**【2.02】：面向对象编程介绍**

1. **两大编程思想**
2. 面向过程
3. 面向对象
4. **什么是面向过程编程？**

面向过程(pop)就是分析出解决问题所需要的步骤，然后用函数把这些步骤一步一步实现，使用的时候再一个一个的依次调用就可以了。(面向过程，就是按照我们分析好了的步骤，按照步骤解决问题)

1. **什么是面向对象编程？**

面向对象是把事务分解成为一个个对象，然后由对象之间分工与合作。(面向对象是以对象功能来划分问题，而不是步骤)，在面向对象编程开发思想只能怪，每一个对象都是功能中心，具有明确分工

1. 优点

面向对象编程具有灵活、代码可复用、容易维护和开发的优点，更适合多人合作的大型微软项目。

1. **面向对象的特性：**
2. 封装性：代码封装
3. 继承性：两个对象一个父级一个子级，子级可以继承父级的一些属性和方法
4. 多态性：同一个对象在不同时刻，呈现不同的状态
5. **面向过程和面向对象的区别：**
6. **面向过程(程序简单用)**
7. 优点：

性能比面向对象高，适合跟硬件联系很紧密的东西。如：单片机就采用的面向过程编程。

1. 缺点：

没有面向对象易维护、易复用、易扩展。

1. **面向对象(程序复杂多人用)**
2. 优点：

易维护、易复用、易扩展，由于面向对象有封装、继承、多态性的特性，可以设计出低藕合的系统，使系统更加灵活、更加易于维护

1. 缺点：

性能比面向过程低

**【3.03】ES6中的类和对象**

1. **面向对象**

面向对象更贴近我们的实际生活，可以使用面向对象描述现实事件事物，但是事物分为具体的事物和抽象的事物。

1. **面向对象的思维特点：**
2. 抽取(抽象)对象共用的属性和行为组织(封装)成一个类(模板)
3. 对类进行实例化，获取类的对象
4. **什么叫对象？**

现实生活中：万物皆对象，对象是一个具体的事物，看得见摸得着的实物。例如，一本书、一辆汽车、一个人可以是”对象”，一个数据库、一张网页、一个与远程服务器的连接也可以是”对象”。

在JavaScript中，对象是一组无序的相关属性和方法的集合，所有的事物都是对象，例如字符串、数值、数组、函数等

1. **对象的属性：**

事物的特征，在对象中用属性来表示(常用名词)

1. **对象的方法：**

事物的行为，在对象中用方法来表示(常用动词)

1. **什么是类(class)?**

在ES6中新增加了类的概念，可以使用class关键字声明一个类，之后以这个类来实例化对象。

类抽象了对象的公共部分，它泛指某一大类(class)。

1. **类和对象：**
2. 类抽象了对象的公共不封，它泛指某一大类(class).
3. 对象特指某一个，通过类实例化一个具体的对象。
4. **ES6中的类和对象中的三个注意点：**
5. 在ES6中类没有变量提升，所有必须先定义类，才能同规格类实例化对象
6. 类里面的共有的属性和方法一定要加this使用
7. 类里面的this指向问题
8. Constructor里面的this指向实例对象，方法里面的this指向这个方法的调用者

**【4.04】创建类和生成实例**

1. **创建类**

**语法：**

Class 类名{

Constructor([参数1],[参数2]…){

This.参数1=参数1

This.参数2=参数2

}

}

创建实例：

Var 变量名=new 类名([参数1],[参数2]…);

注意：类必须使用new实例化对象

**注意：**

1. 通过class关键字创建类，类名我们还是习惯性定义首字母大写。
2. 类里面有个constructor函数，可以接受传递过来的参数，同时返回实例对象
3. Constructor函数只要new生成实例时，就会自动调用这个函数，如果我们不写这个函数，类也会自动生成这个函数
4. 生成实例new不能省略
5. 最后注意语法规范，创建类 类名后面不要加小括号，生成实例 类名后面必须加上小括号，构造函数不需要加function
6. **constructor构造函数**

constructor()方法是类的构造函数(默认方法)，用于传递参数，返回实例对象，通过new命令生成对象实例时，自动调用该方法。如果没有显示定义，类内部会自动给我们创建一个constructor()

1. **如何在类中添加共有方法？**

将共有方法直接写在类里面就可以了

**注意：**

1. 我们类里面所有函数不需要写function
2. 多个函数方法之间不需要添加逗号分隔

**【6.06】类继承extend和super关键字**

1. **什么是继承？**

现实中的继承：子承父业，比如我们都继承了父亲的姓。

程序中的继承：子类可以继承父类的一些属性和方法。

1. **继承中的super关键字**

Super关键字用于访问和调用对象父类是哪个的函数。可以调用父类的构造函数，也可以调用父类的普通函数

**【注意】：**

子类在构造函数中使用super，必须放到this前面(必须先调用父类的构造方法,在使用子类构造方法)

1. **extend关键字**

继承的意思，表示子类可以继承父类中的方法和属性

**【14.14】tab选项卡**

1. 插入：insertAdjacentHtml(position,text);

Position是相对于元素的位置，并且必须是以下字符串之一：

1. Beforebegin：元素自身的前面。
2. Afterbegin：插入元素内部的第一个子节点之前。
3. Beforeend：插入元素内部的最后一个子节点之后。
4. Afterrend：元素自身的后面。

Text是要被解析为HTML或xml，并插入到DOM树中的字符串。

1. appendChild不支持追加字符串的子元素，insertAdjacentHTML支持追加字符串的元素。

**【24.02】使用构造函数创建对象**

1. **什么是构造函数和原型？**

在典型的OOP的语言只能中(如：Java)，都存在类的概念，类就是对象的模板，对象就是类的实例，但在ES6之前，JS中并没有引入类的概念。在ES6之前，对象不是基于类创建的，而是用一种构建函数的特殊函数来定义对象和他们的特征。

1. **创建对象的三种方法：**
2. 对象字面量
3. New Object()
4. 自定义构造函数
5. **什么是构造函数?**

构造函数是一种特殊的函数，主要用来初始化对象，即为对象成员赋初始值，它总与new一起使用。我们可以把对象中一些公共的属性和方法抽取出来，然后封装到这个函数里面。

1. **在JS中，使用构造函数时要注意以下两点：**
2. 构造函数用于创建某一类对象，其首字母要大写。
3. 构造函数要和new一起使用才有意义。
4. **new在执行时会做四件事情：**
5. 在内存中创建一个新的空对象。
6. 让this指向这个新的对象
7. 执行构造函数里面的代码，给这个新对象添加属性和方法
8. 返回这个新的对象(所以构造函数里面不需要return)
9. **构造函数**

JavaScript的构造函数中可以添加一些成员，可以在构造函数本身上添加，也可以在构造函数内部的this上添加。通过这两种方式添加的成员，就分别称为静态成员和实例成员。

1. **什么是静态成员？**

在构造函数本上添加的成员称为静态成员，只能由构造函数本来访问。

1. **什么是实例成员？**

在构造函数内部创建的对象成员称为实例成员，只能由实例化的对象来访问。

1. **构造函数的问题**

构造函数方法很好用，但是存在浪费内存的问题

**【26.04】构造函数原型对象prototype**

1. **构造函数通过原型分配的函数是所有对象所共享的。**
2. **什么是原型对象prototype对象？**

JavaScript规定，每一个构造函数都有一个prototype属性，指向另一个对象。注意这个prototype就是一个对象，这个对象的所有属性和方法，都会被构造函数所拥有。

【注意：】我们可以把那些不变的方法，直接定义在prototype对象上，这样所有对象的实例就可以共享这些方法。

1. **原型的作用？**

共享方法

1. **对象原型\_\_proto\_\_：**

对象都会有一个属性\_\_proto\_\_指向构造函数的prototype原型对象，之所以我们对象可以使用构造函数prototype原型对象的属性和方法，就是因为对象有\_\_proto\_\_原型的存在。

1. \_\_proto\_\_对象原型和原型对象prototype是等价的。
2. \_\_proto\_\_对象原型的意义就在于为对象的查找机制提供一个方向，或者说一条路线，但是他是一个非标准属性，因此实际开发中，不可以使用这个属性，它只是内部指向原型对象prototype

**【28.06】原型constructor构造函数**

1. **什么是constructor构造函数**

对象原型(\_\_proto\_\_)和构造函数(prototype)原型对象里面都有一个属性constructor属性，constructor我们称为构造函数，因为它指回构造函数本身。

1. constructor构造函数的作用：

constructor主要用于记录该对象引用于哪个构造函数，它可以让原型对象重新指向原来的构造函数。