**一、准备工作**

**1、web （网页）**



**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**

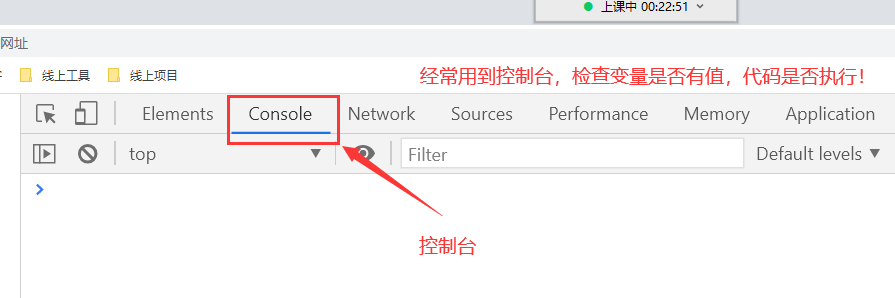
**JS 是一门动态的编程语言（弱类型的脚本语言）**

关键字 var
var a = "hello world";
console.log(typeof a ); // string
a = 100;
console.log(typeof a ); //number
a = true;
console.log(typeof a ); // boolean

**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**

<!--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-->
<script>
//设置文档对象下body标签的背景色 粉色
//document.body.style.backgroundColor = "pink";//报错
//console.log(document.body);// 获取body标签
// 加载事件(加载页面所有内容完毕)
window.onload = function(){
document.body.style.backgroundColor = "pink";
}
//页面window.onload 只执行一次
window.onload = function(){
document.body.style.backgroundColor = "yellowgreen";
}
</script>

**第二种方式：写标签内**

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

**第三种方式：引入外部脚本**

**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。

console.log(1)
console.log(2)
console.log(3)
1
2
3

**JS 知识点列表（由简单的往复杂的过渡）**

变量
数据类型
运算符
DOM
语句 判断语句 控制流语句 循环语句
事件
函数 作用域 返回值 参数
BOM
递归
闭包
字符串
数组
对象
数学对象
日期对象
正则表达式
数据深浅拷贝
面向对象 直接量 工厂模式 构造函数 原型
ajax 数据前后端交互

**JS 注释**

// 注释
/\* 注释 \*/
注释是给开发者看的

**二、如何学习Javacript**

**1、 变量**

var 关键字
var str = "hello world";

**2、数据类型**

基本数据类型

string 字符串
number 数字
boolean 布尔值
undefined 未定义
null 空
symbol 唯一 ES6

引用数据类型

object 对象
array 数组
function 函数

**3、数据类型的转换**

// 数据类型： string number boolean undefined null symbol
// "string"
// undefined
// 构造函数 String Number Boolean
//1.0 显式转换
var str = "99";
console.log(typeof str);// string
var num = Number(str);
console.log(typeof num);// number
var str2 = String("100");
console.log(typeof str2);// string
var bool = Boolean("1");
console.log(bool);// true
console.log(typeof bool);// boolean
console.log(Boolean(""));
//例如：
//数据类型没有转换
var a = "100";
var b = 99;
console.log(a + b);// 10099 字符串拼接
//数据类型有转换
var a1 = "100";
var b1 = 99;
console.log(Number(a1) + 99);// 199 进行计算
console.log("------------------------")
//转换的作用：相同的数据类型才可以进行计算
//2.0 隐式转换
// 计算机自动转换
// 通过运算符转换
// 任何数减 0 等于其自身
var count = "98";
console.log(count);// 字符串
console.log(count - 0 );// 数字
var res = count - 0;
console.log(typeof res);// number
// -
var str3 = 100;
console.log(str3);// 数字
console.log(str3+"");// 字符串
var res2 = str3+"";
console.log(typeof res2);//string
var res3 = true + "";
console.log(typeof res3);// string
// + 除了可以做加法运算 ，还可以做字符串拼接
//通过语句转换
//if 如果
//if(条件){ 条件成立，执行代码 }
var res4 = true;
//if(res4 === true){
if(res4){
console.log("条件成立11")
}
// var res5 = "";
// var res5 ;
// var res5 = 100 ;
var res5 = "hello world";
if(res5){
console.log("条件成立22")
}
console.log("------------------------")
// 1 == true
// 0 == false
// "" == false
// [] == false
//记住有哪些隐式转换
//查字典
// 在JS中 0 表示 false , 1 表示true
console.log(1 == true); // true
console.log(0 == false); // true
console.log("" == false); // true
console.log([] == false); // true
console.log("" == []); // true
console.log(undefined == null);//true

**4、运算符**

**算术运算符**

// + 加
// - 减
// \* 乘
// / 除
// % 取余 （取余数）
23 % 10 = 2.....3

**比较运算符**

// > 大于号
// < 小于号
// = 赋值 （不是做比较的）
// == 值比较
console.log("99" == 99);// true
// === 值、数据类型比较
console.log("99" === 99);// false
// <= 小于等于
// >= 大于等于

**逻辑运算符**

**5、特殊数值（NaN）**

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

**6、简单dom操作【DOM】**

**获取元素的方法**

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

**设置元素的文本**

box1.innerText = "hello";

**设置元素的样式**

box1.style.backgroundColor = "green";
box1.style.width = "200px";
box1.style.height = "400px";