**1.什么是数据：**文字，图片，视频，色香味等都是数据 光和热也是数据。

**2.javascript数据类型有几种，**

JavaScript中有5种简单数据类型（也称为基本数据类型）：Undefined、Null、Boolean、Number和String。还有1种复杂数据类型——Object，Object本质上是由一组无序的名值对组成的。

Undefined类型：Undefined类型只有一个值，即特殊的undefined。在使用var声明变量但未对其加以初始化时，这个变量的值就是undefined

例：

var message;

alert(message == undefined) //true

### Null类型：Null类型是第二个只有一个值的数据类型，这个特殊的值是null。从逻辑角度来看，null值表示一个空对象指针，而这也正是使用typeof操作符检测null时会返回"object"的原因

例：

### var car = null;

alert(typeof car); // "object"

Boolean类型：该类型只有两个字面值：true和false。这两个值与数字值不是一回事，因此true不一定等于1，而false也不一定等于0。虽然Boolean类型的字面值只有两个，但JavaScript中所有类型的值都有与这两个Boolean值等价的值。要将一个值转换为其对应的Boolean值，可以调用类型转换函数Boolean()；

例：

var message = 'Hello World';

var messageAsBoolean = Boolean(message);

var message = 'Hello World';

if(message)

{

alert("Value is true");

}

### Number类型：这种类型用来表示整数和浮点数值，还有一种特殊的数值，即NaN（非数值 Not a Number）。这个数值用于表示一个本来要返回数值的操作数未返回数值的情况（这样就不会抛出错误了)

例

alert(NaN == NaN); //false

### String类型：String类型用于表示由零或多个16位Unicode字符组成的字符序列，即字符串。字符串可以由单引号(')或双引号(")表示。

例：

var str1 = "Hello";

alert(str1.length); //输出5

### Object类型：对象其实就是一组数据和功能的集合。对象可以通过执行new操作符后跟要创建的对象类型的名称来创建。而创建Object类型的实例并为其添加属性和（或）方法，就可以创建自定义对象

例：

　var o = new Object();

1. **javascrip性能优化**

例：

Var arr=[1,2,3,4,5];

for(i=0;i<arr.length;i++){

Document.write(arr[i]);

}

//优化后

Var x=arr.length;

Var html;

For(var i=0;i<=x;i++){

Html=html+arr[i];

}

扩展：javascript语言与html标签执行的优先级：就是那个在前面先执行那个。

javascript写到当前文档里和用js文件加载那个优先级高：js应写在body下面。

1. **javascript排错有几种？**
2. ）语法错误. 解决：一般语法错误编辑器都会有提示报错。
3. )运行异常. 解决：可以打开浏览器开发者工具进行调试。

（3.）语义化错误.(就是没有按照我们的想法) 解决：一般比较难，比较逻辑判断，可以用 try{}cacth(e){alert(e.name+e.message);}

**5.javascript继承**

什么是继承：指一个对象直接使用另一对象的属性和方法，并扩展定义自己特有的属性，

继承方式

1.原型链继承

function Person(){

this.name="Mike";

this.sayGoodbye=function(){alert("GoodBye!");};

}

Person.prototype.sayHello=function(){alert("Hello");};

function Student(){}

Student.Prototype=new Person();

优点：从instanceof关键字来看，实例既是父类的实例，又是子类的实例，看起来似乎是最纯粹的继承。

缺点：无法实现多重继承。

2.利用构造函数继承

function Person (name, age) {

this.name = name;

this.age = age;

}

Person.prototype.say = function(){

console.log('hello, my name is ' + this.name);

};

function Man(name, age) {

Person.apply(this, arguments);

}

//Man.prototype = new Person('pursue');

var man1 = new Man('joe');

var man2 = new Man('david');

console.log(man1.name === man2.name);//false

man1.say(); //say is not a function

优点：可以实现多重继承，可以把子类特有的属性设置放在构造器内部。

缺点：使用instanceof发现，对象不是父类的实例。

3.组合继承

function Person (name, age) {

this.name = name;

this.age = age;

}

Person.prototype.say = function(){

console.log('hello, my name is ' + this.name);

};

function Man(name, age) {

Person.apply(this, arguments);

}

Man.prototype = new Person();

var man1 = new Man('joe');

var man2 = new Man('david');

console.log(man1.name === man2.name);//false

console.log(man1.say === man2.say);//true

man1.say(); //hello, my name is joe

4、继承的优点：新的实现很容易，因为大部分是继承而来的 很容易修改和扩展已有的实现

4.1，继承的缺点：

1，打破了封装，因为基类向子类暴露了实现细节

2，白盒重用，因为基类的内部细节通常对子类是可见的

3，当父类的实现改变时可能要相应的对子类做出改变

4，不能在运行时改变由父类继承来的实现

4.寄生组合继承

说实话我真不知道下面的这种形式叫这名字，但是它确实是最流行，最经典的javascript的继承方式。其实，只需要明白原型对象的结构即可：

function Person (name, age) {

this.name = name;

this.age = age;

}

Person.prototype.say = function(){

console.log('hello, my name is ' + this.name);

};

function Man(name, age) {

Person.apply(this, arguments);

}

Man.prototype = Object.create(Person.prototype);//a.

Man.prototype.constructor = Man;//b.

var man1 = new Man('pursue');

var man2 = new Man('joe');

console.log(man1.say == man2.say);

console.log(man1.name == man2.name);

1. **html4和html5区别**

**Html5与html4区别**

1. 在文档类型声明上：

HTML5化繁为简；

**a,以浏览器原生能力替代复杂的JAVASCRIPT代码；**

**b,简化了DOCTYPE；**

**c,简化的字符集声明；**

**d,简单而的强大的HTML5 API**

1. 在结构语义上：

Html5相对于html4新增了很多语义化标签对开发人员友好，有利于搜索引擎的索引整理

1. **HTML5 支持了不同类型的存储类型：**

HTML5 支持本地存储，在之前版本中是通过 Cookie 实现的。HTML5 本地存储速度快而且安全。并且HTML5有两种不同的对象可用来存储数据

1. **全新的表单**  
   HTML5对 <form> 和 <forminput> 标签进行了大量修改，添加了很多新的属性，也修改了很多属性
2. HTML5更安全;

HTML5引入了一种新的基地来源的安全模型，该模型不仅容易用，而且对各种不同的API都通用

1. HTML5将表现与内容分离

7.HTML5无插件范式；

原来的HTML4插件存在一些问题：

**a,插件安装可能失败；**

**b,插件可以被禁用或屏蔽；**

**c,插件自身有漏洞；**

**d,插件不容易与HTML文档的其他部分集成，因为插件的边界、剪裁和透明度问题；**

### **Dom**

Dom

HTML Dom中最常用的几个方法之查找元素 :

1.document :引用当前整个网页文档

2.getElementById :快速在网页中查找某一元素

3.getElementsByTagName :查找一组标签，具有相同名称的标签元素

4.getElementsByName :根据标签的Id属性值或name属性来查找多个元素

5.createElement() :创建一个具体的元素

6.createTextNode() :创建一个文本节点

7.childnodes :查找所有的子元素/获取所有的子节点

8.appendChild() :添加子节点

Dom中创建，添加，删除，修改，替换，复制，网页中的标签和文本内容的方法如下．

网页：创建【create】 删除：【remove】 替换：【replae】 复制：【clone】 修改或获取【inner】 文本【text】

添加或设置一个属性【set】 插入【insert】 添加【append】

表格：插入【insert】 删除【delete】 创建【create】 移动【move】 行【Row】 列【Cell】

**8.数组**

**什么是数组？**

所谓数组，就是相同数据类型的元素按一定顺序排列的[集合](http://baike.baidu.com/subview/15216/10703233.htm" \t "http://baike.baidu.com/_blank)，就是把有限个类型相同的变量用一个名字命名，然后用编号区分他们的变量的集合，这个名字称为数组名，编号称为[下标](http://baike.baidu.com/view/729047.htm" \t "http://baike.baidu.com/_blank)。组成数组的各个变量称为数组的分量，也称为数组的元素，有时也称为下标变量。数组是在程序设计中，为了处理方便， 把具有相同类型的若干变量按有序的形式组织起来的一种形式。这些按序排列的同类数据元素的集合称为数组。

**数组创建？**

Var a=[] var a=[(1,2)] var a=([1,2],[3,4]...)

Var a=array(1,2,3,4) var a=array(5) var a=new array()

Var a={[1,2],[3,4],[5,6],...} var x=document.GetElementsByname()

1. **跨域**

域名：url,资源统一定位器;网址：就是用此地址找到访问对应的网站网页。

同于策略：同端口，同域名，同协议。