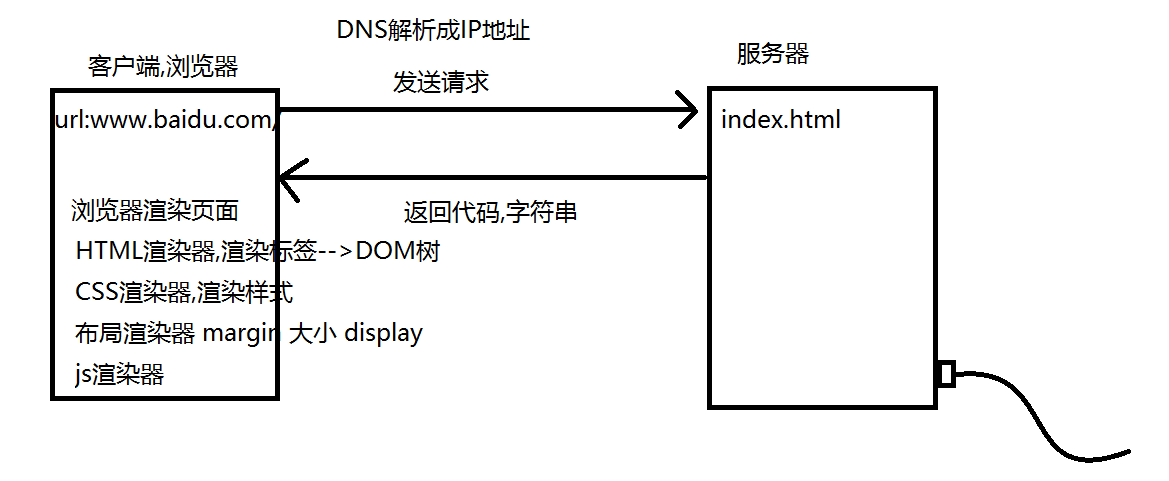
1. 676949624
2. 页面组成部分

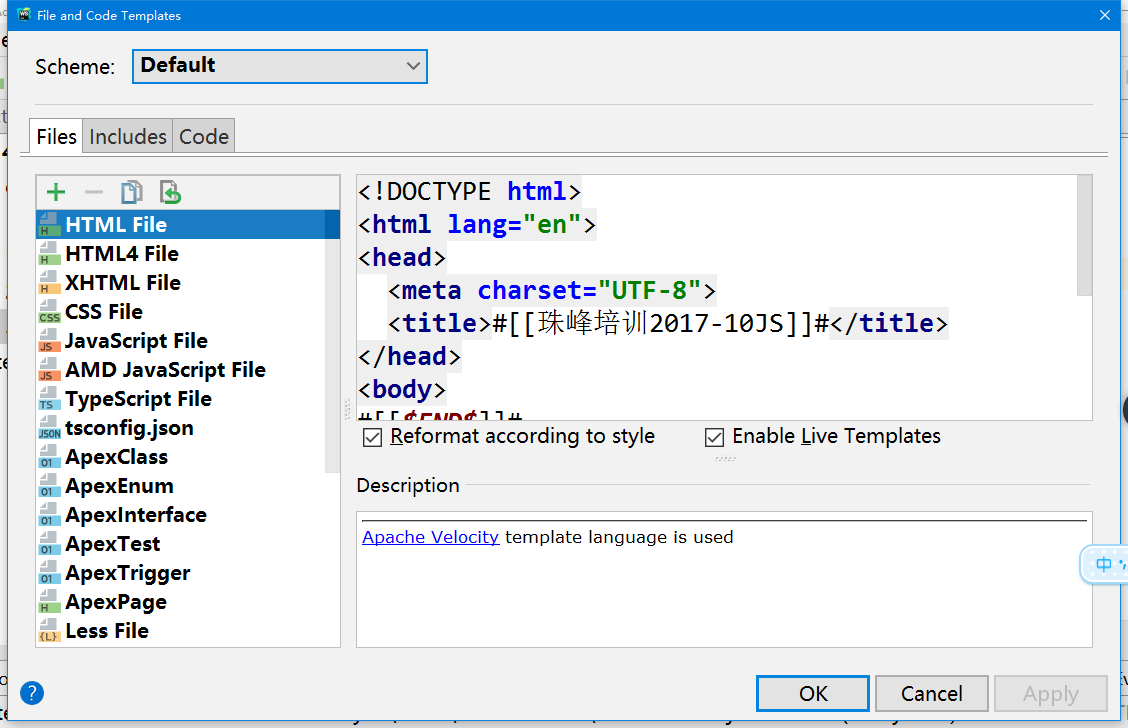
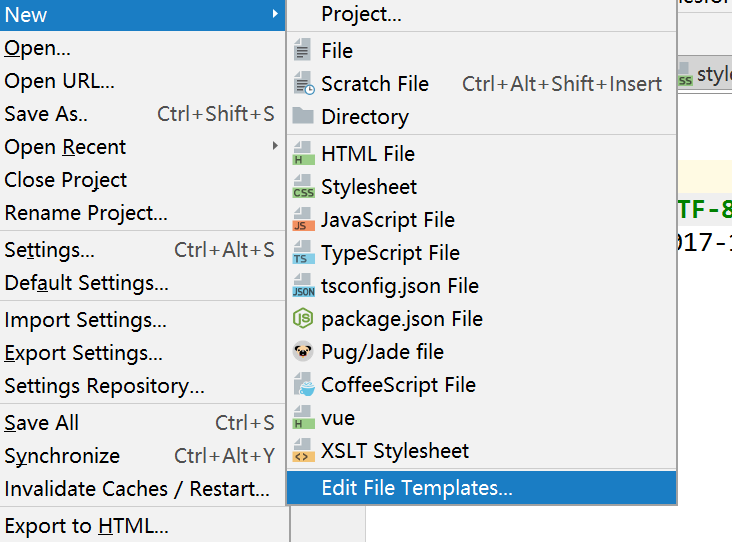
HTML:H5

CSS:css3

JS:

面试题:从打开一个url到页面呈现出中间经历了什么(重点)  


**编辑文件模板**



1. 页面的渲染顺序(必须记住)

理想:先渲染HTML,在渲染css,最后JS

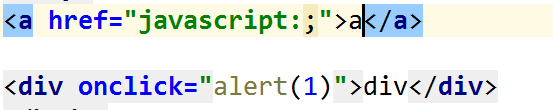
注意:

页面渲染是从上到下依次加载渲染

当页面从上到下一次渲染的时候遇到link和Style标签的时候,让这俩标签去请求或者加载样式,但是不会阻碍页面继续加载(异步),当页面的html加载完成形成DOM树这个时候,这个时候css树形成,DOM树和css树合并渲染

1. js的引入方式

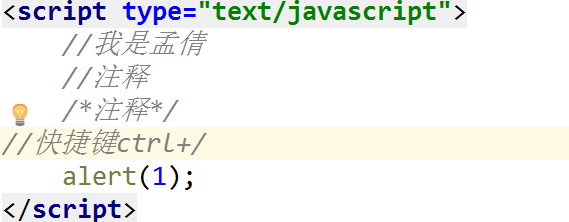
行内,写在标签上,不建议写



点击a标签的时候什么都不干(阻止默认行为的方式),注意:a标签的:hover没有兼容性

点击div弹出字符串”1”

Script标签内

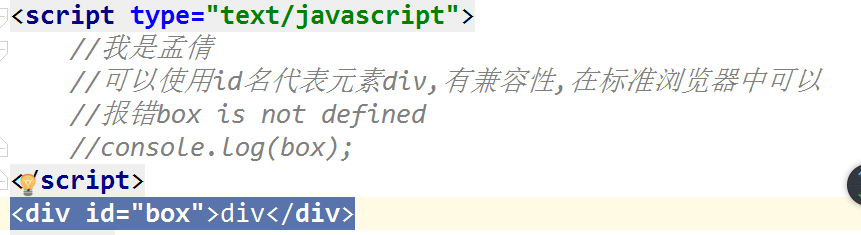


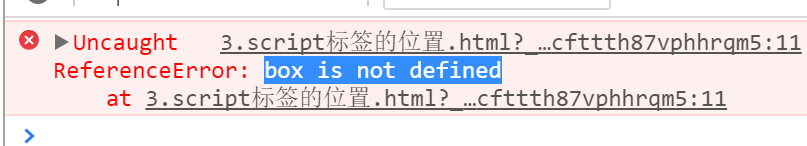
外链式 <script type=”text/javascript” src=”地址”>

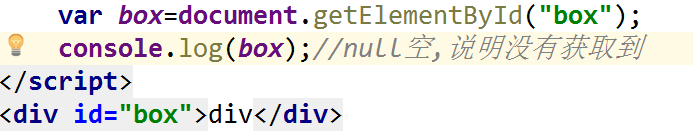
<**script src="js/1.js"**>  
 *//这里不要写,写了也没有用  
 //alert(2)*</**script**>

1. js的写入位置

页面渲染的时候遇到Script标签,加载他但是会阻碍页面的继续加载,只有这个Script加载完成后才会继续加载下面的html,如果在这个Script中使用了未加载的元素此时就会报错或者是获取不到



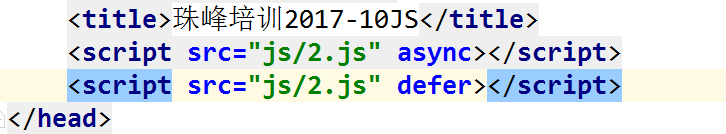






为了保险我们都是将Script标签写在body/html最后面

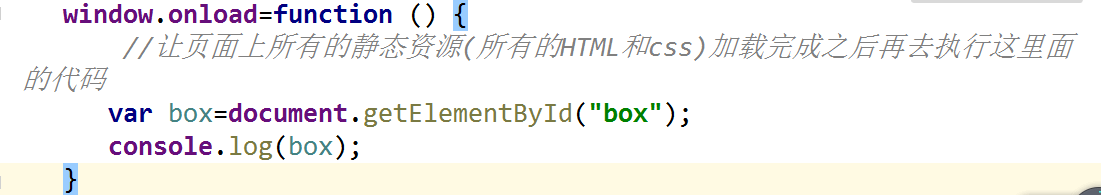
如果非要写在上面还有异步的效果(不会阻碍HTML加载)也是可以的,但是只能使用外链的写法



给Script标签加一个async(异步)属性,注意IE下不兼容使用的是defer属性

告诉浏览器先不要加载这个js等页面加载完成后再去加载这个js文件

如果不是外链式的Script中的js文件想要实现一个异步的效果可以写在window.onload=function(){代码写在这里面}**(重点记住)**



1. js组成部分

运行在浏览器上的时候我们可以说

BOM:浏览器对象模型 提供了一系列可以操作浏览器的API(属性/方法) window,location,history,screen,document,,

DOM:文档对象模型 HTML标签元素节点属性,关系,提供了一些操作dom的方法和属性

ECMAScript(核心):规定了Js的语法,变量,数据类型,属性..

运行在node环境中(服务器上)

宿主环境+ECMAScript

注意

1. document存在再DOM和BOM中都有,window.document==document
2. ES6:ECMAScript2015 配置ES6 环境 setting->language->javascript->选择ECMAScript6
3. BOM:history.go(-1),history.back(),history.forward(),可以省略window
4. 获取当前页面地址url: location.href也可以写成 window.location.href
5. self==window这俩是一个东西 window是个全局对象(老大)

上午的重点记住的内容

1. 页面的渲染方式,顺序,异步问题
2. Js文件加载实现异步功能的方式
3. 通过ID名获取元素,document.getElementByID(“id名”)
4. BOM:history.go(-1),history.back(),history.forward(),可以省略window
5. location.href获取页面URL.也可以写成 window.location.href