# 操作内容

对元素的内容进行操作包括取得该元素的内容，修改该元素的内容。

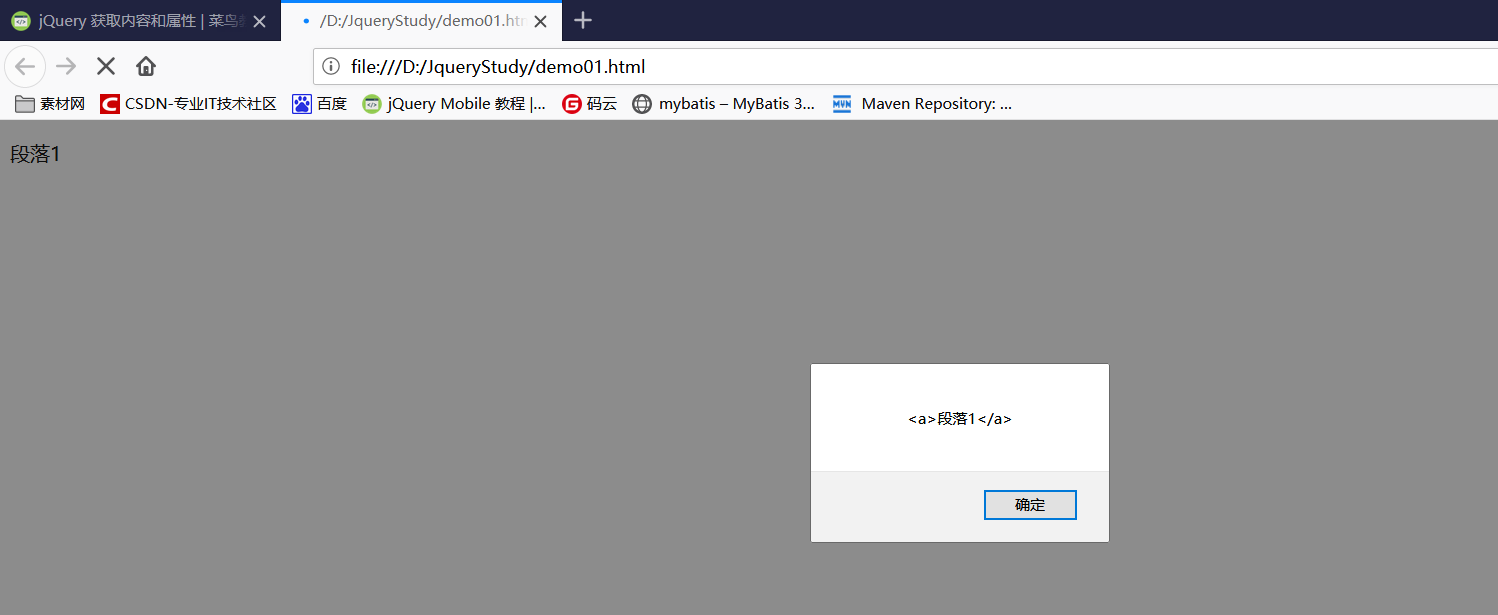
## html( )方法

### 读取内容

1. 在JavaScript中使用元素的innerHTML属性来读取它的内容，读取的内容中包括标签元素。例如：



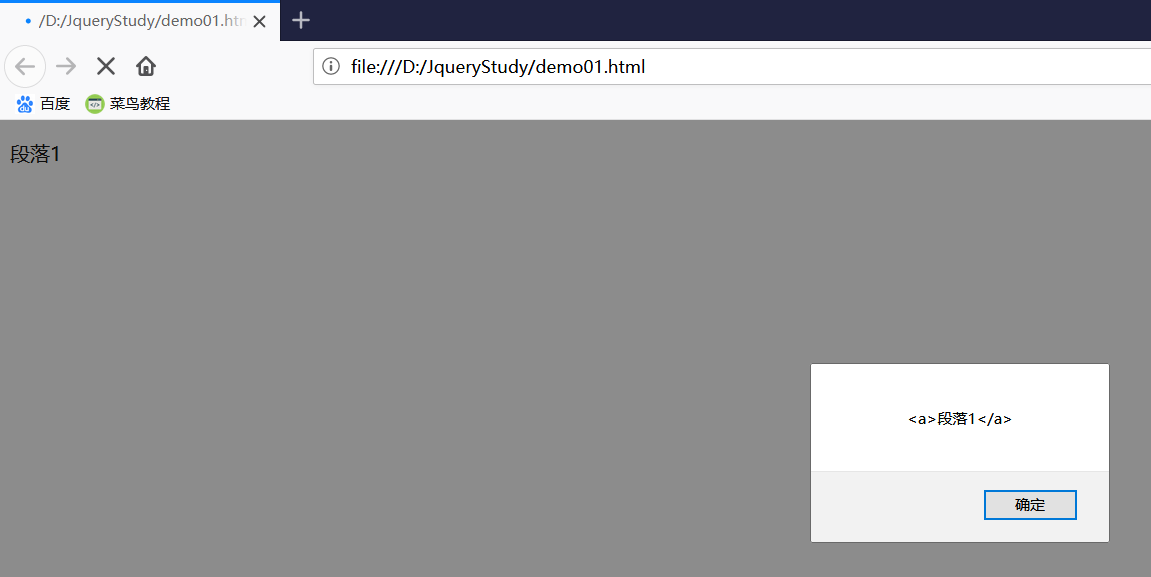
运行结果为：



1. 通过JQuery简化后，如下：

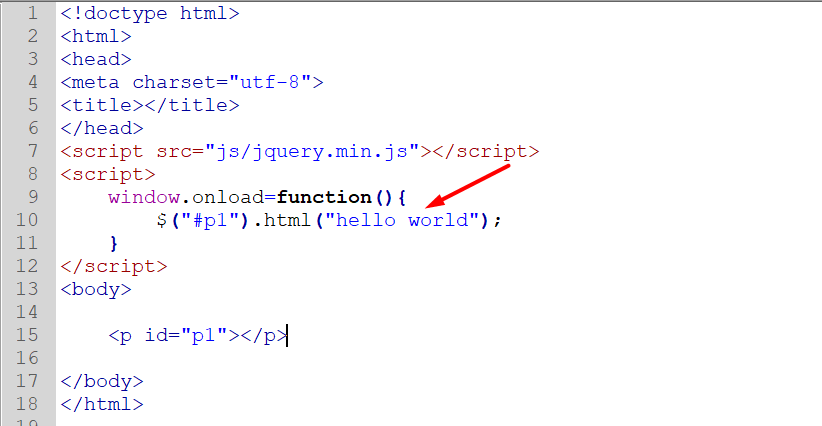


运行结果为：

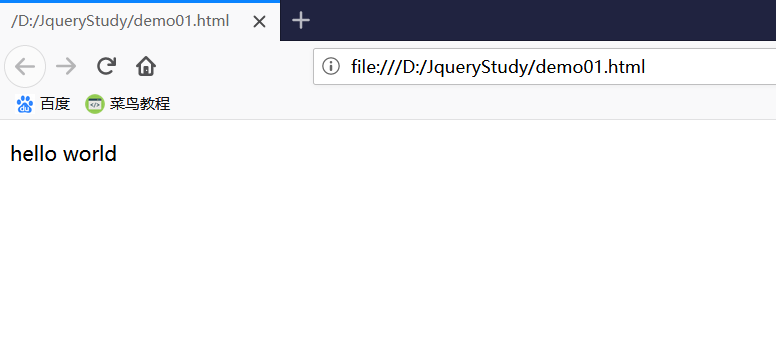


### 修改内容

使用html( )方法修改元素的内容时，如下：



运行结果如下：



如果内容中包含了HTML标签，则会将其转化为HTML元素进行显示。

## text( )方法

text( )方法与html( )方法的区别在于text( )只会读取该元素下所有的文本内容，而不会读取到HTML标签元素。

### 读取内容

JQuery中使用text( )方法读取元素的内容，如下：



### 修改内容

使用JQuery修改元素的内容，如下:



## val( )方法

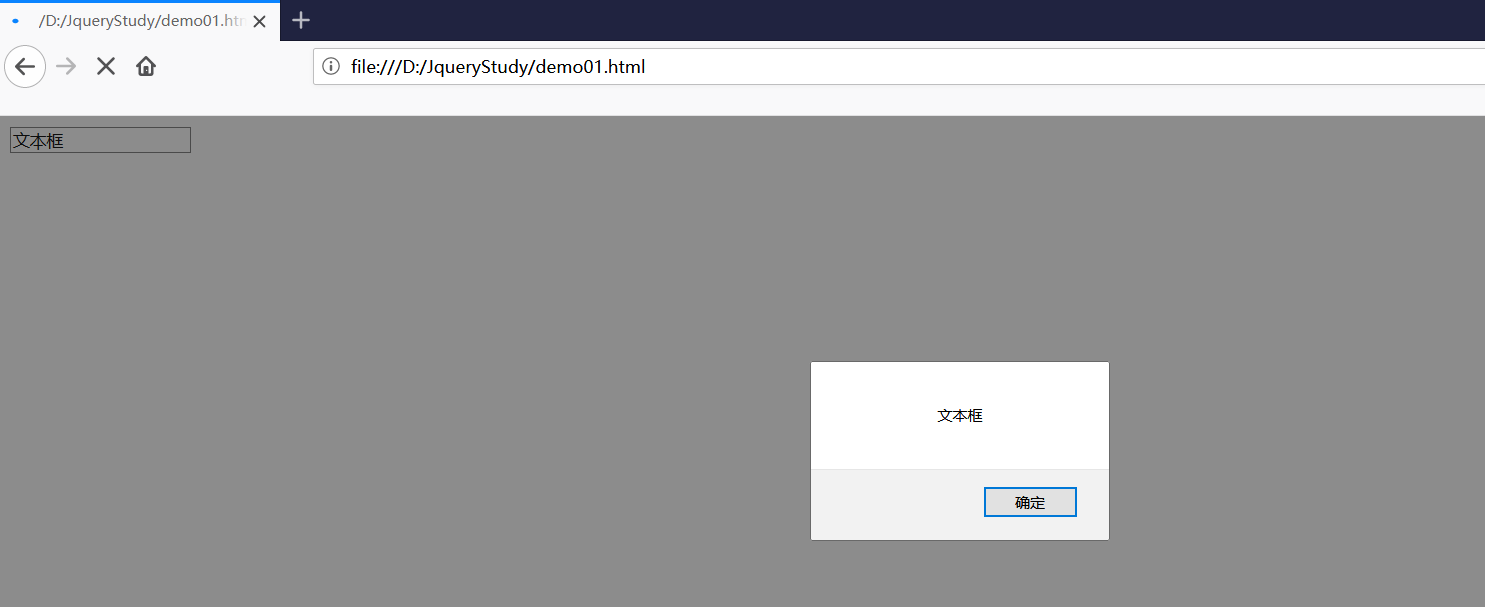
val( )方法用于读取或修改表单标签元素的内容。

### 读取内容

使用val( )方法读取表单内容。如下：



运行结果为：

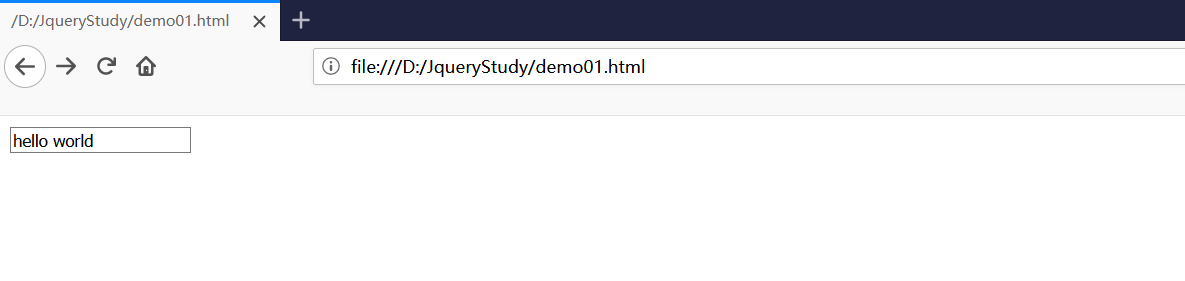


### 修改内容

使用val( )修改表单元素的内容。如下：



运行结果为：



# 操作属性

对某个元素对象的属性进行操作包括设置属性、删除属性、访问属性。在JQuery中，可以使用attr( )方法和prop( )方法完成属性的操作。

## 设置属性

为元素设置属性指的是为元素添加属性或修改属性。可以使用attr( )方法或prop( )方法完成。

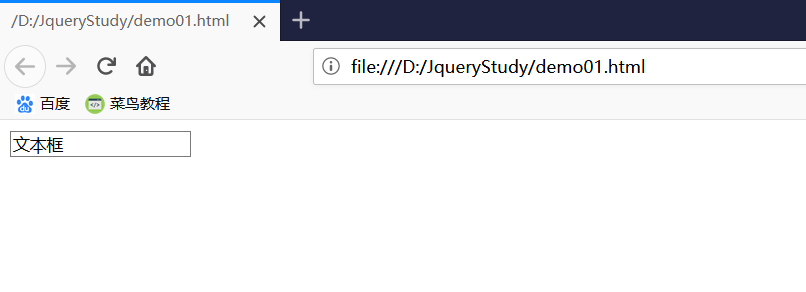
### attr( )方法

使用attr( )方法为元素对象添加属性或修改属性。如下：



其中，attr( )方法的第一个参数表示属性名称，第二个参数表示属性值。

运行结果如下：



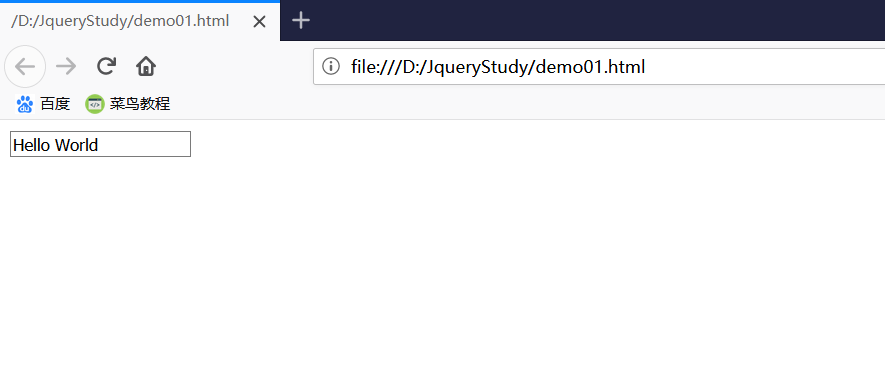
### prop( )方法

使用prop( )方法为元素对象添加属性或修改属性。如下：



prop( )方法的第一个参数值表示属性名，第二个参数值表示属性值。

运行结果如下：



### attr( )与prop()的区别

如果attr( )方法有的属性设置不了可以使用prop( )方法设置；如果prop( )方法有的属性设置不了可以使用attr( )方法设置。

## 访问属性

1）在JavaScript中访问元素对象的属性通过getAttribute( )方法完成。如下：

**var a = document.getElementById( “ID值” ).getAttribute( “属性名” ) ;**

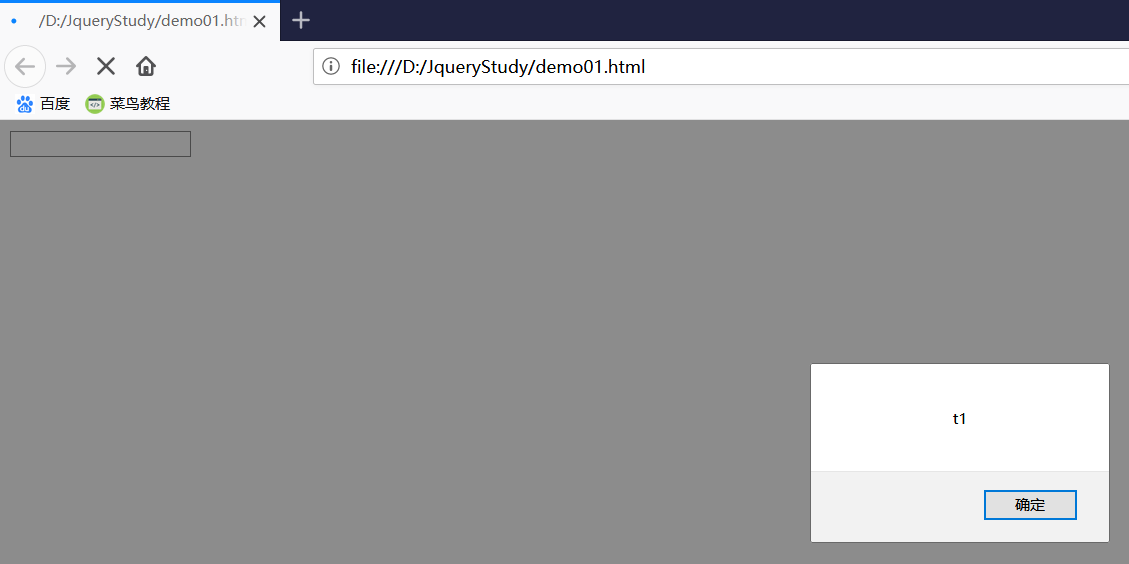
2）在JQuery中，可以使用attr( )方法或prop( )方法来访问元素指定属性的属性值。

### attr( )方法

使用attr( )方法访问元素的属性值。如下：



运行结果如下：

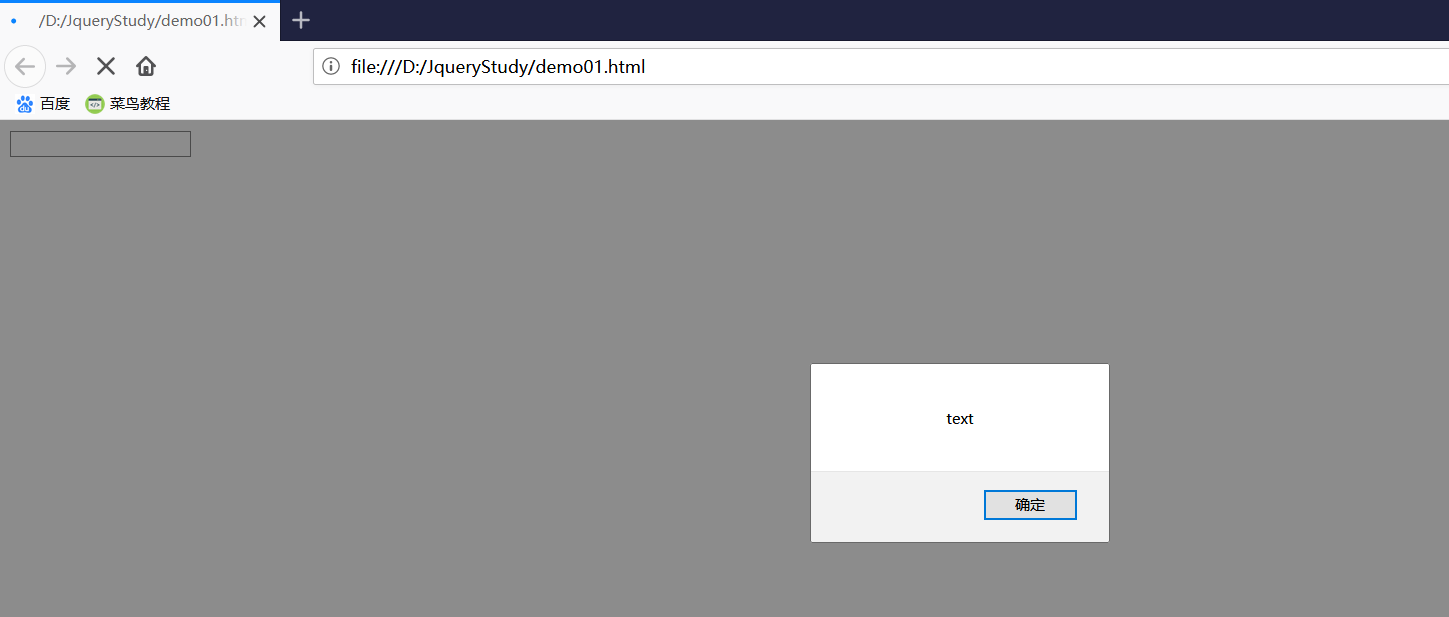


### prop( )方法

使用prop( )方法访问元素的属性值。如下：



运行结果为：



## 删除属性

1. 在JavaScript中，使用removeAttribute( )方法删除元素的属性。语法如下：

**document.getElementById( “ID值” ).removeAttribute( “属性名” ) ;**

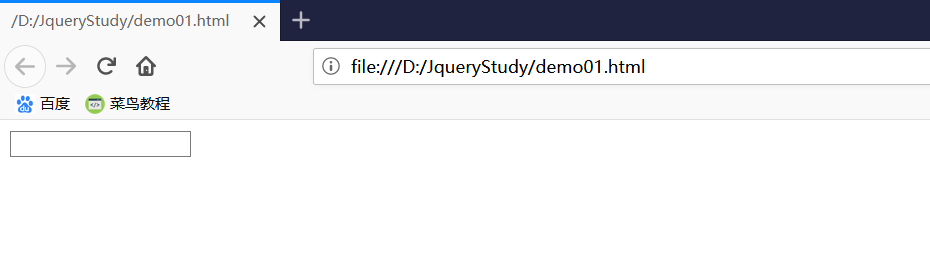
1. 在JQuery中可以使用removeProp( )方法或removeAttr( )方法删除元素的属性。

### removeAttr( )

使用removeAttr( )方法删除元素的属性值。如下：

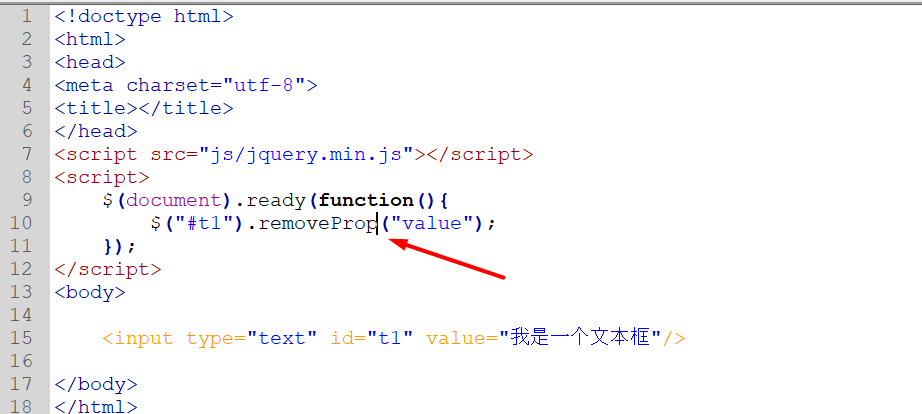


运行结果如下：

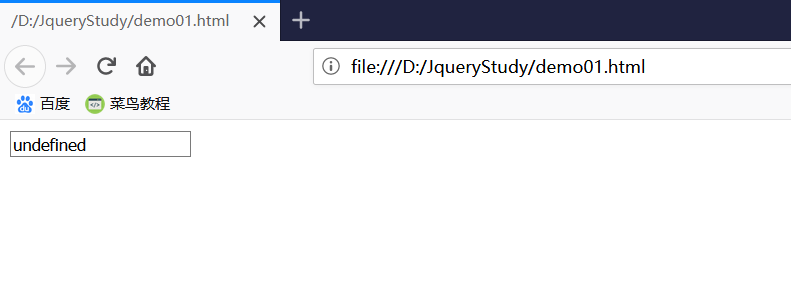


### removeProp( )方法

使用removeProp( )方法删除元素的属性。如下：



运行结果如下：



### removeAttr( )与removeProp( )的区别

1. 在开中，如果使用removeAttr( )不合适或者没有想要的效果，则使用removeProp( )方法完成；如果使用removeProp( )方法不合适或者没有想要的效果，则使用removeAttr( )方法完成。
2. 推荐使用removeAttr( )方法完成，更接近原生。

# 操作样式

操作样式包括操作元素的样式属性、操作元素的样式类。

## 操作样式属性

1）对元素的样式属性操作包括设置样式属性、读取样式属性值。

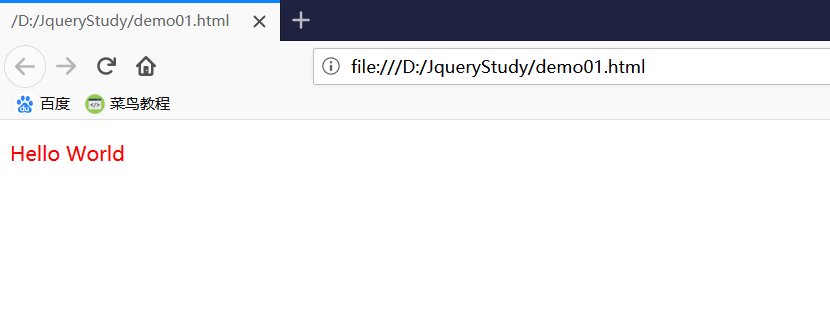
2）在JQuery中，设置样式属性和访问样式属性值都是使用css( )方法完成。

### 设置样式属性

设置元素样式属性指的是为元素添加样式属性或修改元素的样式属性值。如下：



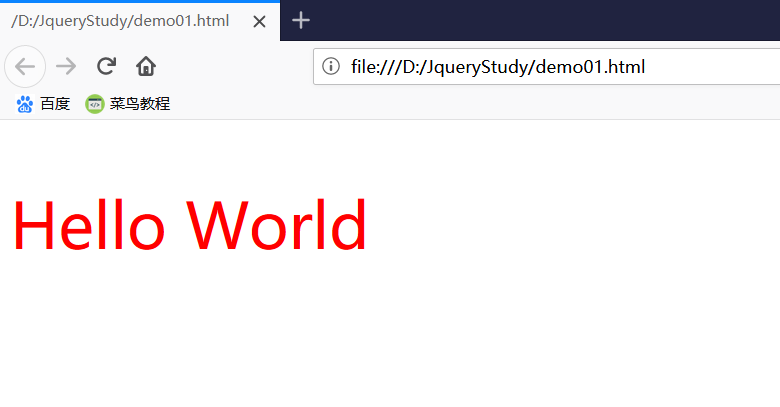
运行结果如下：



如果你需要为元素设置多个样式属性，你可以以对象的形式作为css( )方法的参数。如下：

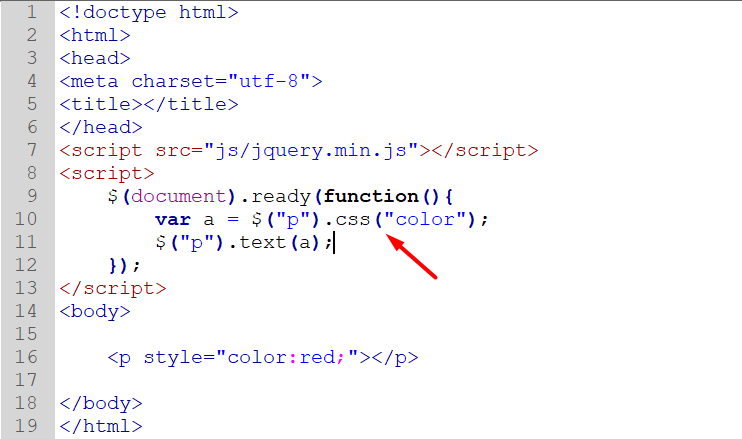


运行结果如下：

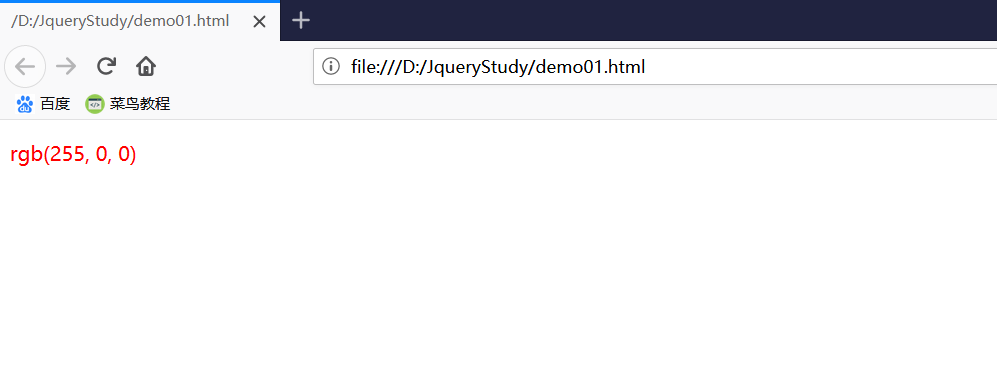


### 访问样式属性

访问元素的样式属性值。如下：



运行结果如下：



## 操作样式类

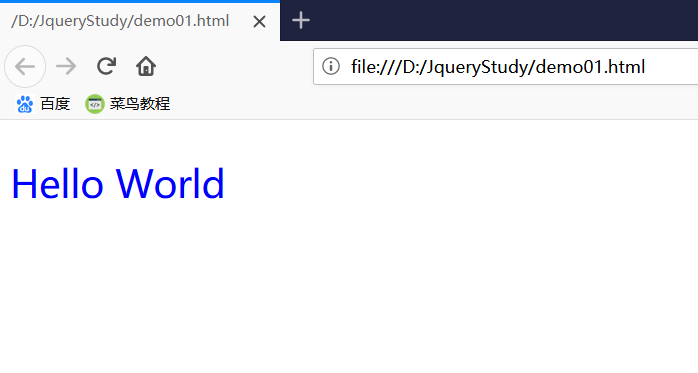
HTML中每一个元素可以拥有class属性，并且class属性的属性值可以是多个CSS类选择器，它们之间使用“，”分割。而操作样式类就是对元素的class属性值进行操作。

### 增加样式类

当元素的class属性只应用了css中的一个类选择器时，这是我们可以使用addClass( )方法增加样式类。如下：



运行结果为：



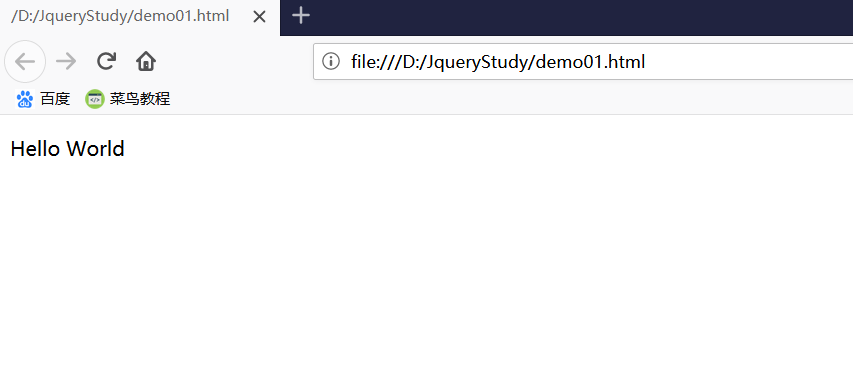
如果要同时增多个多个样式类，多个样式类之间使用“，”分隔。

### 删除样式类

在JQuery中可以使用removeClass( )方法某个删除样式类。如下：



运行结果如下：



### 判断类样式

在JavaScript中，可以使用hasAttribute( )方法来判断元素是否应用了某样式。语法如：

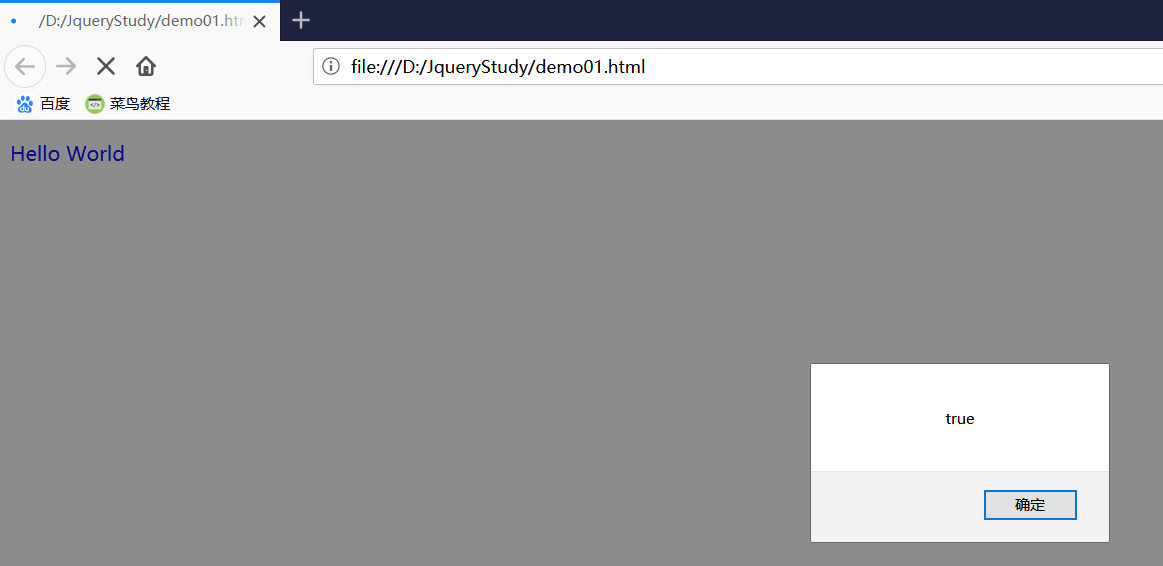
**var flag=document.getElementById(“ID值”).hasAttribute(“样式类名称”) ;**

结果返回一个boolean值。

在JQuery中，你可以使用hasClass( )方法判断元素是否包含指定的样式类。如下：



运行结果如下：



# 操作节点

对节点的操作也就是对元素的操作。包括创建节点、插入节点、删除节点、复制节点、替换节点、移动节点。

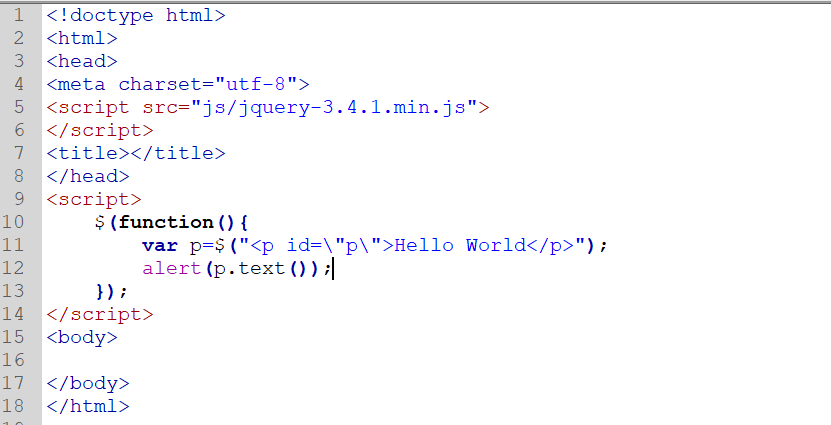
## 创建节点

在JQuery中，使用$( )方法即可创建一个包含内容或属性的节点。其语法格式如下：

**$( html )**

其中，html是一个HTML字符串，JQuery将根据该字符串创建一个DOM对象，并将该对象包装为JQuery对象返回。

实例代码如下：



该实例创建了一个<p>元素，并设置了内容和ID属性。创建了<p>元素后，返回该元素的对象，并取得其内容，以警告框的形式呈现出来。

## 插入节点

创建节点后，根据要插入的位置，可分为内部插入和外部插入。

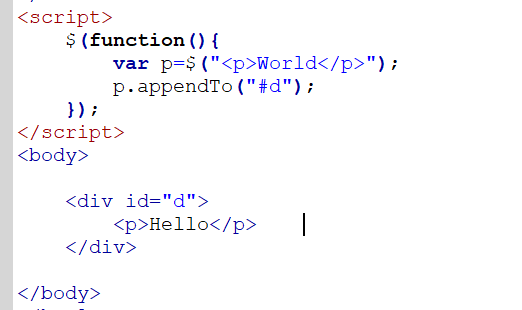
1. 内部插入：将创建的节点插入到目标元素内部的最前面或最后面，作为目标元素的 直接子元素存在。
2. 外部插入：将创建的节点插入到目标元素的前面或后面，作为目标元素的同级兄弟元素存在。

### 内部插入

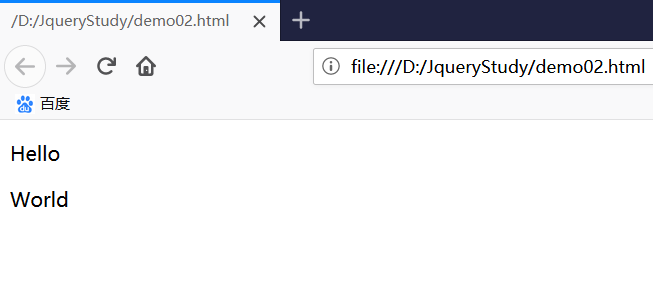
JQuery中定义了2个方法用于实现内部插入。如下：

|  |  |
| --- | --- |
| 方法 | 含义 |
| appendTo( ) | 将创建的元素插入到目标元素内部的最后面，以子级元素的形式存在 |
| prependTo( ) | 将创建的元素插入到目标元素内部的最前面，以子级元素的形式存在 |

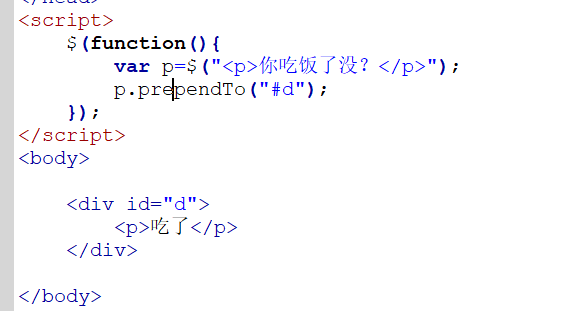
1. appendTo( )用法的示例如下：



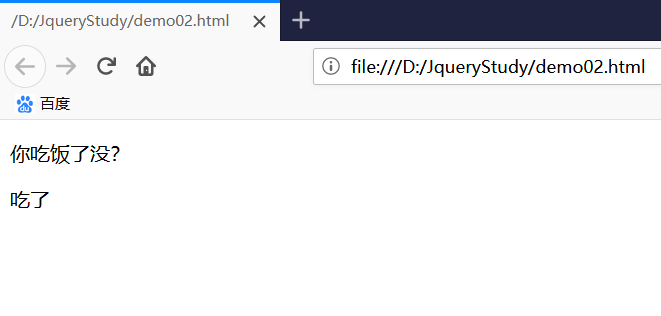
运行结果：



1. prependTo( )用法的示例如下：



运行结果：

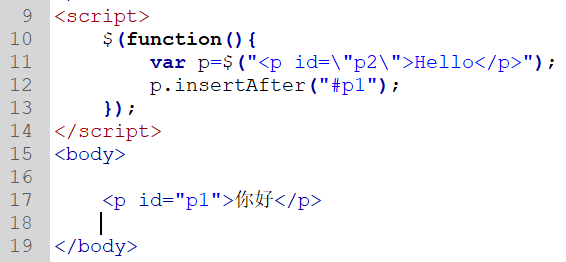


### 外部插入

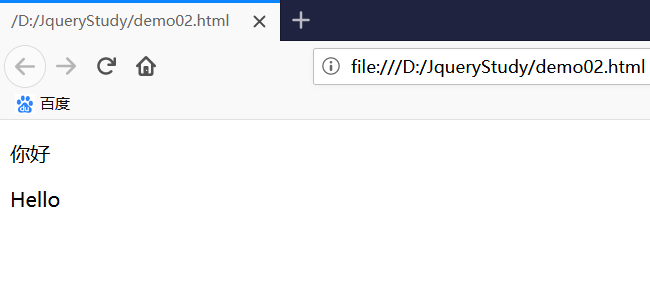
JQuery中定义了2个方法用于实现外部插入。如下：

|  |  |
| --- | --- |
| 方法 | 含义 |
| insertAfter( ) | 将创建的元素插入到目标元素的后面，以同级的形式存在 |
| insertBefore( ) | 将创建的元素插入到目标元素的前面，以同级的形式存在 |

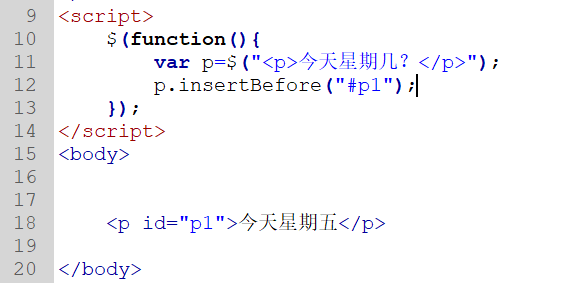
1. insertAfter( )用法示例如下：



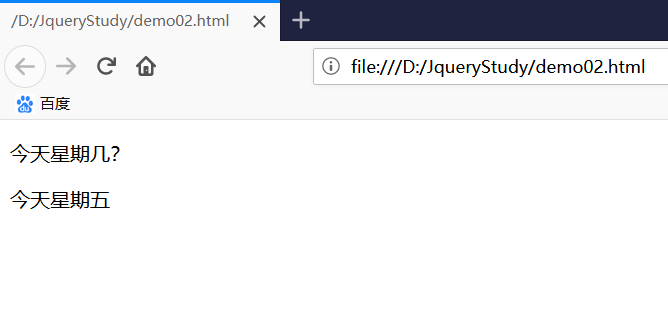
运行结果：



1. insertBefore( )用法示例如下：



运行结果：



## 删除节点

JQuery中定义了3种方法用于删除匹配的元素。如下：

|  |  |
| --- | --- |
| 方法 | 含义 |
| remove( ) | 删除所有匹配的元素（不保留数据） |
| empty( ) | 删除所有匹配元素中的内容 |
| detach( ) | 删除所有匹配的元素（保留数据） |

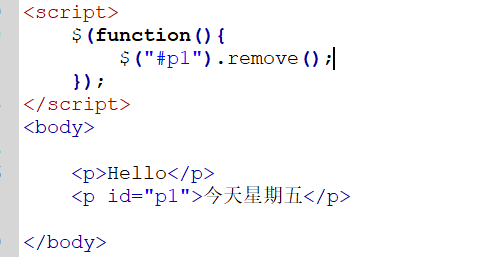
1. remove( )语法格式如下：

**remove ( [selector] )**

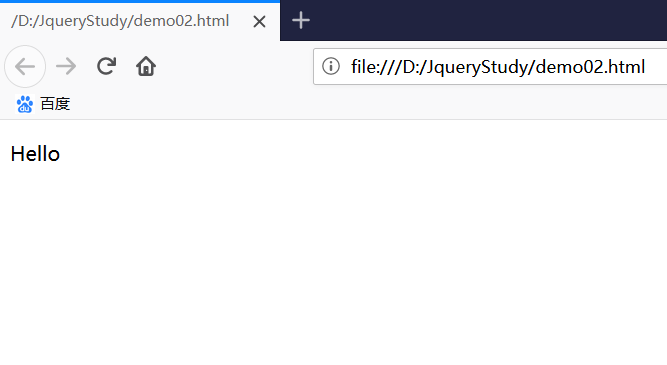
可选参数selector表示一个任意类型的选择器，用来过滤出不需要删除的元素；删除元素后，将返回这个被删除的元素。

需要注意的是使用remove( )方法删除元素时，还将删除该元素的一切，包括绑定的事件、与该元素相关的数据等。

示例代码如下：



运行结果为：

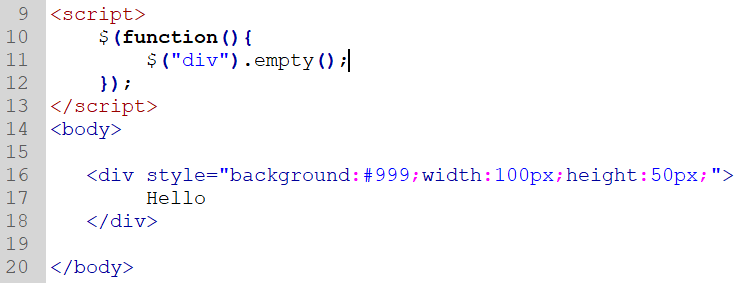


1. empty( )语法格式如下：

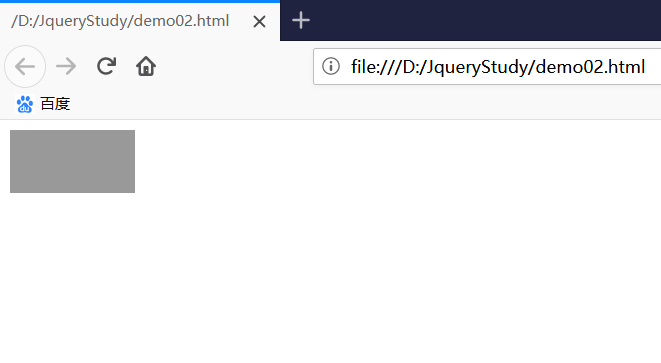
**empty ( )**

该方法不接受任何参数，可以删除匹配元素中的内容，包括子元素中的内容。但不删除元素及其子元素，只是删除内容。

示例如下：



运行结果为：



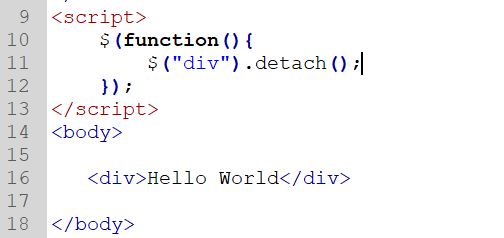
1. detach( )语法格式为：

**detach( [selector] )**

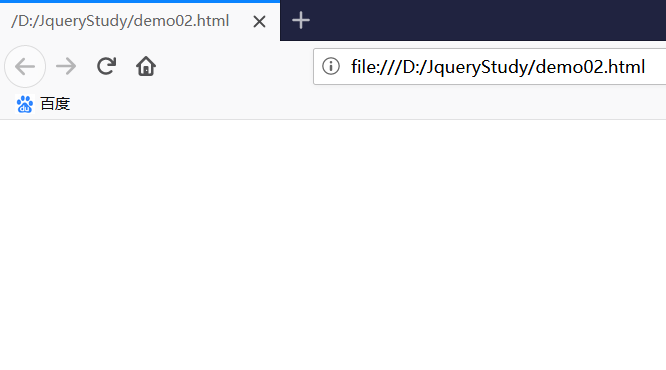
可选参数selector表示一个任意类型的选择器，可以过滤掉不需要被删除的元素。

detach( )方法与remove( )方法不同的是，detach( )方法能够保存所有的数据、事件等；而remove( )方法不可以。所以，当需要删除一个元素，不久又将该元素插入DOM中，使用detach( )方法很有效。

示例如下：



运行结果为：



## 复制节点

JQuery中定义了clone( )方法用于复制节点。其语法为：

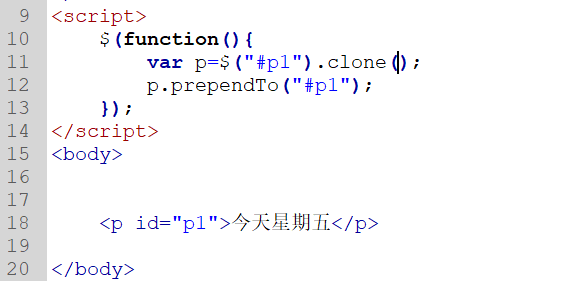
**clone( [withDataAndEvents] )**

**clone( [withDataAndEvents] , [deepWithDataAndEvents] )**

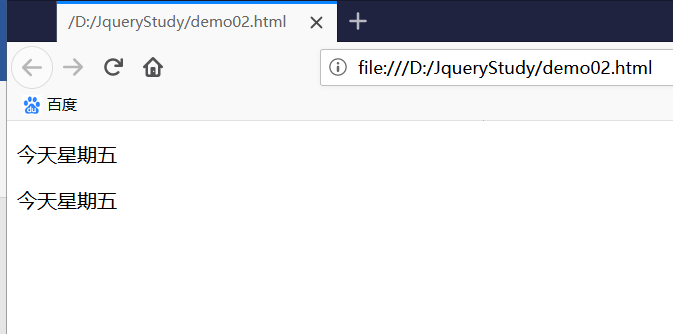
withDataAndEvents表示是否复制事件处理函数等数据，默认值为false。

deepWithDataAndEvents表示是否对事件处理函数和复制元素的所有子元素是否应该 被复制，默认值与withDataAndEvents值一致。

示例代码如下：



运行结果如下：



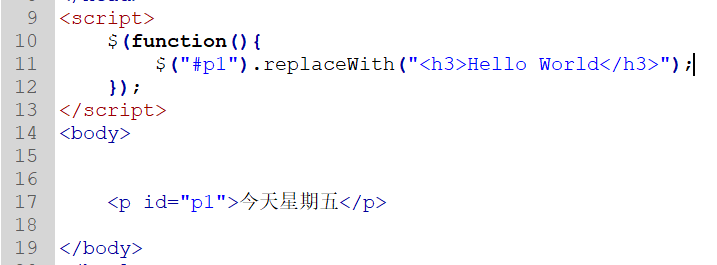
## 替换节点

JQuery中定义了replaceWith( )方法用于替换节点。其语法为：

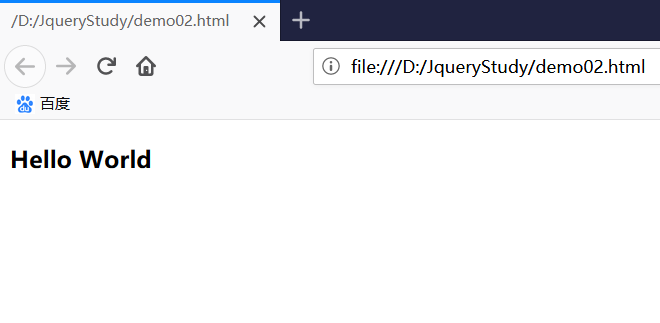
**replaceWith ( newContent )**

参数newContent表示要替换的新内容，可以是HTML字符串或DOM元素或者JQuery对象。

实例代码如下：



运行结果为：

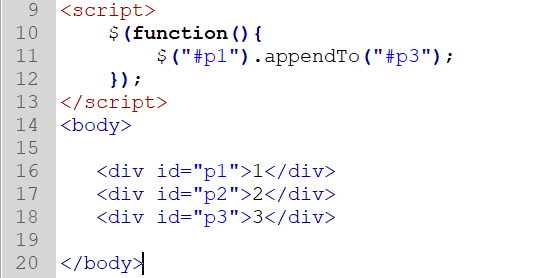


## 移动节点

JQuery中并没有提供移动节点的方法，但可以根据DOM操作的链式语法来实现节点的移动。

1. 方法一

原理：使用appendTo( )实现。如下：



1. 方法二

原理：使用detach( )删除要移动的元素，删除后会获得其对象，然后通过插入将删

除的对象插入到指定位置。如下：

