

1. 面向对象编程介绍

1.1. 两大编程思想

- 面向过程
- 面向对象

1.2. 面向过程编程POP (Process-oriented programming)

面向过程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（大象装冰箱）

1.3. 面向对象编程OOP (Object-oriented programming)

面向对象是把事物分解成一个个对象，然后由对象之间分工与合作

面向对象是以对象功能划分问题，而不是步骤

在面向对象程序开发思想中，每一个对象都是功能中心，具有明确分工

面向对象具有灵活、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软件项目

面向对象的特性：**封装性 继承性 多态性**

1.4. 面向对象和面向过程的对比

面向过程

优点：性能比面向对象高，适合跟硬件联系和紧密的东西，例如单片机就采用的面向对象编程

缺点：没有面向对象易维护、易复用、易扩展

面向对象

优点：易维护、易复用、易扩展，由于面向对象有封装、继承、多态的特性，可以设计出低耦合的系

统，使系统更加灵活、更加易于维护

缺点：性能比面向对象低

2. ES6中的类和对象

面向对象：面向对象更贴近我们的生活，可以使用面向对象描述现实世界事物，但是事物分为具体的事物和抽象的事物

面向对象的思维特点：

抽取（抽象）对象共有的属性和行为组织（封装）成一个类（模板）

对类实例化，获取类的对象

面向对象编程我们考虑的是有哪些对象，按照面向对象的思维特点，不断地创建对象，使用对象，指挥对象做事情

2.1. 对象

现实世界中，万物皆对象，对象是一个具体的事物，看得见摸得着的实物。

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

对象是由属性和方法组成的

- 属性：事物的特征，在对象中使用属性来表示（常用名词）
- 方法：事物的行为，在对象中用方法来表示（常用动词）

2.2. 类 class

在ES6中新增了类的概念，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）

对象特指某一个，通过实例化一个具体的对象

2.3. 创建类

语法：

```
class name{  
    // class body  
}
```

创建实例：

```
var xx=new name();
```

注意：类必须使用new实例化对象

2.4. 类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，类内部会自动创建一个constructor()

// 注意：

// 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

// 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

// 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

// 生成实例new不能省略

// 最后注意语法规则，创建类 类名后面不要加小括号，生成实例 类名后面加小括号构造函数不需要加function

2.5. 类添加方法

语法：

```
class Person{
  constructor(name,age){
    this.name=name;
    this.age=age;
  }
  say(){
    console.log(this.name+'你好');
  }
}
```

// 类里面所有的函数不需要写function

// 多个函数方法之间不需要添加逗号分隔

3. 类的继承

3.1. 继承

子类可以继承父类的一些属性和方法

语法:

```
class Father{ //父类

}

class Son extends Father{ //子类继承父类

}
```

3.2. super 关键字

super关键字用于访问和调用对象父类上的函数，可以调用父类的构造函数，也可以调用父类的普通函数

```
<script>
    class Father {
        constructor(x, y) {
            this.x = x;
            this.y = y;
        }
        sum() {
            console.log(this.x + this.y);
        }
    }

    // 子类继承父类加法方法，同时 扩展减法方法
    class Son extends Father{
        constructor(x, y) {
            // 利用super调用父类的构造函数
            // super 必须在子类的this之前调用
            super(x,y);

            this.x = x;
            this.y = y;

        }
        subtract() {
            console.log(this.x - this.y);
        }
    }

    var son=new Son(5,8);
    son.subtract();
    son.sum();

</script>
```

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，再使用子类构造方法）

三个注意点：

- 在ES6中类没有变量提升，所以必须先定义类，才能通过类实例化对象
- 类里面的共有的属性和方法一定要加this使用
类里面this 的指向问题
- 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

面向对象版tab栏切换

功能需求：

- 点击tab栏，可以切换效果
- 点击加号，可以添加tab项和内容项
- 点击x号，可以删除当前的tab项和内容项
- 双击tab项文字或者内容项文字，可以修改里面的文字内容

抽取对象：Tab对象

- 该对象具有切换功能
- 该对象具有添加功能
- 该对象具有删除功能
- 该对象具有修改功能

添加功能：

点击+可以实现添加新的选项卡和内容

第一步：创建新的选项卡li和新的内容section

第二步：把创建的两个元素追加到对应的父元素当中

以前的做法：动态创建元素createElement，但是元素里面内容较多，需要innerHTML赋值，再appendChild追加到父元素里面

现在的高级做法:利用insertAdjacentHTML()可以直接把字符串格式元素添加到父元素中

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

删除功能：

点击x号，可以删除当前li选项卡和当前的section

x号没有索引号，但是它的父亲li有索引号，这个索引号正是我们想要的索引号

编辑功能

双击选项卡li或者section里面的文字，可以实现修改功能

双击事件是：ondblclick

如果双击文字，会默认选定文字，此时需要双击禁止选中文字

```
window.getSelection?window.getSelection().removeAllRanges():document.selection.empty();
```

核心思路：双击文字的时候，在里面生成一个文本框，当失去焦点或者按下回车然后把文本框输入的值给原先的元素即可