

JavaScript入门

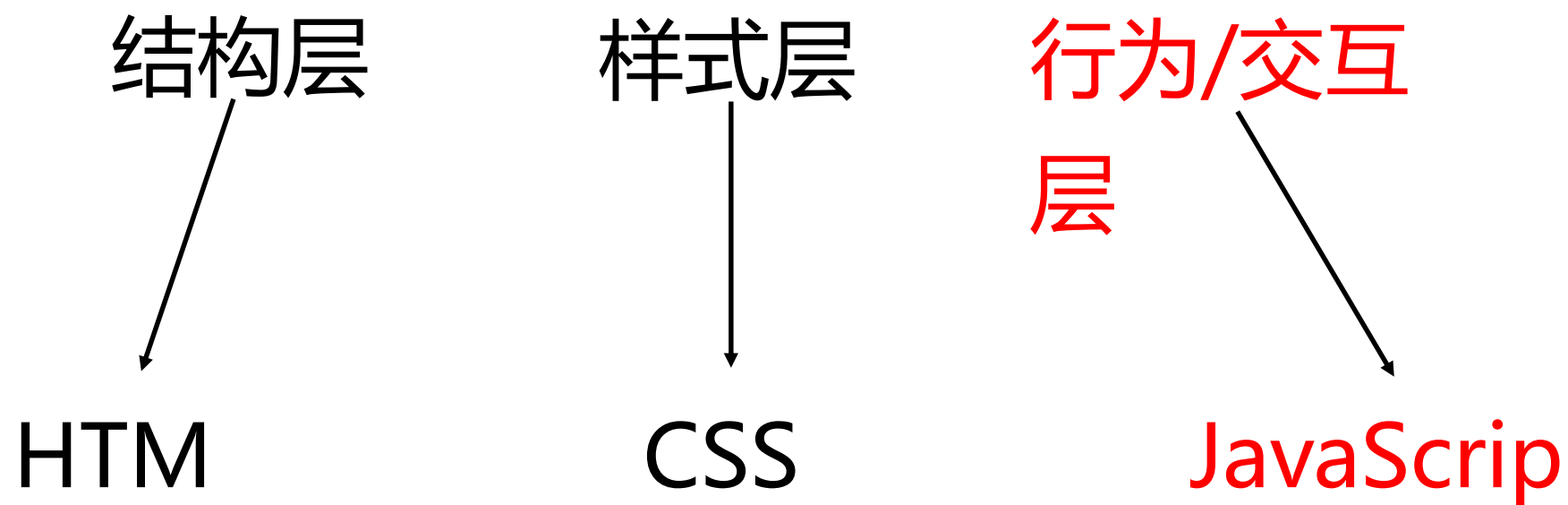


- 了解JavaScript历史
- 能够说出JavaScript的几种引入方式
- 掌握如何对JavaScript代码进行行注释与块注释
- 掌握什么是变量以及变量的应用
- 掌握字面量的概念

- JavaScript 简介
- 第一个JS程序(alert 弹窗)
- 控制台console
- JavaScript行文特性
- JavaScript 注释
- JavaScript的引入方式
- JavaScript标识符、关键字和保留字
- JavaScript变量
- JavaScript字面量

WEB 标准构成

前端三层



JavaScript 用来制作 WEB 页面交互效果，提升用户体验。

| | | | |
|--|---|--|--|
| <p>轮播图</p>  | <p>选项卡 TAB 栏</p>  | <p>Ajax 表单验证</p>  | <p>炫酷 HTML5 页面</p>  |
|--|---|--|--|

1995年Netscape推出JavaScript

随后几个月，微软推出了JScript

1997 ECMAScript

欧洲计算机制造商协会（ECMA）

因为JavaScript基于ECMA标准，因此也称为ECMAScript

参考网址：

http://www.w3school.com.cn/js/pro_js_history.asp

◆ JavaScript特点

- 直译式脚本语言（代码不用预编译，在程序的运行过程中逐行进行解释）
- 一种动态类型、弱类型、基于原型的语言
- 它的解释器被称为JavaScript引擎，为浏览器的一部分，广泛用于客户端的脚本语言，在HTML网页上使用，用来给HTML网页增加动态功能，能在浏览器中实时浏览到交互效果。

兼容性问题： IE6、7、8是个怪胎，对JavaScript语法有兼容性问题，所以要写兼容写法。

花式写法很多，抽象：从简单入手，细细品味代码

◆ 学习方法

- 要多去“品”程序，多去思考内在逻辑，读懂每一行代码！
- JS 机械重复性的劳动几乎为 0，基本都是创造性的劳动。HTML、CSS 都是重复的劳动，margin、padding 挤来挤去。
- 永远不要背程序，每一个程序都必须自己会写，有一个每天的作业，重复老师的上课案例。

◆ alert ()

alert 就是英语中“警报”的意思，能够让页面弹出警告框。
让浏览器弹出一个窗口，窗口里的内容就是alert()中的内容。

```
<script>
    alert("Hello World");
</script>
```

例1-1



JS程序是顺序执行的。

```
<script>
    alert("Hello World");
    alert("床前明月光");
    alert("疑似地上霜");
    alert("举头望明月");
    alert("低头思故乡");
</script>
```

按顺序执行

◆ console.log()

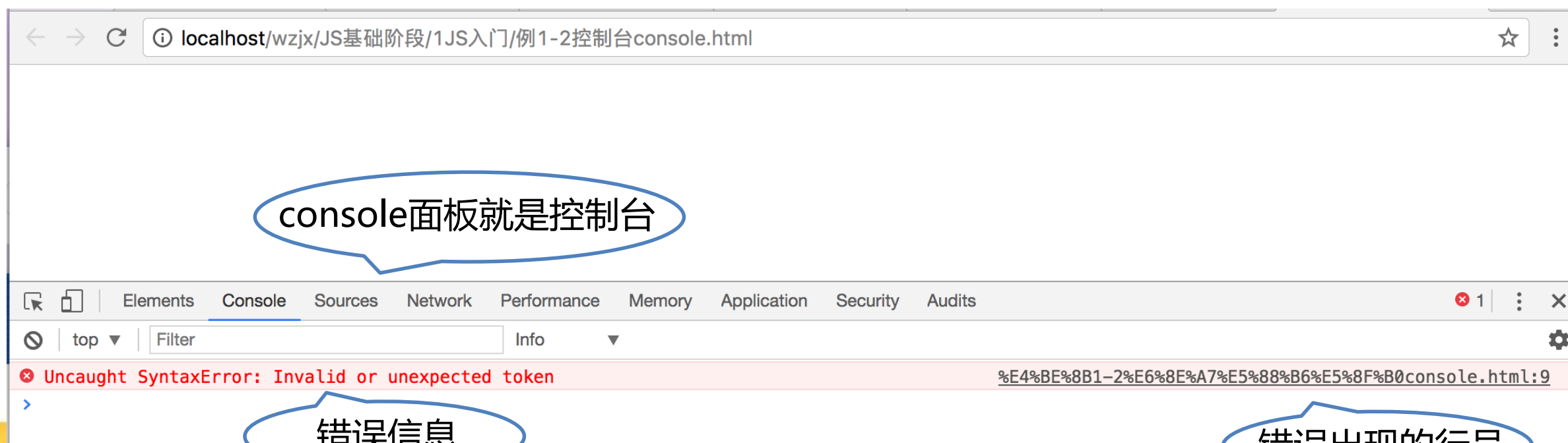
控制台是浏览器“检查”或者“检查元素”里面的功能，快捷键是F12。英文叫做 console。

程序运行后所产生的错误，都会在控制台中输出，控制台是调试程序的一个利器。

比如右边的语句用了中文的符号，有语法错误：

```
<script>  
    alert(“你好”);  
</script>
```

例1-3



◆ console.log()

有些时候我们想在控制台中自己输出点什么，那么就要使用语句：

```
console.log();
```

console 就是控制台，log 就是日志的意思，这里是动词，表示打印。

```
console.log( "你好啊");
```

要养成打开控制台调试程序的习惯哦

◆ JavaScript语句

`alert("你好啊! ");` 这是一条语句

JS中的每条语句以分号 ; 结尾。

如果语句是一行一行写的，没有分号结尾也是有效语句。但项目做好后，通常都会进行压缩，用软件把所有的空格、换行都去掉。此时，语句末尾的分号显得非常重要，如果去掉分号，将不能执行。

注意：所有的符号都是英文的符号，不要用中文

```
<script type="text/javascript">  
  alert("你好")alert("帅")alert("哈哈哈哈哈")  
</script>
```



例1-4

```
<script type="text/javascript">  
  alert("你好");alert("帅");alert("哈哈哈哈哈");  
</script>
```



◆ JavaScript空格、换行

JavaScript 语句和语句之间的换行、空格、缩进都不敏感，会忽略多余的空格。

可以向脚本添加空格缩进，来提高其可读性。

```
<script type="text/javascript">  
alert("你好");alert("帅");  
  
    alert("哈哈哈哈哈");  
</script>
```

等价于



```
<script type="text/javascript">  
    alert("你好");  
    alert("帅");  
    alert("哈哈哈哈哈");  
</script>
```

◆ 单行注释

// 注释内容 不可换行

```
<script type="text/javascript">  
    //我是一行注释  
    //再来一行  
    alert("你好");  
</script>
```

◆ 块级/多行注释

/* 注释内容
可以换行 */

```
<script type="text/javascript">  
    /*  
        我是一行注释  
        再来一行  
        再来一行  
    */  
    alert("你好");  
</script>
```

例1-5

注释用于提高代码的可读性

编辑器里注释的快捷键一般是：ctrl+/
注意：注释的符号，不能嵌套

◆ 内部书写

在HTML文件中直接进行代码的书写，JS 写在<script>标签里面，type属性就是类型，text/javascript 就是纯文本的JavaScript。

注意：这个 type必须写准确，可以不写，但是要写，一定要写准确，否则失效。

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
  <script type="text/javascript">
    alert("hello world1");
  </script>
</head>
<body>
  <script type="text/javascript">
    alert("hello world2");
  </script>

  <div>你好啊,我是HTML</div>

  <script type="text/javascript">
    alert("hello world3");
  </script>
</body>
</html>
```

例1-6

◆ 外部引入

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
  <script type="text/javascript" src="js/app.js"></script>
</head>
<body>
<script type="text/javascript" src="js/app.js"></script>

<div>你好啊,我是HTML</div>

<script type="text/javascript" src="js/app.js"></script>
</body>
</html>
```

例1-6

◆ 直接写在HTML标签内（不推荐使用）

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
</head>
<body>
<div onclick="alert('hello world')">
  点我
</div>
</body>
</html>
```

例1-6

◆ 标识符

就是一个名字，用来对变量、函数、属性等进行命名。

- 标识符命名规范：

第一个字符必须是一个字母、下划线（_）或一个美元符号（\$）；

其他字符可以是字母、下划线、美元符号或数字；

不能含有空格；

不能以关键字或保留字命名。

- 关键字、变量、函数名及所有的标识符都区分大小写

比如：变量ABC； 和 变量abc；

指的是一个变量吗？

◆ 关键字

关键字可用于表示控制语句的开始或结束，或者用于执行特定操作等。Javascript 中的很多内置功能用**关键字**标注起来，便于程序员调用。按照规则，关键字也是语言保留的，**不能用作标识符**。

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

◆ 保留字

保留字有可能在将来被用作关键字来使用，**不能用作标识符。**

| | | | |
|----------|-----------|--------|---------|
| abstract | int | short | boolean |
| export | interface | static | byte |
| extends | long | super | char |
| final | native | class | float |
| throws | const | goto | private |
| double | import | public | |

◆ 变量

变量是存储信息的容器。

JavaScript的变量为弱变量，可以用来保存任何类型的数据。

C语言

int定义整型数据

char定义字符型

float定义浮点型

.....

JavaScript

在JS中，只有var这一种声明变量的类型，它可以声明各种类型的数据

◆ 变量

定义变量——使用关键字 **var** 声明。var 就是英语 variables 变量的缩写

var a;

定义一个变量a，电脑内存中，就会开辟一个空间，来存储这个变量 a。

可以给这个变量赋值，**JS 中给变量赋值用等号，等号右边的值赋给左边。**

```
var a=20;  
console.log(a);
```

可以通过**var**声明多个变量，变量与变量之间以**,**逗号隔开

```
var a=20,b=100;  
console.log(a,b);
```

◆ 变量的命名规范

第一个字符**必须**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、美元符号或数字；

不能含有空格；

不能以关键字或保留字命名。

● 变量名称命名练习

对于变量的**合法命名**是哪几项？

1. var div;

2. var for;

3. var var;

4. var content;

5. var if;

6. var int;

● 变量名称命名练习

对于变量的**不合法命名**是哪几项？

1. var abc;

2. var _abc%;

3. var ABC;

4. var do;

5. var 2abc;

6. var abc2;

7. var int;

8. var \$abc;

9. var a b c;

◆ 变量须先声明

使用一个变量，必须先进行一个 var，才能使用。 var 这个过程可以叫做声明 declaration，也可以叫做定义 definition。

直接运行语句：

```
console.log(b); //这个 b 没有通过var声明，所以要报错
```

A screenshot of a JavaScript error message in a console. It shows a red 'x' icon followed by the text 'Uncaught ReferenceError: b is not defined'.

例1-7

◆ 变量的赋值

变量的赋值用等号，等号就是赋值符号

```
var a;      // 定义
a = 100;    // 赋初值
console.log(a); // 输出: 100
```

```
var a = 100; // 定义和赋初值一起写
console.log(a); // 输出: 100
```

```
var a = 100;
var b = 200;
// 改变 a 变量的值, a 的值变为 200。b 的值不变。
a = b;
console.log(a); // 输出: 200
console.log(b); // 输出: 200
```

当声明了一个变量，想改变该变量值时，不需要 var 重新声明

```
var a = 100;
a = 50;
console.log(a); // 输出: 50
```

一个变量，通过var声明但未被赋初值时，变量的值为 **undefined**;

```
var a;
console.log(a); // 输出 undefined
```

已经通过var定义了这个 a，但浏览器是这样解析：如果这个变量没有被赋初值，那么这个变量就视为没有“定义完成”。值就是 undefined。

◆ 变量声明的提升

```
a = 100;  
var a; // 这行代码会自动提升到所有语句之前  
console.log(a); // 输出: 100
```

注意: JS 只能提升变量的声明, 而不能提升变量的赋初值

| | | |
|--|-----|---|
| console.log(a); var a = 100; // 输出: undefined | 等价于 | var a; //自动提升 console.log(a); a = 100; //赋初值还留在原地 |
|--|-----|---|

不写 var 的情况

```
a = 123;  
console.log(a); // 输出: 123
```

没有通过var定义变量, 程序没有报错, 正常输出值。

但用var和不用var定义变量是有区别的, 作用域那一节再具体讲述。

◆ 练习

声明一个变量a, 值为: 3

声明一个变量b, 值为: null

声明一个变量c, 值为: "Hello!"

声明一个变量d, 值为: true

声明一个变量e, 不赋值

声明一个变量f, 值为 ""



思考

以上声明的变量分别是什么数据类型?

◆字面量

在计算机科学中，字面量 (literal) 是用于表达源代码中一个固定值的表示法 (notation)。由字母，数字等构成的字符串或者数值，它只能作为右值出现，所谓右值是指等号右边的值。

字面量分为：字符串字面量(string literal)、数字字面量、数组字面量(array literal)和对象字面量(object literal)，另外还有函数字面量(function literal)。

```
var a=1;           // 1是数字字面量
var b= "你好" ;    // " 你好 " 是字符串字面量
var c=[1,2,3];      // [1,2,3]是数组字面量
var d={ "age" : " 10" } // { "age" : " 10" }是对象字面量
function myFunction(a, b) { return a * b;} // 函数字面量
```

◆ 区分变量还是字面量

输出变量的时候，不能加引号。如果变量加上引号，就变成字符串了。

```
var a = 100;  
console.log( "a"); // 这里输出a，而不是输出100
```

引号是“字符串”的定界符，在双引号里面的 a 已经失去了变量的意思，代表的就是一个字符串 a。

• 练习

```
var a;  
var b=3;  
console.log( 'a' );  
console.log( a );  
console.log( b );  
console.log( 123 );
```