了基础的功能和 API。

然而，基于 Node.js 提供的这些基础能，很多强大

的工具和框架如雨后春笋，层出不穷，所以学会了 Node.js，可以让前端程序员胜任更多的工作和岗位：

- ① 基于 Express 框架 (<http://www.expressjs.com.cn/>)，可以快速构建 Web 应用
- ② 基于 Electron 框架 (<https://electronjs.org/>)，可以构建跨平台的桌面应用
- ③ 基于 restify 框架 (<http://restify.com/>)，可以快速构建 API 接口项目
- ④ 读写和操作数据库、创建实用的命令行工具辅助前端开发、etc...

npm和包

从哪里下载包

国外有一家 IT 公司，叫做 npm, Inc. 这家公司旗下有一个非常著名的网站：

<https://www.npmjs.com/>，它是全球最

大的包共享平台，你可以从这个网站上搜索到任何你需要的包，只要你有足够的耐心！

到目前位置，全球约 1100 多万的开发人员，通过这个包共享平台，开发并共享了超过 120 多万个包 供我们使用。

npm, Inc. 公司提供了一个地址为 <https://registry.npmjs.org/> 的服务器，来对外共享所有的包，我们可以从这个服务器上下载自己所需要的包。

使用 npm 包管理工具下载的包，共分为两大类，分别是：

- 项目包
那些被安装到项目的 node_modules 目录中的包，都是项目包。
项目包又分为两类，分别是：

- 开发依赖包（被记录到 devDependencies 节点中的包，只在开发期间会用到）
- 核心依赖包（被记录到 dependencies 节点中的包，在开发期间和项目上线之后都会用到）
- 全局包
在执行 npm install 命令时，如果提供了 -g 参数，则会把包安装为全局包。
全局包会被安装到 C:\Users\用户目录\AppData\Roaming\npm\node_modules 目录下。

解决下包速度慢的问题:

1.切换 npm 的下包镜像源

1、首先输入 `npm get registry` 查看当前镜像源，

```
npm get registry
```

`https://registry.npmjs.org/`（npm默认镜像源）

2，修改镜像源 `npm config set registry xxx(镜像源地址)`，国内一般使用淘宝镜像源

```
npm config set registry https://registry.npm.taobao.org
```

3，npm 修改成功，输入 `npm get registry` 再次查看镜像源

```
npm get registry
```

2.为了更方便的切换下包的镜像源，我们可以安装 nrm 这个小工具，利用 nrm 提供的终端命令，可以快速查看和切换下包的镜像源。

8.css

html本身显示没问，但是随着需要更多的样式能力，促使css的诞生。css，层叠样式表。

由于css是一门非程序式语言，它没有变量，没有函数，作用域等。当需要编写大量的代码时，缺点很多。于是出现了sass，less

- sass
 - Sass 完全兼容所有版本的 CSS。Sass扩展了CSS3，增加了规则、变量、混入、选择器、继承、内置函数等等特性。Sass生成良好格式化的 CSS 代码，易于组织和维护。Sass文件后缀为.scss。浏览器并不支持Sass 代码。因此，你需要使用一个Sass 预处理器将Sass代码转换为CSS代码。
- less
 - Less 是一门CSS预处理语言，它扩展了CSS语言，增加了变量、Mixin、函数等特性，使CSS更易维护和扩展。Less可以运行在Node或浏览器端。

9.动态，静态网页

10.同步异步加载

异步技术 ajax(Asynchronous JavaScript and XML)

11.CommonJS

CommonJS是一个项目，其目标是**为JavaScript在网页浏览器之外创建模块约定**。创建这个项目的的主要原因是当时缺乏普遍可接受形式的JavaScript脚本模块单元，模块在与运行JavaScript脚本的常规网页浏览器所提供的不同的环境下可以重复使用。

注意：浏览器之外，在浏览器内我可不认！

应用场景：

- `Node` 是 CommonJS 在服务器端一个具有代表性的实现；
- `Browserify` 是 CommonJS 在浏览器中的一种实现；
- `webpack` 打包工具对 CommonJS 的支持和转换；也就是前端应用也可以在编译之前，尽情使用 CommonJS 进行开发。

特点：

- 在 `commonjs` 中每一个 js 文件都是一个单独的模块，我们可以称之为 module；
- 该模块中，包含 CommonJS 规范的核心变量: exports、module.exports、require；

- exports 和 module.exports 可以负责对模块中的内容进行导出；
- require 函数可以帮助我们导入其他模块（自定义模块、系统模块、第三方库模块）中的内容；

12.ES6 Module

ES 模块是 ES6（2015 年）年引入的标准。创建的目的是为了标准化 JS 模块运作，和在浏览器中使用模块的方法（在此之前并不支持模块）。泛指 JavaScript 语言的下一代标准。

13.Babel