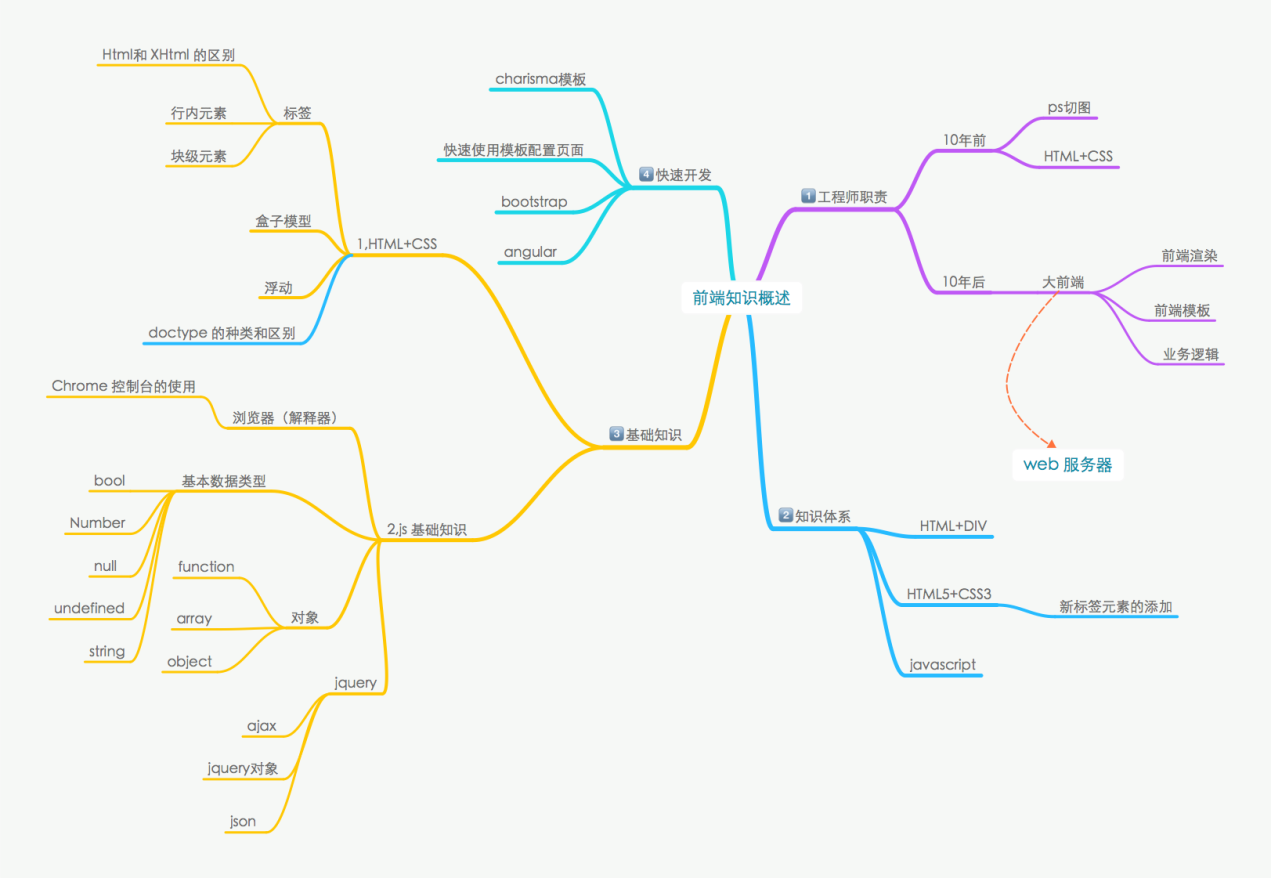
1. 基础的前端知识，有的知识也只是做了解简单介绍。主要是想让大家对前端有一个基本的了解。

知识导图：



主要讲解基础知识这块

HTML+CSS+JS

HTML :

DOCTYPE的种类：

<!--DOCTYPE的声明种类

在 HTML 4.01 中，<!DOCTYPE> 声明引用 DTD，因为 HTML 4.01 基于 SGML。DTD 规定了标记语言的规则，这样浏览器才能正确地呈现内容

-->

<!--html5-->

<!DOCTYPE html>

<!--HTML 4.01 Strict

该 DTD 包含所有 HTML 元素和属性，但不包括展示性的和弃用的元素（比如 font）。不允许框架集（Framesets）

-->

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">

<!--过渡类型

该 DTD 包含所有 HTML 元素和属性，包括展示性的和弃用的元素（比如 font）。不允许框架集（Framesets）

-->

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<!--允许框架集的使用

该 DTD 等同于 HTML 4.01 Transitional，但允许框架集内容

-->

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN" "http://www.w3.org/TR/html4/frameset.dtd">

Mate标签：

元数据（Metadata）是数据的数据信息。

<meta> 标签提供了 HTML 文档的元数据。元数据不会显示在客户端，但是会被浏览器解析。

META元素通常用于指定网页的描述，关键词，文件的最后修改时间，作者及其他元数据。

元数据可以被使用浏览器（如何显示内容或重新加载页面），搜索引擎（关键词），或其他 Web 服务调用。

**实例 1 - 定义文档关键词，用于搜索引擎：**

<meta name="keywords" content="HTML, CSS, XML, XHTML, JavaScript">

**实例 2 - 定义web页面描述：**

<meta name="description" content="Free Web tutorials on HTML and CSS">

**实例 3 - 定义页面作者：**

<meta name="author" content="Hege Refsnes">

**实例 4 - 每30秒刷新页面：**

<meta http-equiv="refresh" content="30">

可以访问其他链接的标签

<link rel="stylesheet" href="">

<a href=""></a>

<script src=""></script>

<img src="" alt="">

Href:

href是Hypertext Reference的缩写，表示超文本引用。用来建立当前元素和文档之间的链接。常用的有：link、a。

浏览器会识别该文档为css文档，并行下载该文档，并且不会停止对当前文档的处理。

Src:

src是source的缩写，src的内容是页面必不可少的一部分，是引入。

src指向的内容会嵌入到文档中当前标签所在的位置。常用的有：img、script、iframe。

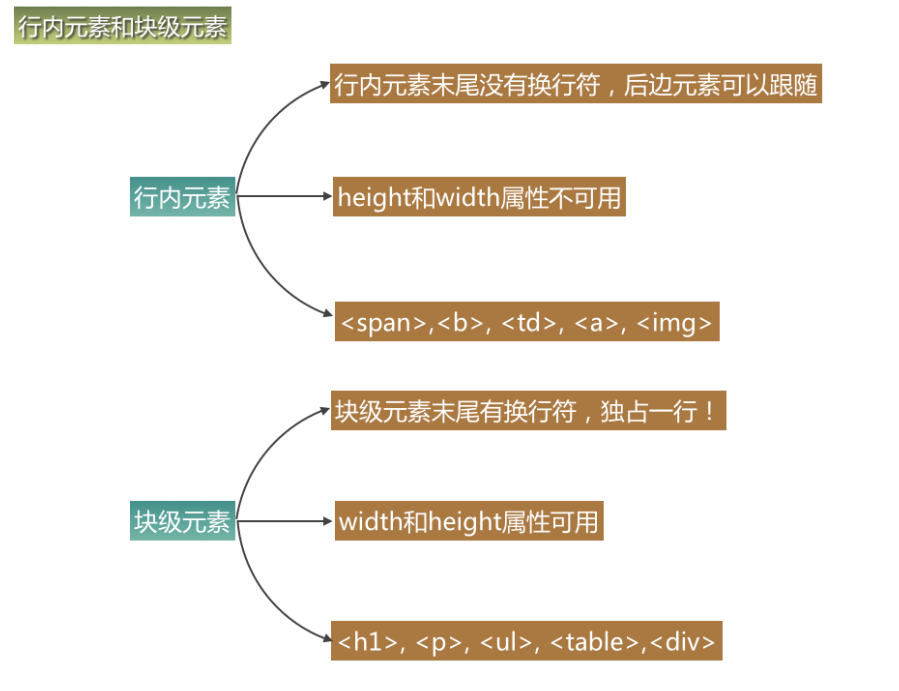
当浏览器解析到该元素时，会暂停浏览器的渲染，知道该资源加载完毕。

这也是将js脚本放在底部而不是头部得原因。

这里有个同源策略，后面js做测试

Html中都是由标签元素构成的，元素可以分为

## ****行内元素和块级元素****

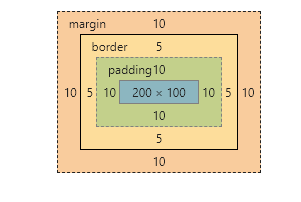


有一个例子：

CSS盒模型：

margin

padding  
border  
content



css布局和BFC

1）文档流（普通流，常规流）

浏览器会把 html 元素按照从上到下，从左到右依次排放。

Html 中全部元素都是盒模型，盒子会占用一定的空间，依次排放在HTML中，形成了文档流。

某些特殊的 css 样式会把元素脱离文档流。某个元素脱离文档流后，在文档流中的其他元素将忽略该元素并填补其原先的空间。

* 在常规流中，元素一个接着一个排列;
* 在**块级格式化上下文**里面， 它们**竖着**排列；
* 在**行内格式化上下文**里面， 它们**横着**排列;
* 对于**相对定位**(relative positioning)，position: relative，盒偏移位置由这些属性定义top，bottom，left and right。**即使有偏移，仍然保留原有的位置**，其它常规流不能占用这个位置。

脱离文档流：

浮动(Floats)

* 盒称为浮动盒(floating boxes)；
* 它位于当前行的开头或末尾；
* 这**导致常规流环绕在它的周边**，除非设置 clear 属性；

绝对定位(Absolute positioning)

* 绝对定位方案，**盒从常规流中被移除**，不影响常规流的布局；
* 它的定位相对于它的包含块，相关CSS属性：top，bottom，left及right；
* 如果元素的属性position为absolute或fixed，它是绝对定位元素；
* 对于position: absolute，元素定位将相对于最近的一个relative、fixed或absolute的父元素，如果没有则相对于body；

到这里，已经对CSS的定位有一定的了解了，从上面的信息中也可以得知，块格式上下文是页面CSS 视觉渲染的一部分，**用于决定块盒子的布局及浮动相互影响范围的一个区域**。

此处有个演示例子：

一个css 3d动画的例子：

Js: