

JavaScript 进阶 - 第2天

了解面向对象编程的基础概念及构造函数的作用，体会 JavaScript 一切皆对象的语言特征，掌握常见的对象属性和方法的使用。

- 了解面向对象编程中的一般概念
- 能够基于构造函数创建对象
- 理解 JavaScript 中一切皆对象的语言特征
- 理解引用对象类型值存储的特征
- 掌握包装类型对象常见方法的使用

知道解构的语法及分类，使用解构简洁语法快速为变量赋值。

解构赋值是一种快速为变量赋值的简洁语法，本质上仍然是为变量赋值，分为数组解构、对象解构两大类型。

数组解构

对象解构

数组解构是将数组的单元值快速批量赋值给一系列变量的简洁语法

总结：

- 赋值运算符 = 左侧的 [] 用于批量声明变量，右侧数组的单元值将被赋值给左侧的变量
- 变量的顺序对应数组单元值的位置依次进行赋值操作
- 变量的数量大于单元值数量时，多余的变量将被赋值为 undefined
- 变量的数量小于单元值数量时，可以通过 ... 获取剩余单元值，但只能置于最末位
- 允许初始化变量的默认值，且只有单元值为 undefined 时默认值才会生效
- 注：支持多维解构赋值，比较复杂后续有应用需求时再进一步分析

对象解构是将对象属性和方法快速批量赋值给一系列变量的简洁语法

总结：

- 赋值运算符 = 左侧的 {} 用于批量声明变量，右侧对象的属性值将被赋值给左侧的变量
- 对象属性的值将被赋值给与属性名相同的变量
- 对象中找不到与变量名一致的属性时变量值为 undefined
- 允许初始化变量的默认值，属性不存在或单元值为 undefined 时默认值才会生效
- 注：支持多维解构赋值，比较复杂后续有应用需求时再进一步分析

了解面向对象的基础概念，能够利用构造函数创建对象。

- 构造函数
- 字面量
- 自定义构造函数

构造函数是专门用于创建对象的函数，如果一个函数使用 new 关键字调用，那么这个函数就是构造函数。

总结：

- 使用 new 关键字调用函数的行为被称为实例化
- 实例化构造函数时没有参数时可以省略 ()
- 构造函数的返回值即为新创建的对象
- 构造函数内部的 return 返回的值无效！
- 注：实践中为了从视觉上区分构造函数和普通函数，习惯将构造函数的首字母大写。

通过构造函数创建的对象称为实例对象，实例对象中的属性和方法称为实例成员。

总结：

- 构造函数内部 `this` 实际上就是实例对象，为其动态添加的属性和方法即为实例成员
- 为构造函数传入参数，动态创建结构相同但值不同的对象
- 实例对象的 `constructor` 属性指向了构造函数
- `instanceof` 用于检测实例对象对应的构造函数
- 注：构造函数创建的实例对象彼此独立互不影响。

在 JavaScript 中底层函数本质上也是对象类型，因此允许直接为函数动态添加属性或方法，构造函数的属性和方法被称为静态成员。

总结：

- 静态成员指的是添加到构造函数本身的属性和方法
- 一般公共特征的属性或方法静态成员设置为静态成员
- 静态成员方法中的 `this` 指向构造函数本身

体会 JavaScript 一切皆对象的语言特征，掌握各引用类型和包装类型对象属性和方法的使用。

- 在 JavaScript 中**最主要**的数据类型有 6 种，分别是字符串、数值、布尔、undefined、null 和 对象，常见的对象类型数据包括数组和普通对象。其中字符串、数值、布尔、undefined、null 也被称为简单类型或基础类型，对象也被称为引用类型。
- 在 JavaScript 内置了一些构造函数，绝大部的数据处理都是基于这些构造函数实现的，JavaScript 基础阶段学习的 Date 就是内置的构造函数。
- 甚至字符串、数值、布尔、数组、普通对象也都有专门的构造函数，用于创建对应类型的数据。

复杂类型

- Object：是内置的构造函数，用于创建普通对象。
- Array：是内置的构造函数，用于创建数组。
- RegExp：内置的构造函数，用于创建正则表达式。

Object 是内置的构造函数，用于创建普通对象。

普通对象赋值后，无论修改哪个变量另一个对象的数据值也会相当发生改变。

总结：

- 推荐使用字面量方式声明对象，而不是 Object 构造函数
- Object.assign 静态方法创建新的对象
- Object.keys 静态方法获取对象中所有属性
- Object.values 静态方法获取对象中所有属性值

面试回答堆与栈的区别：

- 堆和栈是内存中的数据存储空间
- 简单类型的数据保存在内存的栈空间中
- 引用类型的数据保存在内存的堆空间中，栈内存中存取的是引用类型的地址（房间号）

Array 是内置的构造函数，用于创建数组。

数组赋值后，无论修改哪个变量另一个对象的数据值也会相当发生改变。

- 总结：推荐使用字面量方式声明数组，而不是 Array 构造函数
- 实例方法 forEach 用于遍历数组，替代 for 循环
- 实例方法 filter 过滤数组单元值，生成新数组
- 实例方法 map 迭代原数组，生成新数组
- 实例方法 join 数组单元元素拼接成了字符串
- 实例方法 concat 合并两个数组，生成新数组
- 实例方法 sort 对原数组单元值排序
- 实例方法 splice 删除或替换原数组单元
- 实例方法 indexOf 检索数组单元值
- 实例方法 reverse 反转数组
- 静态方法 from 伪数组转成数组

RegExp 内置的构造函数，用于创建正则表达式。
Regular Expression

总结：

- 推荐使用字面量定义正则表达式，而不是 RegExp 构造函数

创建正则两种方法

- 字面量
- 构造函数

在 JavaScript 中的字符串、数值、布尔具有对象的使用特征，如具有属性和方法，

- 之所以具有对象特征的原因是字符串、数值、布尔类型数据是 JavaScript 底层使用 Object 构造函数“包装”来的，被称为包装类型。
- **String**
- **Number**
- **Boolean**

String 是内置的构造函数，用于创建字符串。

- 总结：推荐使用字面量方式声明字符串，而不是 String 构造函数
- 实例属性 length 用来获取字符串的度长
- 实例方法 split 用来将字符串拆分成数组
- 实例方法 toUpperCase 用于将字母转换成大写
- 实例方法 toLowerCase 用于将字母转换成小写
- 实例方法 slice 用于字符串截取
- 实例方法 indexOf 检测是否包含某字符
- 实例方法 startsWith 检测是否以某字符开头
- 实例方法 endsWith 检测是否以某字符结尾
- 实例方法 replace 用于替换字符串，支持正则匹配
- 注：String 也可以当做普通函数使用，这时它的作用是强制转换成字符串数据类型。

Number 是内置的构造函数，用于创建数值。

总结：

- 推荐使用字面量方式声明数值，而不是 Number 构造函数
- 实例方法 toFixed 用于设置保留小数位的长度
- 注：Number 也可以当做普通函数使用，这时它的作用是强制转换成数值数据类型。

Boolean 是内置的构造函数，用于创建布尔值。

总结：

- 推荐使用字面量方式声明布尔值，而不是 Boolean 构造函数
- 注：Boolean 也可以当做普通函数使用，这时它的作用是强制转换成布尔类型数据，由其它数据类型转换成布尔类型的数据被称为真值（truly）或假值（falsly）。

至此对 JavaScript 有了更深的理解，即 JavaScript 中一切皆为对象，还有以前学习的 window、Math 对象，最后补充一点无论是引用类型或是包装包类型都包含两个公共的方法 toString 和 valueOf

总计：

- valueOf 方法获取原始值，数据内部运算的基础，很少主动调用该方法
- toString 方法以字符串形式表示对象



传智教育旗下高端IT教育品牌