T-Query功能

- 我们可以通过自定义的T-Query完成对一些基本操作的封装，如果以后想使用某些功能的话只需要调用T-Query的方法即可。
- T-Query功能大体如下：
  - 选择器的封装
  - 常用操作方法的封装
  - 简易动画封装
  - 事件绑定封装

知识讲解

什么是选择器

- 选择器是支持开发者所熟悉的CSS语法，快速且轻松的对页面进行设置
- CSS中常用的选择器
  - ID选择器 ： #id{ }
  - 类选择器 ： .class{ }
  - 标签选择器 ： element{ }
  - … …

知识讲解

TQObject对象

- TQObject对象是一个自定义的JS对象，其主要目的是为了保存通过各个选择器获取的一个或一组DOM对象。
- TQObject对象除了能够保存DOM对象外，还要具备一组方法，用来方便的操作这些对象。

知识讲解

```
function TQObject(){
    this.data=[];
}

TQObject.prototype={ }
```

$对象

- $对象实际上是模仿jQuery中的$,用于根据指定的选择器获取页面上的元素，并将获取的元素封装到TQObject中,并进行返回。

知识讲解

```
var $ = function(selector){
    tqObject = new TQObject();
    //根据不同的selector获取不同的DOM元素
    return tqObject;
}
```

#id

- ID选择器：通过页面元素的id属性值去定位页面上的一个元素。
- 封装id选择器的目的是简化document.getElementById
- 语法 ： $(“#id”)

知识讲解

```
if(selector.substring(0,1) === "#"){
    //判断是否为$("#id")
    var elem =
    document.getElementById(selector.substring(1));
    this.tqObject = new TQObject();
    this.tqObject.data.push(elem);
```

.class

知识讲解

- 类选择器：通过页面元素的class属性值获取页面上一组元素。
- 其目的是简化document.getElementsByClassName
- 由于document.getElementsByClassName的兼容性不是很好，所以对不支持该方法的浏览器需要通过正则表达式进行匹配
- 语法：$(“.class”)

.class(续1)

知识讲解

```
if(selector.substring(0,1) === "."){
    if(document.getElementsByClassName){
        var elems = document.getElementsByClassName(selector.substring(1));
        var len = elems.length;
        for(var i=0;i<len;i++){
            this.tqObject.data.push(elems[i]);
        }
} else {
        var elems = document.getElementsByTagName("*");
        var reg = new RegExp('(^|\\s)'+className+'($|\\s)');
        var arr=[];
        for(var i=0 ; i < elems.length ; i ++){
            if(reg.test(elems[i].className)){
                this.tqObject.data.push(elems[i]);
            }       }       }       }
```

element

- 标签选择器：通过页面上的标签名称获取一组元素
- 其目的是简化document.getElementsByTagName
- 语法 ： $(“element”)

知识讲解

```
var elems = document.getElementsByTagName(selector);
var len=elems.length;
for(var i=0;i<len;i++){
        this.tqObject.data.push(elems[i]);
}
```

html ( ) / html (text)

- html ( ) 用于向指定的T-Query元素赋html值
  html (text) 用于获取指定的T-Query元素的html值
- html ( ) 方法是对 innerHTML属性进行了封装
- 语法：$(selector).html([text])

知识讲解

```
html:function(text){
if(text){
        for(var i=0;i<this.data.length;i++)
                this.data[i].innerHTML=text;
        return this;
}else {
        if(this.data.length == 0)
                return;
        else
                return this.data[0].innerHTML;     }          }
```

text（ ）/ text（text）

知识讲解

- text（ text ）用于向指定的T-Query元素赋text值
  text（ ） 用于从指定的T-Query元素中获取text值
- text（ ）方法相当于对innerText属性进行了封装
- 注意：Firefox中不支持innerText属性，需要通过textContent属性来取代。
- if(window.navigator.userAgent.toLowerCase().indexOf('firefox') != -1)
- 语法：$(selector).text([text]);

val ( ) / val (value)

- val ( ) 方法用于获取或设置T-Query元素的value属性值，如 <input type= “text” />等一些表单元素。
- 相当于对 dom.value 属性进行封装
- 语法：$(selector).val([value])

知识讲解

attr（name）/ attr（name,value）

知识讲解

- 获取或设置T-Query元素的指定属性堉
- 相当于 dom.setAttribute(name,value) / dom.getAttribute(name)
- 语法：
  - $(selector).attr（“class”）：获取class属性值
  - $(selector).attr（“class”，“red”）；设置元素class属性的值为red

addClass（）

- addClass（）用于向匹配的元素追加class样式
- 是对className属性的进一步封装，但与className不同的是addClass（）方法可以在原有的样式上追加新有样式，两个样式可以并存。
- 语法：$(selector).addClass(className);

知识讲解

removeClass（ ）

- removeClass（className）用于从匹配的元素上移除指定的class样式
- removeClass（ ）用于移除所有的class样式
- 语法：$(selector).removeClass([className]);

知识讲解