HTML新特性 – Web Storage

1. Web Storage
   1. 基础

Web storage提供了一种比cookie更加直观、更加便捷的数据存储方式，其结构为名/值对。

web storage提供两种存储机制：

* sessionstorage:为指定的每一个源维护一个独立的区域，该存储区域的数据只在页面会话期间可用(只要浏览者关闭浏览器数据立即消失)
* localstorage: 为指定的每一个源维护一个独立的区域，该存储区域的数据一直有效，除非浏览者手动清除。
  1. 方法
* setItem()方法

setItem()方法用于实现将键名添加到存储中，如果键名已经存在，则更新键值，语法结构是：

storage.setItem(key,value)

* getItem()方法

getItem()方法用于返回指定键名的值，其语法结构是：

variable = storage.getItem(key)

说明：如果指定的key不存在，则返回null

* removeItem()方法

removeItem()方法用于删除指定的键名，如果键名不存在，则不进行任何执行，语法结构是：

storage.removeItem(key)

* clear()方法

clear()方法用于删除所有的键名，其语法结构是：

storage.clear()

* 1. 属性

length属性

length属性用于返回storage中存储的数据项的数量，其语法结构是：

variable = storage.length

1. 再谈Vuex

Vuex是一个专为Vue应用程序而开发的状态管理模式，它采用集中式的存储来管理应用中的组件状态。

每一个Vuex应用的核心就是store，store是一个容器，它包含了应用中、在各个组件共享的状态。

核心概念

* state

state定义了应用中状态的数据结构，可以为任意数据类型，如

  state: {

      username:'Tom',

      age:22,

      sex:true, //true表示男,false表示女

      salary:7896.33

   }

在组件中访问state数据:

{{this.$store.state.变量名称}}

{{$store.state.变量名称}}

* Getters

Vuex允许在store中定义getters(可以将其认为是store的计算属性)，它的返回值可依赖于store中的state(结果会被缓存起来)，只有当state发生变化时，才会重新计算。

定义getters，语法结构如下：

getters:{

propertyName(state){

...

}

}

state将作为getters中的唯一一个参数，该参数在调用时无需传递，Vuex将自动把state注入到该方法内。

* Mutations

Mutations是状态改变的操作方法，也是Vuex修改state的唯一方法。其语法结构是：

mutations:{

方法名称(state[,payload]){

...

}

}

state将作为方法的第一个参数出现, 该参数在调用时无需传递，Vuex将自动把state注入到该方法内

payload，称为载荷，是指除state之外的参数；如果payload多个参数的话，建议使用对象

组件中调用Mutations：

this.$store.commit("方法名称"[,payload])

Mutations只能进行同步操作

* Actions

Actions类似于Mutations，但是存在以下区别：

1. Actions可以包含任何异步操作，而Mutations只能进行同步操作
2. Actions提交的是mutation，而不是直接修改state

其语法结构是：

actions:{

方法名称(context){

...

}

}

组件中调用Actions方法

this.$store.dispatch("方法名称")