

# 前端知识体系复盘-第一阶段

---

## 一、多个团队配合

产品经理：负责需求调研、需求可行性分析、设计低保真原型图

UI:开始设计原稿，PS软件、XD软件 提供原稿

前端：开始还原，蓝湖在线工具，免费的功能限制了人数 10个人

后端：开始设计数据库，数据库---业务---接口

测试：开始设计测试用例，开发会评审。

前后端联调：前端自己mock了数据，后端提供接口，两个团队一起调试接口。

## 二、HTML标签

### HTML核心概念

HTML超文本标记语言：

超文本：超级文本，一个文本可以加载各种资源，包括音频、视频、文本、超链接

普通文本一般只负责文字显示和渲染。

超级文本可以连接所有资源，形成网状结构

标签语言特点：

只是一种开发模式。

标签语言：显示内容，传输内容

默认版本:H5

```
<!DOCTYPE HTML>
```

H4版本

```
<!DOCTYPE HTML public="" "dtd 4.01 xxx.dtd">
```

浏览器会根据你的版本来决定如何解析你的HTML代码

开发使用H5版本，但是用到了很多H4的内容。

浏览器会有兼容模式，高版本的HTML代码，里面使用以前HTML标签。

如果你HTML是低版本的，浏览器也能解析，兼容模式解析你们代码

## SEO优化

### 搜索引擎优化

保证你们的网页通过关键字搜索尽量排名靠前

Vue、React项目、单页面应用。不适合SEO。

多页面开发。

一般公司需要SEO网站，一般都不会用单页面开发。

## 三、CSS整理

### 基本概念

层叠样式表：网站中引入样式很多种类型。

研究：开发者自己写样式（内联、内部、外部）

浏览器默认样式：浏览器默认给标签设置了一些样式，比如H1标签

浏览器用户自定义：你们可以利用浏览器自己的设置，来修改我们的样式

最终浏览器要渲染页面：最终样式（根据用户自己写的代码+浏览器用户自定义样式+浏览器默认样式）

HTML标签本身没有任何样式，能看到效果都是CSS作用。

### 浏览器解析网页

浏览器底层有很多工具解析我们页面

HTML解析器：专门解析HTML代码。

CSS解析器：解析你们的CSS代码，计算出最终样式。

JS解析器：v8引擎 专门解析js代码，编译机器语言

音频、视频、图片解析器：

DOM树：表现出现就是一个JS对象

```
<div>
  <span></span>
  <img src="" title/>
</div>
```

DOM对象：

```
{
  tagName:"div",
  attr:[],
  children:[
    {
      tagName:"img",
      attr:[
        {name:"src",value:"xxx"},
        {name:"src",value:"xxx"},
      ],
    },
  ],
}
```

```
        children:[]
      }
    ]
  }
}
```

CSS代码：

内部样式、内联样式，立即用CSS解析器解析

外部连接：发送请求到服务器拿到这个CSS资源，CSS解析器解析这个代码

HTML解析器会先解析HTML代码。生成DOM树。

CSS解析器会解析你们CSS样式，生成最终样式（自己样式、浏览器样式）

将解析DOM树和CSS最终样式结合再一起形成渲染树，

调用浏览器底层layout进行页面排列布局（重绘回流）

进行页面渲染和显示

JS代码专门交给V8引擎来解析，最终作用页面

## 选择器

基础选择器：

组合选择器：

权重：计算权重，两种方式

scss或者less过后嵌套规则

嵌套规则一般不超过三层

```
div{
  span{
    .as{

    }
  }
}
.as{
  .title{

  }
}
```

写出来伪类选择器：

写出为伪元素选择器：

区别：

## 标签分类

块和行

```
display:inline-block
```

行内块：可以让一个行标签支持宽和高

比如：img标签

不要用行内块布局：

IFC标准：解决这些排列问题

[https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Reference/Global\\_Objects/Date](https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Reference/Global_Objects/Date)

## 盒模型

两类：

标准盒子模型：现代浏览器默认都是标准盒子模型

IE盒子模型：默认再老版本的IE浏览器里面，采用盒子模型。

默认情况下都是标准盒子模型，自己计算盒子真正的大小

```
finalWidth=border*2+padding*2+contentWidth
```

IE盒子最终大小

```
finalWidth = width(width=padding+border+contentWidth)
```

转化过程：

```
box-sizing:border-box\content-box
```

场景：

1. PC端一般默认使用标准盒子
2. 移动端开发基本上都使用IE盒子。移动端百分比布局。

面试题：

1. margin-top传递性：子盒子设置了margin-top，传递父盒子。BFC盒子、border
2. 盒子margin、padding设置一个值、两个值、三个值、四个值代表什么意思

width:100% //100%代表内容区域为100%，如果盒子边框内边距，整个盒子大小变大

width:auto //也能达到100%的效果，如果你给盒子设置了边框或者border，盒子不会变大，

## 浮动

来源：浮动这个技术最早解决图文混排的问题。

后来将浮动引用于布局。

标准文档流：行原生共享一行空间、块原生独占一行空间，按照这个规则来，说明遵循标准文档流

浮动特点：

1. 浮动的元素脱离文档流。
2. 多个浮动元素，后面元素挨着前面元素边框来排列
3. 浮动元素原来空间小时。其他元素被占据这个空间
4. 给行标签设置浮动，标签默认变成行内块的特性

清楚浮动：

1. clear: left、right、 both
- 2.

```
.clearfix::after{
  content: "",
  display: block,
  clear: both
}
```

```
.clear{
  clear: both
}
<div class="clearfix">
  <div><div>
  <div class="clear"></div>
</div>
```

## 定位

分类：

static静态、relative相对定位、absolute绝对、fixed固定定位、sticky(粘性定位)

z-index：定位层级

盒子居中技巧：

要求大家至少用三种方式来实现

## H5模块

新增了一些标签

```
<input type="email"> //
  <input type="date">
    <input type="number">
      <input type="range">
```

语义化标签：标签名字更加具有语义化

门户网站，为了考虑SEO，我们可能采用语义化标签

video

audio

拖拽：表格拖拽、

地图开发:

网页版本: 需要再每个地图产商他们的官网注册一个账号, 申请一个key (身份认证)

一般都是将地图的SDK引入项目中。

小程序版本: 需要下载对应腾讯地图sdk, 引入小程序中。调用sdk提供的接口, 开始进行地图的数据获取

```
<map></map>
```

## CSS3内容

圆角: 原理

```
border-radius: 10px/20px 10px/5px 20px/8px
```

阴影效果:

```
box-shadow: 0px 0px 10px 10px rgba() outset
```

渐变效果:

径向渐变:

线性渐变:

结构选择器:

```
nth-child:  
nth-of-type:
```

透明:

盒子整体透明

```
opacity:  
rgba(0,0,0,0.3)
```

过渡效果

2D动画

3D动画 (了解)

animation动画:

## 响应式布局

一般再开发移动端页面时候

两种设计思路:

1. 一套代码适配多端, 响应式布局 (写起来很麻烦, 维护起来也很麻烦)  
媒体查询思路: bootstrap工具
2. 专门针对H5端开发一套页面 (主流方案)

绝对单位: px

相对单位: em、rem、vw、vh

em: 先找自己这个模块font-size这个属性, 如果没有这个属性, 找父元素font-size

rem: 只会寻找html标签身上的font-size

vw: 参考屏幕宽度

vh: 参考屏幕高度

flexiblejs这插件, 封装了一套工具, 可以自己根据你们屏幕来给html标签动态设置font-size

flexiblejs经常用于: 大屏可视化、后台管理系统

## **sass编程 (扩展)**

嵌套结构:

变量定义:

sass写的代码如何编译