

对象在JavaScript中被称为引用类型的值，而且有一些内置的引用类型可以用来创建特定的对象：

1. 引用类型与传统面向对象对象程序设计中的类相似，但实现不同（具体区别）
2. Object 是一个基础类型，其他所有类型都是从 Object 继承了基本的行为
3. Array 类型是一组值的有序列表，同时还提供了操作和转换这些值的功能

检测数组：

instanceof Array：当网页中包含多个框架时，会有多个不同的全局执行环境，可能会导致存在两个具有不同构造函数的数组

Array.isArray():ES5新增方法

转换方法：

toLocaleString ()

toString ()

valueOf ()

栈方法：(LIFO:Last-In-First-Out)

push():将任意数量的参数添加到数组末尾，返回修改后的长度

pop():从数组末尾移除最后一项，并返回移除的项

队列方法 (FIFO:First-In-First-Out)：

shift():移除数组第一项并返回该项

unshift():在数组前端添加任意个项并返回新数组的长度

重排序方法：

reverse ()：反转数组顺序

sort ()：接受一个比较函数(a,b)作为参数,a在前返回负数，b在前返回正数

操作方法

concat():创建当前数组的副本，将接收到的参数添加到副本的末尾，返回新构建的数组（数组深拷贝）

slice():

splice()

4. Date 类型提供了有关日期和时间的信息，包括当前日期和时间以及相关的计算功能
5. RegExp 类型是 ECMAScript 支持正则表达式的一个接口，提供了最基本的和一些高级的正则表达式功能

函数实际上是 Function 类型的实例，因此函数也是对象；而这一点正式 Javascript 最有特色的地方。由于函数是对象，所有函数也拥有方法，可以用来增强其行为

因为有了基本包装类型，所以 JavaScript 中的基本类型值可以被当作对象来访问。三种基本包装类型分别是：Boolean、Number 和 String。其共同特征为：

1. 每个包装类型都映射到同名的基本类型
2. 在读取模式下访问基本类型值时，就会创建对应的基本包装类型的一个对象，从而方便了数据操作
3. 操作基本类型值的语句一经执行完毕，就会立即销毁新创建的包装对象
4. 在所有代码执行之前，作用域中就已经存在两个内置对象：Global 和 Math。在大多数ECMAScript实现中都不能直接访问 Global 对象；不过，Web浏览器实现了承担该角色的 window 对象。全局变量和函数都是 Global 对象的属性。Math 对象提供了很多属性和方法，用于辅助完成复杂的数学计算任务。