### JS-1

1. 中科软
   1. Vue2.0脚手架搭建
      1. Npm uninstall -g vue-cli
      2. Npm install -g vue-cli
      3. Vue --version //查看版本号
      4. Vue init webpack 项目名（小写）
   2. Vue3.0脚手架搭建
      1. Npm uninstall -g @vue/cli
      2. Npm install -g @vue/cli
      3. Vue --version //查看版本号
      4. Vue create 项目名（小写）
   3. Vue.config.js配置
      1. Module.exports = {
         1. publicPath: ‘./’
         2. productionSourceMap: false,
         3. devServer: {
            1. Proxy: {

‘/api’: {

Target: ‘http://localhost:3000’,

changeOrigin: true,

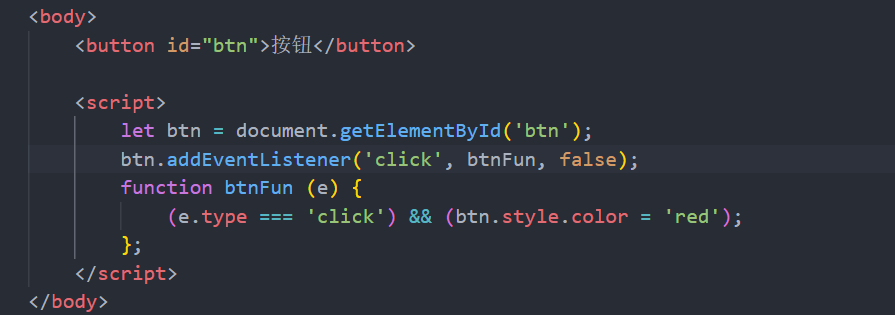
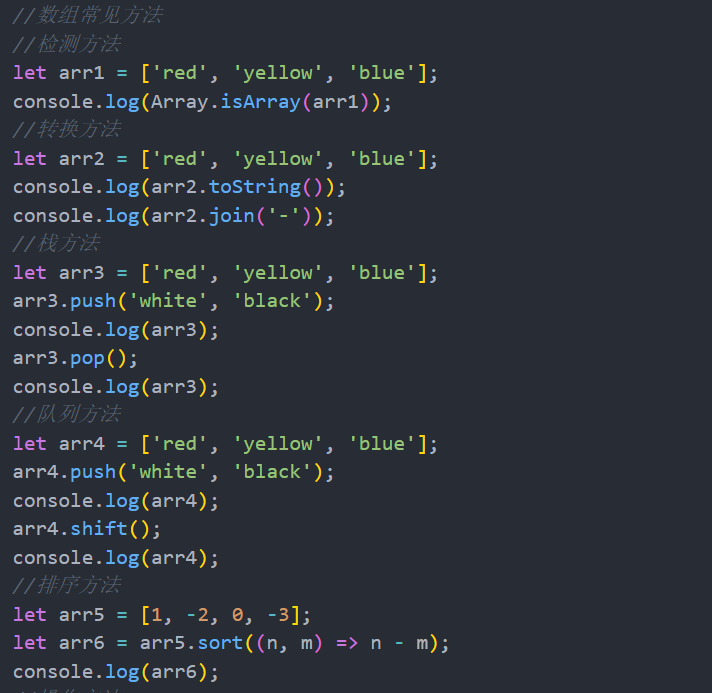
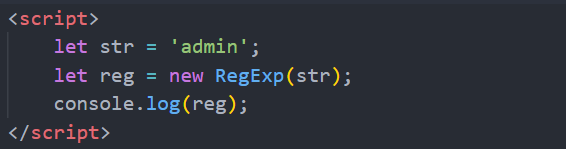
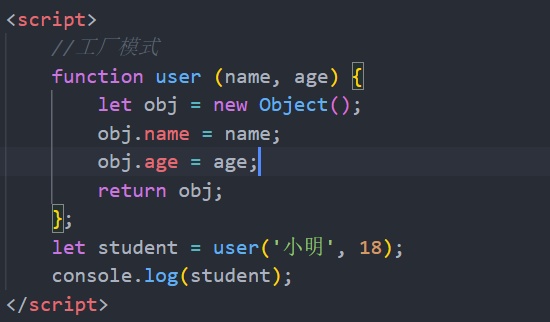
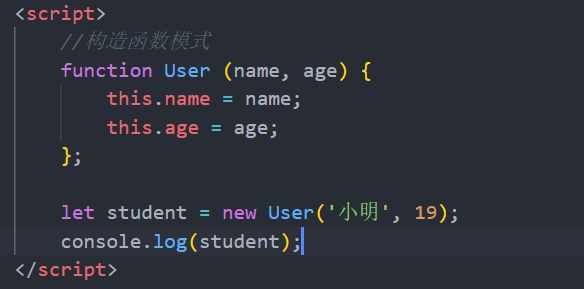
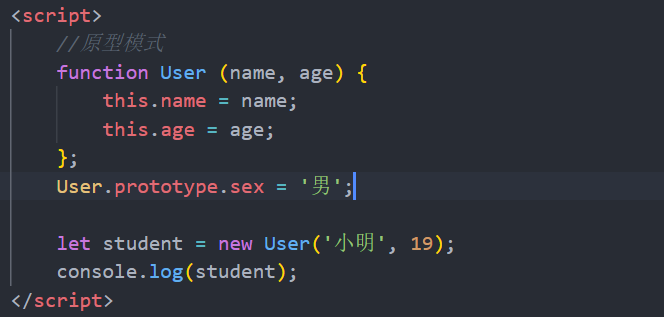
pathRewrite: {

‘^/api’: ‘’

}

}

* + - * 1. }
      1. }
    1. }

1. 事件
   1. HTML与JS之间的交互是通过事件完成的
   2. 事件增强了页面的行为（JS）与页面的外观（HTML+CSS）之间的耦合度
      1. 事件源
         1. 事件源包含：事件冒泡，事件捕获和DOM事件流
         2. 事件冒泡：给某个div元素添加一个click点击事件，执行顺序：div==body==html==document
         3. 事件捕获：给某个div元素添加一个click点击事件，执行顺序：document==html==body==div
         4. DOM事件流：给某个div元素添加一个click点击事件，执行顺序：document==html==body==div==body==html==document
      2. 事件处理程序
         1. 响应事件的函数，称为事件处理程序
         2. 事件处理程序是以’on’开头的，click点击事件的事件处理程序是onclick，mouseover鼠标移动事件的事件处理程序是onmouseover，load加载事件的事件处理程序是onload
         3. 添加事件监听：addEventListener(‘事件名’, 函数, false); false代表事件冒泡
         4. 移出事件监听：removeEventListener(‘事件名’)
         5. 
      3. 事件对象
         1. Event.type event.target
         2. 
2. 基本概念
   1. 函数是对一段代码的封装，可以重复使用
   2. 函数使用return返回值
   3. Return语句一旦执行，函数终止
   4. 没有return，或者return；函数的返回值是undefined
3. 引用数据类型
   1. Object引用数据类型
      1. 创建Object的两种方式
      2. Let obj = new Object();
      3. Obj.name = ‘小明’;
      4. Obj.showName = function () {
         1. Return this.name;
      5. }
      6. Let obj = {
         1. Name: ‘小红’,
         2. showName: function () {
            1. Return this.name;
         3. }
      7. }
   2. Array引用数据类型
      1. 创建Array的两种方式
      2. Let arr = new Array();
      3. Let arr = [];
      4. arr具有index索引和length属性
      5. Arr常见方法
      6. 检测方法
      7. 转换方法
      8. 栈方法 后进先出
      9. 队列方法 先进先出
      10. 排序方法
      11. 操作方法
      12. 位置方法
      13. 迭代方法
      14. 归并方法
      15. 
      16. 
   3. RegExp引用数据类型
      1. //g，全局匹配
      2. //m，多行匹配
      3. //i，忽略大小写
      4. 元字符需要转义，转义符是\ 常见的元字符有：()[]{} \*+? \|.^$
      5. 
4. 面型对象编程
   1. 在js中，对象是一个实例，对象的值可以是数值，也可以是方法
      1. 属性类型
         1. Writable：表示值是否可以修改
         2. Configurable：表示是否可以删除属性
         3. Enumerable：表示是否可以for-in循环
         4. Value：值
         5. 设置单个属性：Object.defineProperty(对象, ‘name’, {
            1. Writable: true,
            2. Configurable: true,
            3. Enumerable: true,
            4. Value: ‘小明’
         6. })
         7. 
         8. 获取单个属性：Object.getOwnPropertyDescriptor(对象, ‘name’)
         9. 
      2. 创建对象
         1. 工厂模式
            1. 
         2. 构造函数模式
            1. 
         3. 原型模式
            1. 
            2. 获取原型属性：Object.getPrototypeOf(对象).属性
            3. 
            4. Instanceof 判断是否是对象
            5. 
            6. in与hasOwnProperty()
            7. In判断原型中是否包含此属性
            8. hasOwnProperty()判断实例对象是否包含此属性
5. JOSN对象与JS对象
   1. JSON对象转换成JS对象
      1. JSON.parse()
   2. JS对象转换成JSON对象
      1. JSON.stringify()
   3. 日期对象转换成JSON对象
      1. New Date().toJSON()