

# 认识JavaScript-Js

@Taching 1006082685



Front-end

Js

JavaScript 诞生于1995 年。它当时的目的是为了验证表单输入的验证。因为在JavaScript问世之前，表单的验证都是通过服务器端验证的。而当时都是电话拨号上网的年代，服务器验证数据是一件非常痛苦的事情。经过许多年的发展JavaScript 从一个简单的输入验证成为一门强大的编程语言。



# 1 Front-end Js

## 1.1 JavaScript 基本特点

JavaScript是一种属于网络的脚本语言,已经被广泛用于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。通常JavaScript脚本是通过嵌入在HTML中来实现自身的功能的。

- 是一种解释性脚本语言。
- 是一种基于对象的脚本语言。
- 是一种简单的弱类型脚本语言。
- 是一种相对安全的脚本语言。
- 是一种动态性脚本语言。
- 是一种跨平台特性脚本语言。



# 1 Front-end Js

## 1.2 JavaScript 用途

- 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。
- 在与服务器交互，返回的数据进行页面渲染。
- 检测访客的浏览器信息。
- 控制cookies，包括创建和修改等。
- 基于Node.js技术进行服务器端编程。
- ....





Front-end

Js

### 1.3 JavaScript我们必需要学?

- 所有浏览器都支持JavaScript.
- 现在, 全世界大部分的网页都在使用JavaScript.
- JavaScript 可以让网页呈现各种效果
- JavaScript 帮我们实现友好的用户体验
- 前后端分离模式开发 让团队效率更快, 更容易维护
- 让擅长的人做擅长的事。



# 1 Front-end Js

## 1.4 JavaScript 组成

核心：ECMAScript

ECMAScript是一个标准。

因为网景开发了JavaScript，为了让JavaScript成为全球标准，几个公司联合ECMA（European Computer Manufacturers Association）组织定制了JavaScript语言的标准，被称为ECMAScript标准。

简单描述：语法、类型、语句、关键字、保留字、运算符、对象（内置对象）

文档对象模型：DOM

Document Object Model。文档对象模型

浏览器对象模型：BOM

Browser Object Model。浏览器对象模型





## Front-end Js

### 2.1 JavaScript 引入方式

在HTML文件中书写

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="renderer" content="webkit"/>
  <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1"/>
  <meta http-equiv="Cache-Control" content="no-transform"/>
  <title>小试牛刀</title>
  <meta name="keywords" content=""/>
  <meta name="description" content=""/>
  <script src="/lib/js/jquery.js"></script>
  <script src="/lib/js/base.js"></script>
  <link rel="stylesheet" href="/css/index.css"/>
</head>
<body>
<script type="text/javascript">
  document.write('hello, world');
  alert('hello, world');
</script>
</body>
</html>
```



## 2 Front-end Js

### 2.1 JavaScript 引入方式

#### 外部引用

<script> 元素放在页面head元素中。

<script> 元素放到页面body元素中（body闭合标签前）

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="renderer" content="webkit"/>
  <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1"/>
  <meta http-equiv="Cache-Control" content="no-transform"/>
  <title>JavaScript外部引入方式</title>
  <script type="text/javascript" src="/lib/js/jquery.js"></script>
  <script type="text/javascript" src="/lib/js/base.js"></script>
  <script type="text/javascript" src="test.js"></script>
</head>
<body>
  <!-- 这是内容 -->
  <!-- content -->
  <script type="text/javascript" src="index.js"></script>
  <script type="text/javascript" src="banner.js"></script>
</body>
</html>
```







Front-end

Js

## 2.2 JavaScript 注释方式

```
<script>  
    // 这是注释，单行注释    （注释内容不可以换行）  
    /*  
    多行/块级注释：  
    这是多行注释  
    注释内容可以换行  
    */  
    // 弹出一个提示窗口  
    alert('这是什么');  
</script>
```



## 2 Front-end Js

### 2.3 JavaScript 标识符

所谓标识符就是一个名字。

如：变量，函数，属性的名字或函数的参数。

标识符的命名规则：

- 第一个字符**必需**是一个字母、下划线或一个美元符号 (\$)；
- 其他字符可以是字母、下划线、美元符或数字（数字是绝对不允许出现在首位）
- 不能包含空格
- 不用以关键词和保留字命名
- 写法都推荐使用驼峰大小写格式

如：doSomething、myBook、appStore、sublimeText





### 3.1 JavaScript 关键字和保留字

在JavaScript中不能被用作标识符（变量名、函数名以及循环标记）

全部关键字:

break、case、catch、continue、default、delete、do  
else、finally、for、function、if、in、instanceof  
new、return、switch、this、throw、try、typeof  
var、void、while、with



# 3 Front-end Js

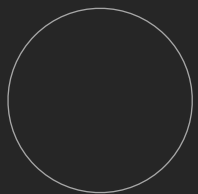
## 3.1 JavaScript 关键字和保留字

在JavaScript中不能被用作标识符（变量名、函数名以及循环标记）

保留字:（目前未被定为特定用途）

abstract、boolean、byte、char、class、const  
debugger、double、enum、export、extends  
final、float、goto、implements、import  
int、interface、long、native、package  
private、protected、public、short、static  
super、synchronized、throws、transient、volatile





Front-end

# 作业

- 1、Js组成
- 2、Js引入方式



行动是成功的阶梯，行动越多，登得越高  
成功的法则极为简单，但简单并不代表容易  
动手写代码是提高自己的最快捷途径

# THANK YOU

@Taching 1006082685