

一、准备工作

1、web （网页）

web 页面

结构 html

表现 css

行为 javascript (JS)

基础部分

高级部分

三驾马车 【HTML - CSS - JS 】

HTML - CSS 不是编程语言

JS 是编程语言

客户端：浏览器

服务端：存放资源的环境

2、JS (javascript)

爪哇脚本

ECMA (制定标准的联盟组织)

javascript 简称 JS

JS 是一门动态的编程语言（弱类型的脚本语言）

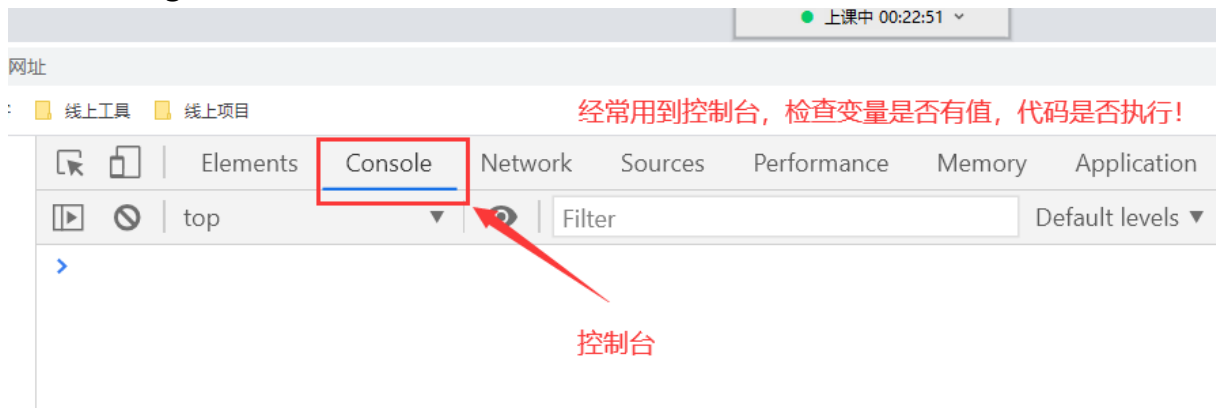
```
1 关键字 var
2  var a = "hello world";
3  console.log(typeof a ); // string
4
5  a = 100;
6  console.log(typeof a ); //number
```

```
7
8 a = true;
9 console.log(typeof a ); // boolean
10
11
```

JS 可以提高页面的交互效果（提升用户体验）

学会使用控制台检查代码错误

console.log("test")



JS 的书写

第一种方式：写在html文档中

```
1 <!--如果script标签放在head标签内，那么需要执行onload事件，才能获取页面元素-->
2 <script>
3 //设置文档对象下body标签的背景色 粉色
4 //document.body.style.backgroundColor = "pink";//报错
5 //console.log(document.body);// 获取body标签
6 // 加载事件(加载页面所有内容完毕)
7 window.onload = function(){
8 document.body.style.backgroundColor = "pink";
9 }
10 //页面window.onload 只执行一次
11 window.onload = function(){
12 document.body.style.backgroundColor = "yellowgreen";
13 }
14 </script>
```

第二种方式：写标签内

```
1 <!--onclick 点击事件 -->
2 <button onclick="console.log('hello world!')">点击按钮</button>
```

第三种方式：引入外部脚本

```
1 <!--引入外部的脚本文件-->
2 <script src="./demo-3.js"></script>
3 <!--通常脚本放在标签后面-->
4 <!--因为浏览器工作顺序：从上往下执行的-->
5 <!--等html标签渲染完毕，执行脚本方可控制元素的样式-->
6 <!--初学者：推荐第一和第三种方式-->
7 <!--JS 执行顺序 从上往下 -->
8
```

JS 是一种单线程的脚本语言

那么执行的顺序是自上往下的。

```
1 console.log(1)
2 console.log(2)
3 console.log(3)
4 1
5 2
6 3
```

JS 知识点列表（由简单的往复杂的过渡）

```
1 变量
2 数据类型
3 运算符
4 DOM
5 语句 判断语句 控制流语句 循环语句
6 事件
7 函数 作用域 返回值 参数
8 BOM
9 递归
10 闭包
11 字符串
12 数组
13 对象
14 数学对象
15 日期对象
16 正则表达式
17 数据深浅拷贝
18 面向对象 直接量 工厂模式 构造函数 原型
19 ajax 数据前后端交互
20
21
```

JS 注释

```
1 // 注释
2 /* 注释 */
3 注释是给开发者看的
```

二、如何学习Javascript

1、变量

```
1 var 关键字
2 var str = "hello world";
```

2、数据类型

基本数据类型

```
1 string 字符串
2 number 数字
3 boolean 布尔值
4 undefined 未定义
5 null 空
6 symbol 唯一 ES6
```

引用数据类型

```
1 object 对象
2 array 数组
3 function 函数
```

3、数据类型的转换

```
1 // 数据类型: string number boolean undefined null symbol
2 // "string"
3 // undefined
4
5 // 构造函数 String Number Boolean
6 //1.0 显式转换
7 var str = "99";
8 console.log(typeof str);// string
```

```
9  var num = Number(str);
10 console.log(typeof num);// number
11
12 var str2 = String("100");
13 console.log(typeof str2);// string
14
15 var bool = Boolean("1");
16 console.log(bool);// true
17 console.log(typeof bool);// boolean
18 console.log(Boolean(""));
19
20 //例如:
21 //数据类型没有转换
22 var a = "100";
23 var b = 99;
24 console.log(a + b);// 10099 字符串拼接
25
26 //数据类型有转换
27 var a1 = "100";
28 var b1 = 99;
29 console.log(Number(a1) + 99);// 199 进行计算
30
31 console.log("-----")
32 //转换的作用: 相同的数据类型才可以进行计算
33 //2.0 隐式转换
34 // 计算机自动转换
35 // 通过运算符转换
36 // 任何数减 0 等于其自身
37 var count = "98";
38 console.log(count);// 字符串
39 console.log(count - 0 );// 数字
40 var res = count - 0;
41 console.log(typeof res);// number
42 // -
43 var str3 = 100;
44 console.log(str3);// 数字
45 console.log(str3+"");// 字符串
46 var res2 = str3+"";
47 console.log(typeof res2);//string
48 var res3 = true + "";
49 console.log(typeof res3);// string
```

```

50 // + 除了可以做加法运算 ， 还可以做字符串拼接
51
52 //通过语句转换
53 //if 如果
54 //if(条件){ 条件成立，执行代码 }
55
56 var res4 = true;
57 //if(res4 === true){
58 if(res4){
59 console.log("条件成立11")
60 }
61
62 // var res5 = "";
63 // var res5 ;
64 // var res5 = 100 ;
65 var res5 = "hello world";
66 if(res5){
67 console.log("条件成立22")
68 }
69
70 console.log("-----")
71 // 1 == true
72 // 0 == false
73 // "" == false
74 // [] == false
75
76 //记住有哪些隐式转换
77 //查字典
78 // 在JS中 0 表示 false , 1 表示true
79 console.log(1 == true); // true
80 console.log(0 == false); // true
81 console.log("" == false); // true
82 console.log([] == false); // true
83 console.log("" == []); // true
84 console.log(undefined == null); //true

```

4、运算符

算术运算符

```

1 // + 加
2 // - 减

```

```
3 // * 乘
4 // / 除
5 // % 取余 （取余数）
6 23 % 10 = 2.....3
```

比较运算符

```
1 // > 大于号
2 // < 小于号
3 // = 赋值 （不是做比较的）
4 // == 值比较
5 console.log("99" == 99); // true
6 // === 值、数据类型比较
7 console.log("99" === 99); // false
8 // <= 小于等于
9 // >= 大于等于
```

逻辑运算符

```
1 //只需要一个条件成立，结果为true（真）
2 console.log( 1 < 2 || 2 > 3 ); // true
3 console.log( 1 < 0 || 2 > 3 ); // false
4 console.log( 1 < 0 || 2 > 3 || true ); // true
5 console.log( 1 < 0 || 2 > 3 || 9 < 10 ); // true
6
7 console.log("-----")
8 //所有条件都需要成立，结果为true（真）
9 console.log(1 < 2 && 2 < 3 ); // true
10 console.log(1 < 2 && 2 < 2 ); // false
```

5、特殊数值（NaN）

```
1 // NaN (not a number) 是一种特殊的数值（非数字）
2 // NaN 不是一种数据类型
3 console.log("11aa" - 0); // NaN
4 // 非数字字符串 减 数字，就会出现 NaN
5
6 // isNaN()
7 // 改函数用于检测当前数据是否为非数字
8 console.log(isNaN("11aa" - 0)); // true 表示其为非数字
9 console.log(isNaN(100)); // false 如果是数字 返回值是false
10 console.log(isNaN("hello")); // true 不是数值
```

6、简单dom操作【DOM】

获取元素的方法

```
1 var box1 = document.getElementById("box1");  
2 console.log(box1);// div#box1标签
```

设置元素的文本

```
1 box1.innerText = "hello";
```

设置元素的样式

```
1 box1.style.backgroundColor = "green";  
2 box1.style.width = "200px";  
3 box1.style.height = "400px";
```