1. jQuery：一个优秀的JS函数库。Write Less，Do More。
   1. 功能：HTML元素选取（选择器），HTML元素操作，CSS操作，HTML事件处理，JS动画效果，链式调用，读写合一，浏览器兼容，易扩展插
   2. - 件，ajax封装
   3. 版本：1.x版兼容老IE，文件大 2.x部分IE8及以下不支持，文件小效率高 3.x兼容IE9+，提供了一些新的API
      1. 开发版：jquery.js 开发时使用 压缩版：jquery.min.js 对代码进行过压缩，仅在上线时使用
   4. 引入：方法一：本地引入。方法二：CDN远程库（项目上线时使用，减轻服务器负担，需要网络）
      1. 引入js文件：const $ = require（“jquery”） // es5 或 import $ from ‘jquery’； // es6
2. 语法：
   1. [window.] $ 即[window.] jQuery
   2. jQuery（）函数：返回jQuery对象，函数内可存放四种参数形式：
      1. $（选择器字符串，[context]） 通过选择器获取对应的DOM对象，然后将这些DOM封装到一个jQuery对象中并返回
      2. $（DOM对象，即Node的实例） 将该DOM对象封装成jQuery对象并返回
      3. $（HTML标签字符串） 根据传入的文本创建好HTML元素并封装成jQuery对象返回
      4. $（回调函数） 当DOM加载完毕并执行完脚本之后（与DOMContentLoaded同时），执行此回调函数
         1. 是$（document）.ready（）的简写，onload和ready的区别：
            1. window.onload：所有页面包括图片加载完才会回调（慢），只能有一个监听回调
            2. $（document）.ready（）：页面加载完就能回调（快），可以绑定多个监听回调
   3. 选择器规则：
      1. 基本选择器（同CSS规则）：id选择器，类选择器，\*选择器，并集选择器，交集选择器
      2. 层次选择器（同CSS规则）：后代选择器，子选择器，相邻兄弟选择器，后续兄弟选择器
      3. 过滤选择器：对原有选择器选中的元素进行过滤
         1. :first / :last 选择第一个/最后一个元素
         2. :not（选择器1） 选择选择器1不选中的元素，选择器不需要加引号
         3. :even / :odd 选择下标为偶数/奇数的元素（从0开始）
         4. :eq（下标） / :gt（下标） / :lt（下标） 选择下标等于/大于/小于指定值的元素，过滤完一次后下标会变，复合使用先lt再gt
         5. :contains（文本） 选择内容包含给定文本的元素
         6. :hidden / :visible 选择隐藏/可见的元素
         7. [属性] / [属性=值] / [属性!^$\*=值] 选择指定属性存在/等于值/不等于值/以值开头/以值结尾/包含值的元素（CSS3中有）
         8. :input 选择所有input，textarea，select，button元素
         9. :text / :password / :radio / :checkbox / :submit / :image / :reset / :button / :file 选择表单类型为指定类型的表单
         10. :enabled / :disabled / :checked / :selected 选择表单对象属性为指定属性的元素
   4. jQuery对象：一般在变量前加$用来表示其为jQuery对象。为伪数组
      1. jQuery对象的说明：
         1. jQuery对象是jQuery函数的实例，他可以调用jQuery原型中定义的一些方法
         2. jQuery对象是一个类数组对象，内部存放的元素是DOM对象
         3. jQuery对象的获取通常是使用选择器来获取 例: $(选择器字符串)
         4. jQuery对象的操作实际上就是对jQuery数组中DOM对象的批量操作
      2. jQuery类的静态方法：
         1. jQuery.noConflict（[true]） 将$变量释放，实现多库共存
         2. $.each(数组或对象，回调函数) 隐式遍历，函数可有两个参数，分别为属性名与值（对象）/ 下标与值（数组）
         3. $.trim(字符串) 去除两端的空格
         4. $.type(对象) 判断数据的类型（类似typeof）
         5. $.isArray(对象) / $.isFunction(对象) 判断是否为数组/函数
         6. $.parseJSON(json字符串) 解析json字符串转换为js对象/数组
         7. $.extend（对象1，对象2） 将对象2合并到对象1中，覆盖同名属性
      3. jQuery对象的属性和方法：$表示jQuery类，$a表示jQuery对象
         1. 笔记简写备注：
            1. properties：{属性名：值字符串，。。。} 可以任意添加多条属性
            2. fn：function（[index，attr/class]） 根据函数的返回值修改对应属性，参数为当前元素索引值和原先属性值
         2. 基本：
            1. $a.size（）/ $a.length 获取对象内DOM对象的数量
            2. $a[index] / $a.get(index) 根据下标返回对应DOM对象
            3. $a.each(fn([index, dom])) 遍历DOM对象，对其执行函数，参数：在兄弟中的下标和dom元素本身（this）
            4. $a.index() 返回该DOM对象在其兄弟元素中的下标
         3. 获取html属性：读写合一（传参数表示写入，不传参数表示读取）
            1. $a.attr（’name’ / properties / ’key’，’value’ / ’key’，fn） 获取或设置所选元素的属性值，详细见上注
            2. $a.prop（’name’ / properties / ’key’，’value’ / ’key’，fn） 功能同上，操作属性值为布尔型的属性必须用prop，防止出错
            3. $a.removeAttr（‘name’）/ $a.removeProp（‘name’） 删除指定属性

attr和prop都可以获取自定义对象的属性值，区别是attr能获取标签元素自定义属性，prop无法获取，返回undefined；两个remove都能移除标签上自带属性。removeAttr不能移除自定义对象的属性，removeProp不能移除标签自定义属性，可以移除自定义对象上的属性

* + - * 1. $a.addClass（’class’ / fn）/ $a.removeClass（同左）/ $a.toggleClass（’class’ / fn[,sw]） 向所选元素添加/删除类/切换类
        2. $a.html（‘val’ / fn）/ $a.text（‘val’ / fn） 获取或修改所选DOM对象的标签体（innerHTML）/内容文本
        3. $a.val（‘val’ / fn / arr） 获取或修改所选DOM对象的value属性

arr：字符串数组，仅对复选框等可选元素有效，将指定的选项选上

* + - 1. CSS样式：获取或设置宽高使用数值型，不需加单位，默认为像素
         1. $a.css（’name’ / properties / ’name’，’value’ / ‘name’, fn） 获取或修改CSS属性（属性名可用驼峰命名），详细见上注
         2. $a.offset（[含有top和left属性的对象/fn]） 获取或修改元素相较页面左上角的位置，返回有top、left属性的对象

fn第一个参数为元素的索引，第二个参数为当前的坐标。仅对可见元素有效

* + - * 1. $a.position（） 获取元素相对父元素的偏移，返回对象同上
        2. $a.scrollTop（[val]）/ $a.scrollLeft（[val]） 获取或设置元素被卷去滚动条顶部/左部的像素数的数值。

获取页面滚动值：ie从html读取，chrome从body读取，兼容写法：

$(document.documentElement或‘html’).scrollTop() + $(document.body或‘body’).scrollTop()

设置：$（‘html，body’）.scrollTop(300)

* + - * 1. $a.height（[val / fn]）/ $a.width（[val / fn]） 获取或设置元素的内容的宽高
        2. $a.innerHeight（）/ $a.innerWidth（） 获取元素的内部尺寸（height+padding）
        3. $a.outerHeight（[boolean]）/ $a.outerWidth（[boolean]） 外部尺寸（height+padding+border），若为true，再加上margin
      1. 文档处理：
         1. $a.append（HTML字符串/fn）/ $a.prepend（同上） 把HTML字符串插入$a中所有元素内部末尾/第一位
         2. $a.appendTo（‘选择器’）/ $a.prependTo（同上） 把$a中的DOM对象插入选择器所选中的元素内部末尾/第一位
         3. $a.after（HTML/fn）/ $a.before（同上） 把内容插入到$a的前面/后面
         4. $a.insertAfter（‘选择器’）/ $a.insertBefore（同上） 把$a插入到选中标签的前面/后面
         5. $a.replaceAll（‘选择器’）/ $a.replaceWith（HTML/fn） 将选中的元素全部替换为$a / 将$a替换为指定内容
         6. $a.empty（）/ $a.remove（） 删除$a的所有子节点（包括文本节点）/ 删除$a，可过滤
      2. 筛选：
         1. $a.eq（i）/$a.first（）/$a.last（） 获取索引值为i的元素（支持负数索引）/第一个/最后一个，返回jQuery对象
         2. $a.hasClass(class) 检查当前的元素是否含有某个特定的类，如果有，则返回true
         3. $a.filter（‘选择器’）/ $a.not（‘选择器’） 返回符合/不符合指定表达式的jQuery对象
         4. $a.has（‘选择器’） 返回含有符合选择器的后代的元素
         5. $a.children（[‘选择器’]）/ $a.find（‘选择器’） 返回子元素/后代元素中符合选择器的元素
         6. $a.parent（[‘选择器’]） 返回父元素，可应用选择器过滤
         7. $a.prev（[‘选择器’]）/ $a.prevAll（[‘选择器’]） 返回当前元素相邻前一个元素/之前所有同辈元素，可应用选择器过滤
         8. $a.siblings（[‘选择器’]） 返回同辈元素，可过滤
      3. 事件：回调函数的this指向绑定事件的dom元素的实例
         1. event事件对象：函数可以指定event事件对象，其属性参考JavaScript
         2. $a.on / one（事件名字符串，函数） 绑定事件事件名可写多个，用空格分隔；one绑定的事件在触发一次后失效

on和click区别：on可以动态绑定，且可绑事件多，建议用on

* + - * 1. $a.off（[事件名字符串]） 解除指定事件监听，若不写参数解除所有事件
        2. $a.delegate（‘选择器’，‘事件’，函数） 为$a中符合选择器的子元素设置事件委托
        3. $a.undelegate（[‘选择器’]，[‘事件’]） 移除事件委托
        4. $a.hover（函数/移入函数，移出函数） 当鼠标移入或移出时触发函数，若只指定一个函数，则移入移出时都触发
        5. $a.click(函数) 绑定点击事件监听
      1. 动画效果：
         1. $a.fadeIn（[‘速度’]，[‘效果’]，[fn]）/ $a.fadeOut（同上） 通过调整不透明度实现淡入/淡出

速度：可用以下字符串slow/normal/fast或用数值型，代表动画时长的毫秒值

效果：字符串swing（慢-快-慢）或linear（线性变化）

fn：回调函数

* + - * 1. $a.fadeToggle（同上）/ $a.slideToggle（同上） 切换是否透明/是否展开
        2. $a.slideUp（同上）/ $a.slideDown（同上） 通过控制元素高度实现收起/展开效果
        3. $a.show（同上）/ $a.hide（同上）/ $a.toggle（同上） 通过同时改变宽高透明度，实现缓慢显示/隐藏/显示隐藏切换
        4. $a.animate（参数对象，[‘速度’]，[‘效果’]，[fn]） 自定义动画

参数对象：一个对象，属性和终值存为键值对，属性名可不用字符串，属性值可用数值型指定终值，也可用‘+=100’之类

若没有参数，表示没有动画

* + - * 1. $a.stop（） 停止$a当前的动画，为防止动画队列堆积，在设置新动画前使用

1. jQuery插件：是基于jQuery编写的扩展库
   1. $.fn.extend（对象） 将对象中的方法拓展给jQuery对象
   2. jquery-validation：表单验证。
      1. 将jquery.js、jquery.validate[.min].js导入项目
      2. input标签新增特性：
         1. required[=“true”] 该表单必须填写
         2. maxlength/minlength/rangelength = “数字” 长度限制
         3. equalTo = “选择器” 与指定元素值相同
      3. 对表单执行语句：$（‘xxx’）.validate（参数） 开启验证
      4. 可对validate函数传入参数来指定提示文本：
         1. validate（｛messages：｛input的name：｛required：‘xxx’，minlength：‘xxx’｝，input2的name：｛。。。｝｝｝）
   3. jquery UI：http：//jqueryui.com/
      1. 将jquery-ui.css、jquery-ui.js、jquery.js导入项目
      2. 指定样式匹配：给指定div添加id，在js代码中$（‘#xxx’）.accordion（）应用，方法名按所需效果改
      3. 文本框自动搜索匹配：
         1. var dataSource = [。。。]
         2. $（‘xxx’）.autocomplete（｛source：dataSource｝）
      4. 选项卡效果：<ul><li><a href=“#tabs-1”>xxx</a></li></ul><div id=“tab-1”>xxx</div>
         1. $（‘xxx’）.tabs（）
2. AJAX：
   1. $.get（‘url’，data对象，回调函数[，‘响应体类型（如json）’]） 发送get请求，post同理
      1. data对象：｛a：100，b：200｝
      2. 回调函数：可传入变量data，表示响应体
      3. 响应体类型，若为json，data会自动转换为对象
   2. $.ajax（对象） 通用的ajax方法
      1. 对象的属性：
      2. url：xx，data：｛xx｝，type：‘GET’，success：fn（data）｛成功的回调｝，headers：｛xx：xx｝，
      3. dataType：‘响应体类型’，timeout：2000，error：fn（xhr，textStatus，errorThrown）｛失败的回调｝

JSONP：

$.getJSON（‘url’，fn（data）｛｝） url的结尾要加上 ?callback=? ，jquery会自动填写callback的属性，其值为服务端应调用的函数名