Vuex的原理：状态管理工具，可以存放公共数据信息，不同组件中可以通用。

语法：

1.引入vuex库

2.创建仓库存放数据

```
const store = new Vuex.Store({ // 单词大小写注意留心
 // 仓库数据（状态 state 其实就是单个组件data）
 state: { 键1:值1,...,键n:值n },
 // 仓库数据过滤
 getters: { 方法名(state) {},.... },
 // 更新仓库数据（推荐方法名 get/set仓库中的键
 // 更新state语法：state.键 = 值
 mutations: { 方法名(state [,参数]) {},.... },
 // 异步请求->调用mutations更新state数据
  // 触发mutations语法：context.commit(mutations方法名 [,参数])
  actions: { 方法名(context) {},.... }
})

state&getters都是在computed里面调用 -> 接着视图使用计算属性语法显示
mutations&actions都是在methods里面调用 -> 接着视图触发事件调用方法即可
```

举例说明：

```
// 创建store对象通过 Vuex.Store
const store = new Vuex.Store({
 // 存放状态
 state: {
 age: 19,
 name: 'Jack'
 },
 // 过滤状态//创建一个属性username: 'helloJack'
 getters: {
  username(state) {
  return `hello ${state.name}`
  }
 },
 // 更新状态(commit触发)
 mutations: {
 setAge(state) {
```

```
 state.age += 1
 }
 },
 // 调用mutation（dispatch触发）
 actions: {
 setAge(context) {
 context.commit('setAge')
 }
 }
})
const vm = new Vue({
 // 激活
 store,
 el: '#app',
 data: {
 sex: "男"
 },
 // 通过computed来获取state和getters
 computed: {
 age() {
 return this.$store.state.age
 },
 username() {
 return this.$store.getters.username
 }
 },
 // 通过methods来存数据 -> 就是触发actions和mutations
 methods: {
 // this.$store.dispatch(actions方法名, 参数)
 // this.$store.commit(mutations方法名, 参数)
 testActionsFn() {
 this.$store.dispatch('setAge') // 触发actions
 },
 testMutationsFn() {
 this.$store.commit('setAge') // 触发mutations
 }
 }
})
</script>
```

辅助函数:

```
computed: {
        age() {
            return this.$store.state.age
        },
        name() {
            return this.$store.state.name
        },
==>数组方法（推荐 ...Vuex.mapState(['age', 'name']),
        // 发现：上面辅助函数导致computed中方法名就是获取的数据名
         // 思考：1 是否会出现重名的情况、 2 如何改变方法名
         // 练习：所有名字加2
        ==>对象方法（推荐 ...Vuex.mapState({
             age2: state => state.age,
             name2: state => state.name,
         }),
    },
```

```
methods: {
===》对象方法
//mutations中的setAge
        ...Vuex.mapMutations({
            fn1: 'setAge'
        }),
//Actions先触发mutations再调用setAge
        ...Vuex.mapActions({
            fn2: 'setAge'
         }),
    },
```

vuex语法模板:

可以传参

vuex模块化：

```
3. 使用数据
    获取
        this.$store.state.键
        this.$store.getters.键
        this.$store.state.模块名.键
        this.$store.getters.模块名.键
    更新
        this.$store.commit('mutations方法名', 参数)
        this.$store.dispatch('actions方法名', 参数)
        this.$store.commit('模块名/mutations方法名', 参数)
        this.$store.dispatch('模块名/actions方法名', 参数)

    辅助函数

    ...Vuex.mapState/mapGetters/mapMutations/mapActions([])
    ...Vuex.mapState({
        键: state => state.键
        键: state => state.模块名.键
    })
    ...Vuex.mapGetters/mapMutations/mapActions({
        键:'名字'
        键:'模块名/名字'
    })
```

举例：

```
...mapMutations({
          addFn:'amodul/addli'
      })
```