张东

zhangdong@tedu.cn

书: 犀牛书(JavaScript权威指南)

微信公众号: 前端大全

上一届笔记

正课:

1. 什么是JavaScript

2. 如何使用JavaScript

3. \*变量

4. \*数据类型

谁记得笔记越多，谁学的越烂！

1. 什么是JavaScript:

前端三大语言:

HTML: 专门编写网页内容的语言

CSS: 专门编写网页样式的语言

问题: 使用HTML和CSS做出的网页，只能看不能用

——静态语言! 静态页面!

解决: 凡是用HTML和CSS做出的静态页面都要添加交互行为后，才能让用户使用。

什么是交互: 3步:

1. 用户输入数据/执行操作

2. 程序接受并处理数据/响应用户的操作

3. 程序返回处理结果

JavaScript: 专门为网页添加交互行为的语言

为什么: 只有JavaScript才能给网页添加交互行为

何时: 凡是HTML和CSS做出的静态页面，都要用JavaScript添加交互行为后，才能给用户使用

js能做什么:

1. 表单验证:

2. 客户端数据计算:

3. 动画效果/网页游戏:

第一个js程序:

如何添加js程序: 2处:

1. 在HTML文件中的<script>内

通常,script会放在body的结尾

问题: 放在html文件中的script，仅当前页面可用，无法重用！

解决:

2. 在独立的js文件中编写js程序

强调: js文件中，只包含js程序，不必添加<script>

再在页面中用<script src="url">引入

优: 一次定义处处使用，一次修改处处生效！

总结: 任何情况下，不要重复自己的代码(DRY原则)

发展历程:

ECMAScript: JavaScript -> ECMA -> ECMAScript标准

ECMAScript标准是规定js语言核心语法的标准

JavaScript: NetScape参照ES标准,制作的JavaScript语言版本

JScript: Microsoft参照ES标准，制作的JavaScript语言版本

ES3.1 存在很多问题

ES5, 6, 2016....

原生js: 不需要下载任何第三方文件，就可执行使用的js程序.

三大组成部分:

1. ECMAScript: 核心语法 7+3

比如: console.log("...")

变量的声明，赋值，取值...

2. DOM: 专门操作网页内容的程序 2

比如: document.write("...")

3. BOM: 专门操作浏览器窗口的程序 1.5

比如: alert("...")

2. 如何使用JavaScript:

JavaScript的开发环境和运行机制:

开发: 编写代码: WebStorm

运行机制: 所有JavaScript程序都要运行在js引擎中

什么是js引擎: 专门解析js程序，并执行计算程序的小软件。

如何运行js程序: 2种:

1. 浏览器中已经内置了js引擎程序

只要有浏览器，就可运行js程序

浏览器的组成: 2个小软件

内容排版引擎: 专门解析HTML和CSS程序，并绘制网页内容的小程序。

脚本解释引擎(js引擎): 专门解析js程序，并执行计算程序的小软件。

问题: js语言只能做前端，不能做后端和数据库

程序员要想完成整个网站必须学习多种语言

解决:

2. 将js引擎从浏览器中脱离出来，独立安装！

Node.js——专门在服务器端执行js程序的小软件

MongoDB——专门采用JS语法规范的数据库

Browser Server DB

js Node.js MongoDB

jQuery

NG...

js的基本语法:

1. 区分大小写

2. 所有字符串，必须放在引号中，但单双引号均可

问题: 如果字符串内容中也包含引号

解决: 将内部的引号和外部的引号分别用单双引号区分开。

3. 每条语句必须用分号结尾

4. 注释: 2种:

//单行注释

/\*

多行注释

\*/

调试js程序:

只要想要的效果没出来，就是出错了！

只要出错就，F12，开控制台！

控制台(console): 显示错误的原因和出错的位置

输出: 3种:

1. 向网页中输出内容:

document.write("输出的内容")

问题: 会影响网页布局

解决:

2. 弹出警告框: alert(“提示内容”)

其实对话框已经被浏览器做好，藏在程序中.

alert()只是显示出来而已。

问题: 1. 模态: 阻止用户操作页面

2. 样式固定的，无法修改！

解决:

3. 输出到控制台: console.log(“输出的内容”)

在 控制台 记录 一句话

控制台的使用:

什么是: 编写,调试并查看程序输出的小窗口

为什么: 即不影响页面，又不阻碍用户操作.

何时: 只要调试程序或查看小程序的输出，都要在控制台执行。

如何:

1. 每输入一条语句，按回车，执行

2. 控制台中积累的内容不想要了，可点漏斗左边圆形叉号，清空控制台.

3. 多行，批量执行: shift+enter 不执行，仅换行

4. 如果写过的程序，想修改: 可用上下键切换已经执行过的程序

3. \*变量:

什么是: 内存中保存\*一个\*数据的存储空间，再起一个名字

为什么: 程序中的数据，默认都只能使用一次

只有保存在变量中的数据，才能被反复使用

何时: 只要一个数据，可能被反复使用时，都要先保存在变量中。

如何: 3件事:

1. 声明:

什么是: 在内存中创建一个变量，再起一个名字.

何时: 所有变量，在使用前，必须先声明

如何: var 变量名;

创建

变量名的要求:

1. 字母,数字,\_和$组成,不能以数字开头!

2. 不能使用保留字:

保留字: js语言中已经提前使用的，有特殊意义的词

3. 见名知义

4. 驼峰命名: 首字母小写,之后每个单词首字母大写！

width, fontSize,backgroundColor,listStyleType

强调: 仅声明，暂时未赋值的变量其中，保存了默认值undefined.

简写: 一个var可同时声明多个变量:

var 变量1,变量2,....;

2. 赋值:

什么是: 将等号右边的值保存到等号左边的变量中。

如何: 变量名=值;

强调: 只有等号=才能改变变量中的值

特殊: 强行给未声明过的变量赋值:

结果会自动创建该变量——广受诟病的缺陷

容易造成误会!

解决: 禁止给未声明的变量赋值!

如何: ES5: 严格模式:

什么是: 比普通js要求更严格的运行模式:

规定: 禁止给未声明的变量赋值!

如何: 在当前代码的顶部: "use strict";

总结: 今后所有项目，都应该在严格模式下运行

简写: 声明同时，就初始化变量的值——强烈建议

var 变量名=值;

声明 赋值

更简化: 如果同时声明并初始化多个变量:

var 变量名1=值1,

变量名2=值2,

... = ... ;

鄙视: var a , b=2; console.log(a,b);//undefined 2

解决: 每个变量一换行

3. 取值: 程序中，任何位置，使用变量名，等效于直接使用变量中的值！

特殊: 尝试使用未声明过的变量——报错!

ReferenceError: 引用错误!

凡是看到引用错误,都说明要用的变量没找到

常量: 一旦创建，值不能改变的量

何时: 如果一个值，一旦创建，不允许随意修改时

如何: const 常量名=值;

强调: 1. 常量在声明时，必须赋值

2. 习惯: 常量名要全大写!

问题: 传统ES标准: 强行修改常量的值:

即不报错，也不让修改——静默失败

解决: ES5: 严格模式:

规定: 将所有的静默失败升级为错误！

4. \*数据类型:

什么是: 数据在内存中的存储结构

为什么: 因为不同类型的数据，可执行的操作，不一样.

不一样的操作，就要求不同的存储结构支持。

包括: 2大类:

1. 原始类型: 值直接保存在变量本地的数据类型

包括5种: number, string, boolean, null, undefined

2. 引用类型: 值无法保存在变量本地的数据类型

number: 专门存储数字的类型

何时: 只要用于计算或比较的数字都要存为number类型——不要加引号

比如: 价格, 数量, 分数, 年龄, ...

反例: 手机号, QQ号,身份证号

如何: 凡是程序中不加引号的数字，自动就是number类型.

存储空间: js中每个数字占8字节

整数占4字节

1字节(byte)=8位二进制数(Bit)

1KB=1024byte

1MB=1024KB

1GB=1024MB

number类型的数值大小和所占存储空间无关!

将十进制转为二进制: n.toString(2)

string:

boolean:

null/undefined: