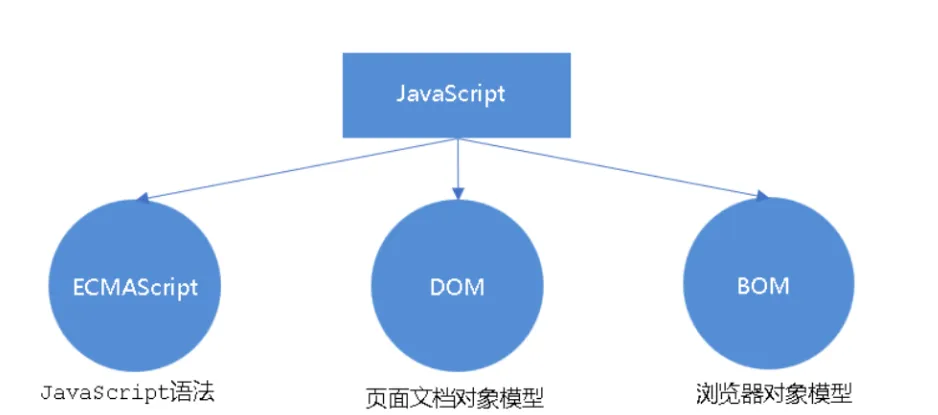
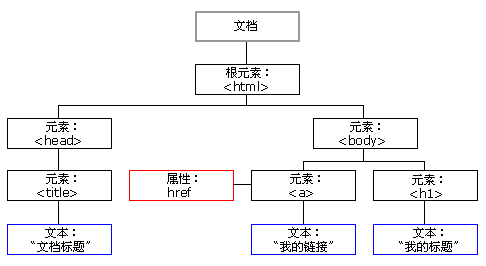
Web开发-JS应用&原生代码&前端数据加密&CryptoJS库&代码混淆&Obfuscator库



#知识点：  
1、安全开发-原生PHP-超级全局变量  
2、安全开发-原生PHP-代码审计案例  
---------------------------------------------  
0、安全开发-原生PHP-数据库通讯  
1、安全开发-原生PHP-身份验证技术  
2、安全开发-Cookie&Session&Token  
3、安全开发-原生PHP-代码审计案例  
---------------------------------------------  
1、安全开发-原生PHP-弱类型脆弱  
2、安全开发-原生PHP-函数&数据类型  
3、安全开发-原生PHP-代码审计案例  
---------------------------------------------  
1、安全开发-原生PHP-开发组件集合  
2、安全开发-原生PHP-模版引擎渲染  
3、安全开发-原生PHP-第三方编辑器  
---------------------------------------------  
1、安全开发-框架技术-ThinkPHP  
2、安全开发-框架安全-版本&写法  
3、安全开发-ThinkPHP-代码审计案例  
---------------------------------------------  
1、安全开发-原生PHP-文件安全操作  
2、安全开发-原生PHP-上传读取删除包含等  
3、安全开发-原生PHP-代码审计文件安全---------------------------------------------  
1、安全开发-原生PHP-RCE安全  
2、安全开发-原生PHP-代码执行&命令注入  
3、安全开发-原生PHP-代码审计案例  
---------------------------------------------  
1、安全开发-原生PHP-PHP.INI安全  
2、安全开发-原生PHP-全局文件&单函数  
3、安全开发-原生PHP-流量检测&AI算法  
---------------------------------------------  
1、安全开发-原生JS-语法模型概念  
2、安全开发-Ajax技术-JQuery&Axios  
3、安全开发-Ajax应用-文件上传&用户登录---------------------------------------------  
1、安全开发-原生JS-DOM树&BOM对象  
2、安全开发-原生JS-DOM安全&安全案例  
---------------------------------------------  
1、安全开发-原生JS-数据加密&代码混淆  
2、安全开发-原生JS-数据解密安全案例  
  
#章节点：(待补充)  
#具体点：(待补充)







演示案例：

* WEB开发-原生JS-语法模型概念
* WEB开发-Ajax技术-JQuery&Axios
* WEB开发-Ajax前端验证-上传&登录
* WEB开发-原生JS对象-DOM&BOM
* WEB开发-原生JS对象-DOM-XSS
* WEB开发-原生JS-数据加密&代码混淆

JavaScript被广泛用于Web应用开发，常用来为网页添加各式各样的动态功能,为用户提供更流畅美观的浏览效果。嵌入动态文本于HTML页面；对浏览器事件做出响应，读写HTML元素，在数据被提交到服务器之前验证数据；检测访客的浏览器信息；控制用户凭据，包括创建和修改等。  
  
#安全结合：  
1、发现更多的有利用价值的信息（URL、域名、路径等等）  
测试站、后台路径、未公开的路径、api地址等等  
2、发现敏感信息（硬编码的帐号、pass、API密钥、注释等等）  
硬编码帐号可登录、测试帐号可被登录、密钥泄露、注释中开发信息等等  
3、发现危险的代码（eval、dangerouslySetInnerHTML等等）  
URL跳转，XSS跨站、模版注入（SSTI）等  
4、了解网站的逻辑校验功能  
前端检测，加密逆向，数据走向等  
  
#学习文档：  
原生JS教程  
https://www.w3school.com.cn/js/index.asp  
  
前端技术JS实现：  
1、非加密数据大致流程：  
客户端发送->明文数据传输-服务端接受数据->处理数据  
  
2、加密数据大致流程：  
明文加密->客户端发送->密文数据传输-服务端接受数据->解密数据->处理数据  
  
疑问：  
安全测试中对数据进行修改提交会在上述那一步操作中  
  
#前端加密 Crypto库  
项目：https://github.com/brix/crypto-js  
参考：https://juejin.cn/post/7382893339181613068  
使用Crypto库进行MD5/SHA1/HMAC/AES/DES等加密  
<script src="crypto-js.js"></script>  
<script src="https://cdnjs.cloudflare.com/ajax/libs/crypto-js/4.0.0/crypto-js.min.js"></script>  
  
#前端加密 jsencrypt库  
项目：https://github.com/travist/jsencrypt  
参考：https://www.cnblogs.com/Lrn14616/p/10154529.html  
使用jsencrypt库进行RSA等加密  
<script src="jsencrypt.js"></script>  
<script src="https://cdn.bootcdn.net/ajax/libs/jsencrypt/3.3.2/jsencrypt.js"></script>  
  
#代码混淆  
混淆代码的主要目的是保护源代码，防止未经授权的复制、篡改或逆向工程。通过对变量名、字符串和控制流的修改，混淆代码看似毫无逻辑，但本质功能未变。混淆技术常用于商业应用和恶意软件中。  
-压缩、混淆、加密技术  
去除js代码中的不必要的空格、换行等内容。使源码压缩为几行内容，降低代码可读性，提高网站的加载速度。使用变量替换、僵尸函数、字符串阵列化、控制流平坦化、调试保护等手段，使代码变得难以阅读和分析，达到最终保护的目的，不影响代码原有功能，是理想、实用的javascript保护方案。  
在线混淆：见打包PDF  
https://obfuscator.io/

涉及资源：[资源下载地址](https://docs.qq.com/doc/DQ3Z6RkNpaUtMcEFr)