jQuery

day01

黑马程序员
www.itheima.com
传智教育旗下
高端IT教育品牌

课程介绍

学习目标

Learning Objectives

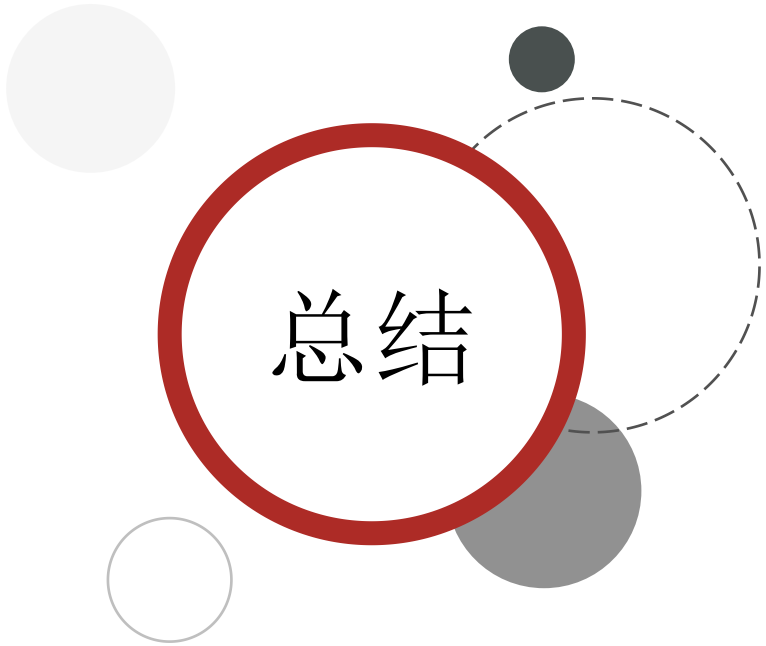
1. 使用jQuery开发常见的网页应用

目录

Contents

课程介绍


黑马程序
www.itheima.
账号登录
安全登录
有什么新鲜事想告诉大家？
已输入5字
表情
图片
视频
话题
头条文章
发布
还没有微博？立即注册！
基它登录：



1. 这个阶段的学习目标是什么?

使用 jQuery 开发常见的网页应用

认识jQuery

认识jQuery

选择器

基本
- #id
- element
- .class
- *
- selector1,selector2,selectorN

层级
- ancestor descendant
- parent > child
- prev + next
- prev ~ siblings

基本筛选器
- :first
- :not(selector)
- :even
- :odd
- :eq(index)
- :gt(index)
- :lang 1.9+
- :last
- :lt(index)
- :header
- :animated
- :focus
- :root 1.9+
- :target 1.9+

内容
- :contains(text)
- :empty
- :has(selector)
- :parent

可见性
- :hidden
- :visible

属性
- [attribute]
- [attribute=value]
- [attribute!=value]
- [attribute^=value]
- [attribute$=value]
- [attribute*=value]
- [attrSel1][attrSel2][attrSelN]

子元素
- :first-child

核心

jQuery 核心函数
- jQuery([sel,[context]])
- jQuery(html,[ownerDoc]) 1.8*
- jQuery(callback)
- ~~jQuery.holdReady(hold)~~ 3.2-
- jQuery.readyException(error) 3.1+

jQuery 对象访问
- each(callback)
- ~~size()~~ 1.8-
- length
- selector
- context
- get([index])
- index([selector|element])

数据缓存
- data([key],[value])
- removeData([name|list]) 1.7*
- ~~$.data(ele,[key],[val])~~ 1.8-

队列控制
- queue(e,[q])
- dequeue([queueName])
- clearQueue([queueName])

插件机制
- jQuery.fn.extend(object)
- jQuery.extend(object)

多库共存
- jQuery.noConflict([ex])

ajax

ajax 请求
- $.ajax(url,[settings])
- $.get(url,[data],[fn],[type])
- $.getJSON(url,[data],[fn])
- $.getScript(url,[callback])
- $.post(url,[data],[fn],[type])

ajax 事件
- ajaxComplete(callback)
- ajaxError(callback)
- ajaxSend(callback)
- ajaxStart(callback)
- ajaxStop(callback)
- ajaxSuccess(callback)

其它
- load(url,[data],[callback])
- $.ajaxPrefilter([type],fn)
- $.ajaxSetup([options])
- serialize()
- serializeArray()

回调函数

- cal.add(callbacks) 1.7+
- cal.disable() 1.7+
- cal.empty() 1.7+
- cal.fire(arguments) 1.7+
- cal.fired() 1.7+
- cal.fireWith([c] [,a]) 1.7+
- cal.has(callback) 1.7+
- cal.lock() 1.7+
- cal.locked() 1.7+
- cal.remove(callbacks) 1.7+
- $.callbacks(flags) 1.7+

属性

属性
- attr(name|pro|key,val|fn)
- removeAttr(name)
- prop(n|p|k,v|f)
- removeProp(name)

CSS 类
- addClass(class|fn)
- removeClass([class|fn])
- toggleClass(class|fn[,sw])

HTML代码/文本/值
- html([val|fn])
- text([val|fn])
- val([val|fn|arr])

事件对象

- eve.currentTarget
- eve.data
- eve.delegateTarget 1.7+
- eve.isDefaultPrevented()
- eve.isImmediatePropag...()
- eve.isPropagationStopped()
- eve.namespace
- eve.pageX
- eve.pageY
- eve.preventDefault()
- eve.relatedTarget
- eve.result
- eve.stopImmediatePro...()
- eve.stopPropagation()
- eve.target
- eve.timeStamp
- eve.type
- eve.which

CSS

CSS
- css(name|pro|[,val|fn]) 1.9*
- jQuery.cssHooks

位置
- offset([coordinates])
- position()
- scrollTop([val])
- scrollLeft([val])

尺寸
- height([val|fn])
- width([val|fn])
- innerHeight()
- innerWidth()
- outerHeight([options])
- outerWidth([options])

延迟对象

- def.done(d,[d])
- def.fail(failCallbacks)
- ~~def.isRejected()~~ 1.7-
- ~~def.isResolved()~~ 1.7-
- def.reject(args)
- def.rejectWith(c,[a])
- def.resolve(args)
- def.resolveWith(c,[a])
- ~~def.then(d[,f][,p])~~ 1.8*
- def.promise([ty],[ta])
- ~~def.pipe([d],[f],[p])~~ 1.8-
- def.always(al,[al])
- def.notify(args) 1.7+
- def.notifyWith(c,[a]) 1.7+
- def.progress(proCal) 1.7+
- def.state() 1.7+

文档处理

内部插入
- append(content|fn)
- appendTo(content)
- prepend(content|fn)
- prependTo(content)

外部插入
- after(content|fn)
- before(content|fn)
- insertAfter(content)
- insertBefore(content)

包裹
- wrap(html|ele|fn)
- unwrap()
- wrapAll(html|ele)
- wrapInner(html|ele|fn)

替换
- replaceWith(content|fn)
- replaceAll(selector)

删除
- empty()
- remove([expr])
- detach([expr])

复制
- clone([Even[,deepEven]])

事件

页面载入
- ready(fn)

事件处理
- on(eve,[sel],[data],fn) 1.7+
- off(eve,[sel],[fn]) 1.7+
- ~~bind(type,[data],fn)~~ 3.0-
- one(type,[data],fn)
- trigger(type,[data])
- triggerHandler(type, [data])
- ~~unbind([,[d|f]])~~ 3.0-

事件委派
- ~~live(type,[data],fn)~~ 1.7-
- ~~die(type,[fn])~~ 1.7-
- ~~delegate(s,[t],[d],fn)~~ 3.0-
- ~~undelegate([s,[t],fn])~~ 3.0-

事件切换
- hover([over,]out)
- toggle([spe],[eas],[fn]) 1.9*

事件
- blur([[data],fn])
- change([[data],fn])
- click([[data],fn])
- dblclick([[data],fn])
- ~~error([[data],fn])~~ 1.8-
- focus([[data],fn])
- focusin([data],fn)
- focusout([data],fn)
- keydown([[data],fn])
- keypress([[data],fn])
- keyup([[data],fn])
- mousedown([[data],fn])
- mouseenter([[data],fn])
- mouseleave([[data],fn])
- mousemove([[data],fn])
- mouseout([[data],fn])
- mouseover([[data],fn])
- mouseup([[data],fn])
- resize([[data],fn])
- scroll([[data],fn])
- select([[data],fn])
- submit([[data],fn])
- ~~unload([[data],fn])~~ 1.8-

效果

基本
- show([s,[e],[fn]])
- hide([s,[e],[fn]])
- toggle([s],[e],[fn])

滑动
- slideDown([s],[e],[fn])
- slideUp([s,[e],[fn]])
- slideToggle([s],[e],[fn])

淡入淡出
- fadeIn([s],[e],[fn])
- fadeOut([s],[e],[fn])
- fadeTo([[s],o,[e],[fn]])
- fadeToggle([s,[e],[fn]])

自定义
- animate(p,[s],[e],[fn]) 1.8*
- stop([c],[j]) 1.7*
- delay(d,[q])
- finish([queue]) 1.9+

设置
- jQuery.fx.off
- ~~jQuery.fx.interval~~ 3.0-

其它

正则表达式
HTML5速查表
源码下载

工具

浏览器及特性检测
- ~~$.support~~ 1.9-
- ~~$.browser~~ 1.9-
- $.browser.version
- ~~$.boxModel~~

数组和对象操作
- $.each(object,[callback])
- $.extend([d],tgt,obj1,[objN])
- $.grep(array,fn,[invert])
- ~~$.sub()~~ 1.9-
- $.when(deferreds)
- $.makeArray(obj)
- $.map(arr|obj,callback)
- $.inArray(val,arr,[from])
- $.toArray()
- $.merge(first,second)
- ~~$.unique(array)~~ 3.0-
- $.uniqueSort(array) 3.0+
- ~~$.parseJSON(json)~~ 3.0-
- $.parseXML(data)

函数操作
- $.noop
- $.proxy(function,context)

测试操作
- $.contains(c,c)
- $.type(obj)
- ~~$.isArray(obj)~~ 3.2-
- ~~$.isFunction(obj)~~ 3.3-
- $.isEmptyObject(obj)
- $.isPlainObject(obj)
- ~~$.isWindow(obj)~~ 3.3-
- $.isNumeric(value) 1.7+

字符串操作
- $.trim(str)

URL
- $.param(obj,[traditional])

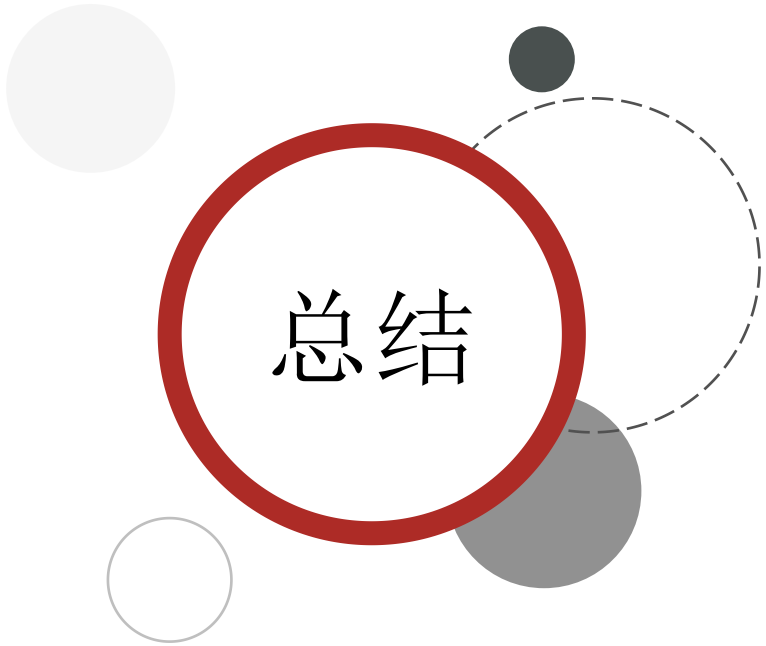
插件编写
- $.error(message)
- $.fn.jquery

筛选

过滤
- eq(index|-index)
- first()
- last()
- hasClass(class)
- filter(expr|obj|ele|fn)
- is(expr|obj|ele|fn)
- map(callback)
- has(expr|ele)
- not(expr|ele|fn)
- slice(start,[end])

查找
- children([expr])
- closest(e|o|e) 1.7*
- find(e|o|e)
- next([expr])
- nextAll([expr])
- nextUntil([e|e][,f])
- offsetParent()
- parent([expr])
- parents([expr])
- parentsUntil([e|e][,f])
- prev([expr])
- prevAll([expr])
- prevUntil([e|e][,f])
- siblings([expr])

串联
- add(e|e|h|o[,c]) 1.9*
- ~~andSelf()~~ 1.8-
- addBack() 1.9+
- contents()
- end()



1. jQuery 是JavaScript类库
2. 使用它封装的方法可以极大的提升开发效率

03 使用准备

使用准备

DOM语法

```
let liArr = document.querySelectorAll('li')
for (let i = 0; i < liArr.length; i++) {
  liArr[i].onclick = function () {
    this.style.backgroundColor = 'pink'
  }
}
```

jQuery语法

```
$('li').click(function () {
  $(this).css('backgroundColor', 'pink')
})
```

使用准备

就两步

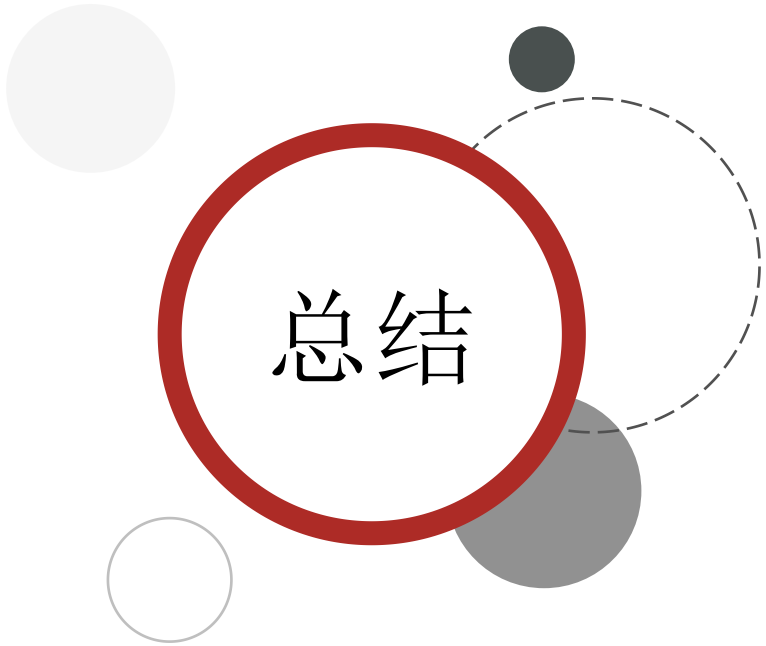
- 下包：把 jQuery 下载到本地
- 导包：在希望使用的页面中导入下载好的 jQuery

小测试

```
console.log($)

$('body').css('backgroundColor','yellowgreen')
```

- 下载地址：https://jquery.com/



1. 使用 jQuery 的准备工作是哪两步？

   下包，导包

2. 完整版本和压缩版本的 jQuery 功能上有区别吗？

   没有

3. 文件名中有 mini 的是压缩版本还是完整版本？

   压缩版本

04 版本区别

版本区别

目前最新版本是：3.5.1

版本区别

可以看成：3.x

- 3：大版本,和上一个版本相比有很多功能
- x：bug修复或者新增小功能

版本区别

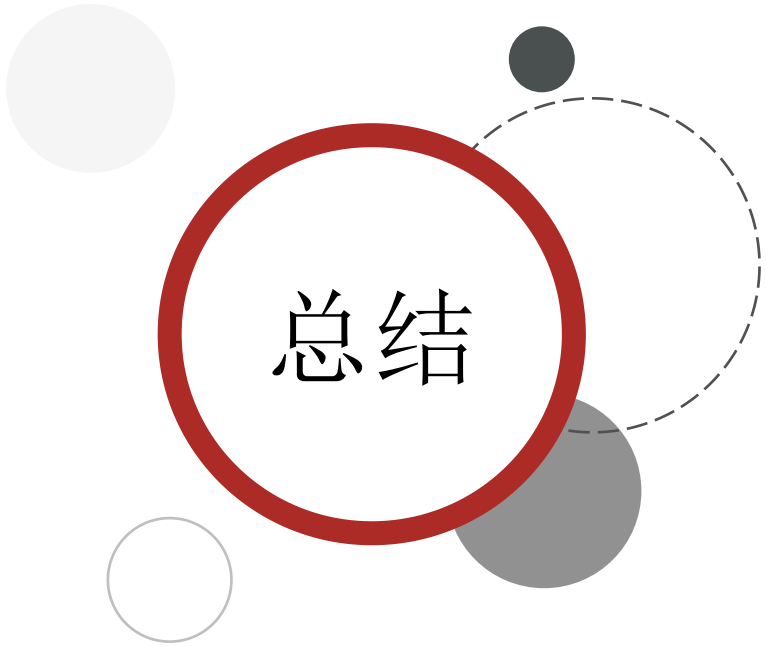
目前有 3 个大版本

- 1.x：兼容低版本的ie
- 2.x：不兼容低版本的ie ， 并且不再更新
- 3.x：　不兼容低版本的ie ， 更新中

下载地址

兼容信息查看：https://jquery.com/browser-support/

历史版本下载：https://code.jquery.com/



1. 有几个大版本 ， 他们分别是什么?

   1.x , 2.x , 3.x

2. 如果不考虑兼容低版本的ie ， 使用哪个版本?

   3.x

04 选择器

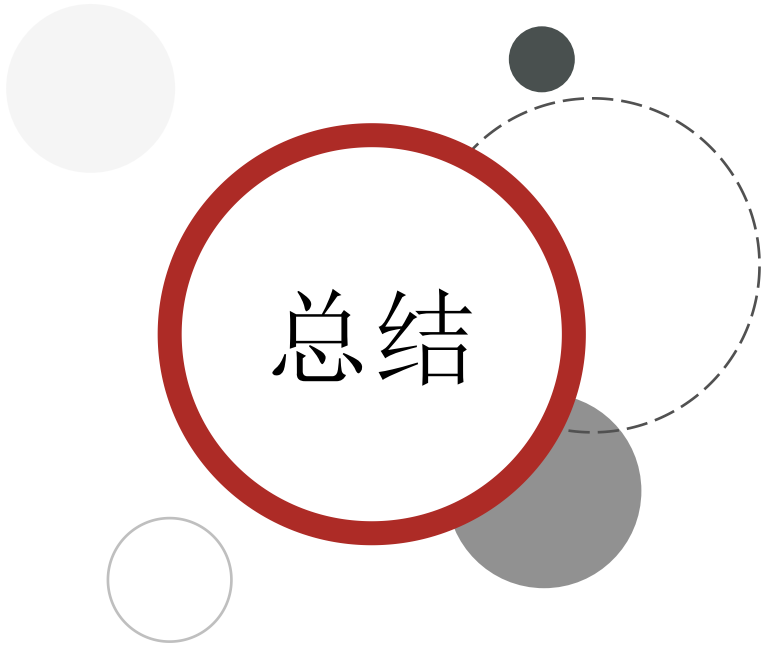
选择器

jQuery 中通过选择器来获取 DOM 节点，功能类似于原生的 querySelectorAll 方法，支持的选择器与 CSS 的选择器几乎一致。

语法

```
// 语法
$('选择器')
// 修改背景色
$('选择器').css('backgroundColor', 'yellowgreen')
```


```
f (selector, context) {
    // The jQuery object is actually just the init constructor 'enhanced'
    // Need init if jQuery is called (just allow error to be thrown if not included)
    return n…
```



1. jQuery 的选择器如何使用?

   $('选择器')

jQuery对象

jQuery对象

jQuery 中利用选择器获取到的并非原生的 DOM 对象，而是 jQuery 对象

示例代码

```
// jQuery 获取p标签
$('p')
```

jQuery对象

jQuery 中利用选择器获取到的并非原生的 DOM 对象，而是 jQuery 对象

示例代码

```
// jQuery 获取p标签 改变背景色为 粉色
$('p').css('backgroundColor', 'pink')
```

jQuery对象

jQuery 中利用选择器获取到的并非原生的 DOM 对象，而是 jQuery 对象

示例代码

```
// jQuery 获取p标签 改变背景色为 粉色
$('p').css('backgroundColor', 'pink')
// dom 获取p标签 改变背景色为 粉色
document.querySelector('p').style.backgroundColor = 'pink'
```

- jQuery对象 和 DOM对象 的语法不能混用

jQuery对象

jQuery 中利用选择器获取到的并非原生的 DOM 对象，而是 jQuery 对象

示例代码

```
// jQuery 获取p标签 改变背景色为 粉色
$('p').style.backgroundColor = 'pink'
// dom 获取p标签 改变背景色为 粉色
document.querySelector('p').css('backgroundColor', 'pink')
```

- jQuery对象 和 DOM对象 的语法不能混用

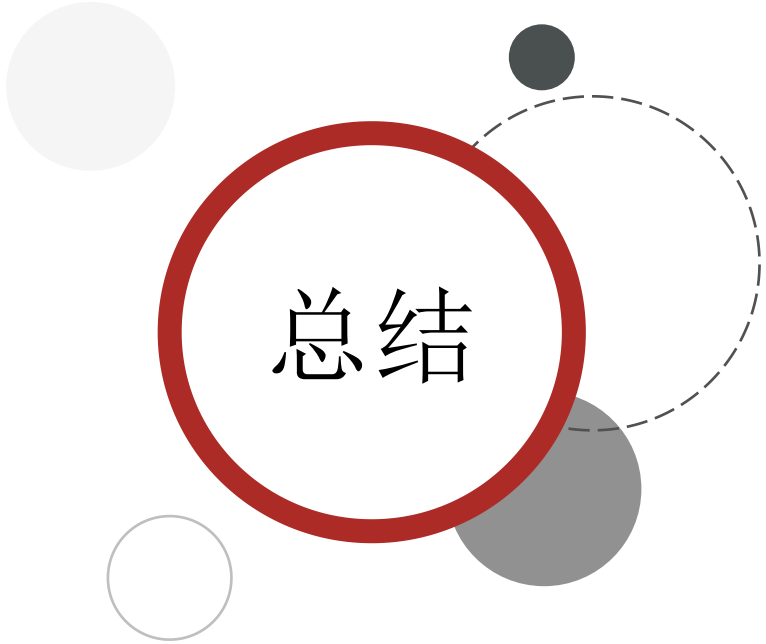
jQuery对象

jQuery 中利用选择器获取到的并非原生的 DOM 对象，而是 jQuery 对象

语法

```
// 选择器获取
$('选择器')
// dom对象转换
$(dom对象)
```

- jQuery对象 和 DOM对象 的语法不能混用

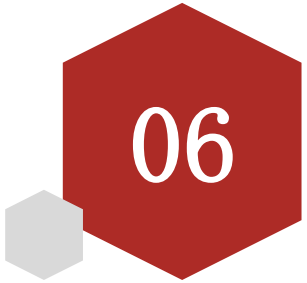


1. 调用$方法传入选择器或dom元素获取到的是什么对象?

   jQuery对象

2. jQuery对象的方法,比如css方法放在什么位置?

   原型



事件绑定

事件绑定

在 jQuery 中以原生事件类型的名称为依据 ， 封装了相对应的事件处理方法

语法

```
$('选择器')
```

事件绑定

在 jQuery 中以原生事件类型的名称为依据 ， 封装了相对应的事件处理方法

语法

```
$('选择器').事件名()
```

事件绑定

在 jQuery 中以原生事件类型的名称为依据 ， 封装了相对应的事件处理方法

语法

```
$('选择器').事件名(function () {
  // 逻辑....
})
```

事件绑定

在 jQuery 中以原生事件类型的名称为依据 ， 封装了相对应的事件处理方法

语法

```
$('选择器').click(function () {
  // 逻辑....
})
```

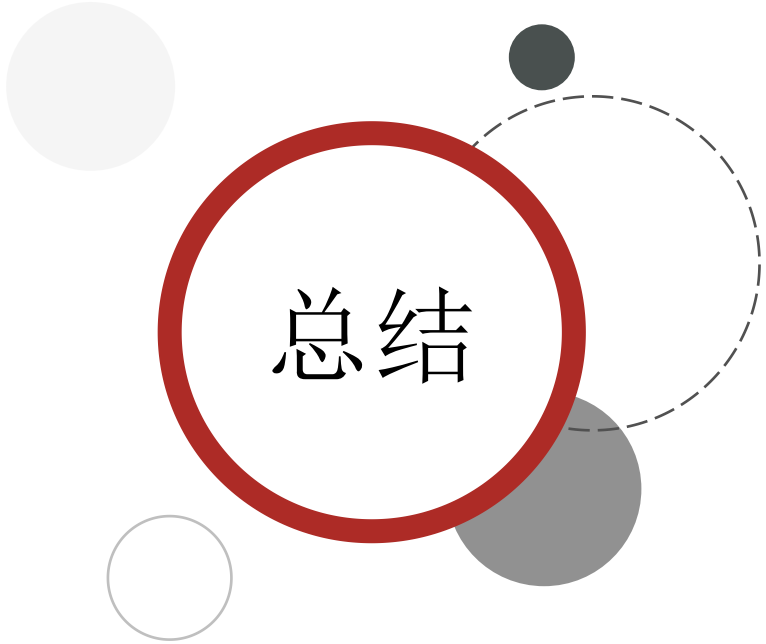
事件绑定

在 jQuery 中以原生事件类型的名称为依据 ， 封装了相对应的事件处理方法

语法

```
$('选择器').dblclick(function () {
  // 逻辑....
})
```

- 事件名开头不需要写 on
- 回调函数中的 this 就是触发事件的 dom 元素



1. 为 jQuery对象 绑定点击事件，方法名是onclick吗?

   click

2. 如何获取触发事件的 dom 元素?

   this

链式编程

链式编程

链式编程 通过点(.) 把多个操作(方法)连续的写下去，形成和 **链子** 一样的结构

语法

```
$('.text').focus(function () {
  console.log('获取焦点')
})
$('.text').blur(function () {
  console.log('失去焦点')
})
```

链式编程

语法

```
$('.text')
  .focus(function () {
    console.log('获取焦点')
  })
  .blur(function () {
    console.log('失去焦点')
  })
```

链式编程

链式编程 通过点(.) 把多个操作(方法)连续的写下去，形成和 **链子** 一样的结构

语法

```
$('.text').focus(回调函数).blur(回调函数)
```

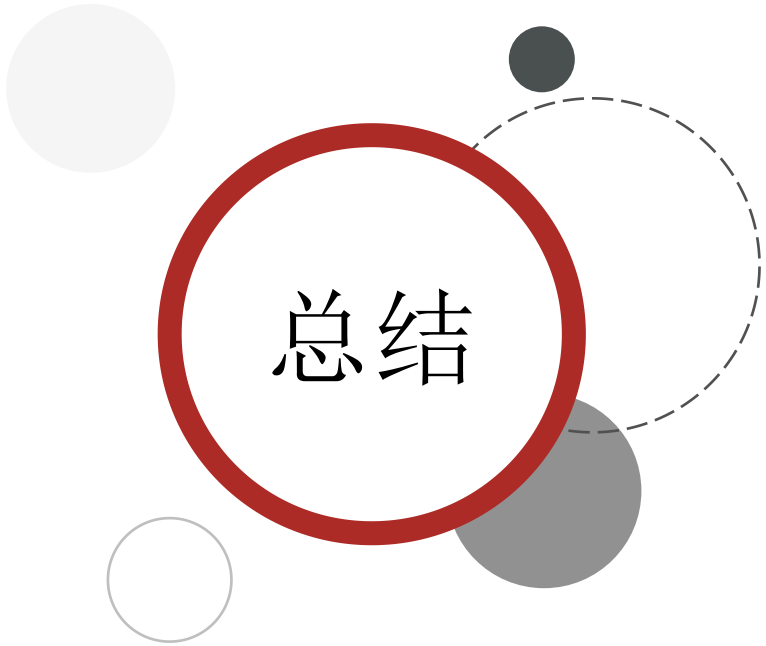
链式编程

链式编程 通过点(.) 把多个操作(方法)连续的写下去，形成和 **链子** 一样的结构

语法

```
$('.text').focus(回调函数).blur(回调函数).change(回调函数)
```

- 大部分 jQ对象 方法的返回值还是同一个 jQ对象



1. 链式编程的含义?

   通过 点 把多个操作连续的写下去，形成和 链子 一样的结构

2. 有没有特殊情况?

   有

08 内容操纵

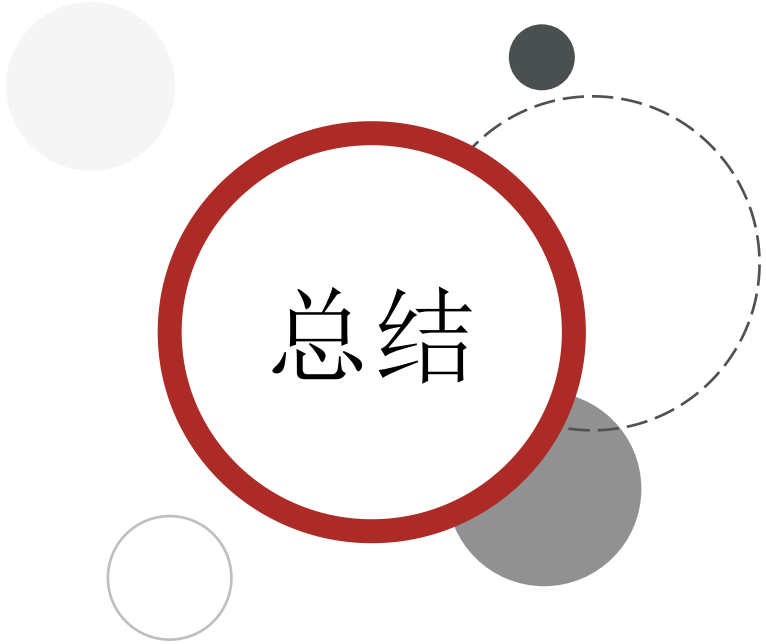
内容操纵

jQuery 中封装了设置和读取网页元素文本内容的方法

语法

```
// 设置
$(‘选择器’).html(’内容’)
$(‘选择器’).text(‘内容’)
// 读取
$(’选择器’).html()
$(’选择器’).text()
```

- 设置时：html 方法解析标签 ， text 不解析标签
- 取值时：html 方法获取标签 ， text 只获取文本
- 有一种使用方式支持链式编程



1. 哪个方法设置的标签会被正常解析？

   html方法

2. 只获取元素的文本使用哪个方法?

   text方法

3. 设置还是取值支持链式编程?

   设置

计数器

计数器

计数器

需求

- 累加
- 递减

步骤

累加 & 递减

需求：点击 加号 累加数字一直到10,点击 减号 递减数字一直到0 ， 到达临界值继续操作则提示用户

分析：

①：绑定点击事件

click ， first-child ， last-child

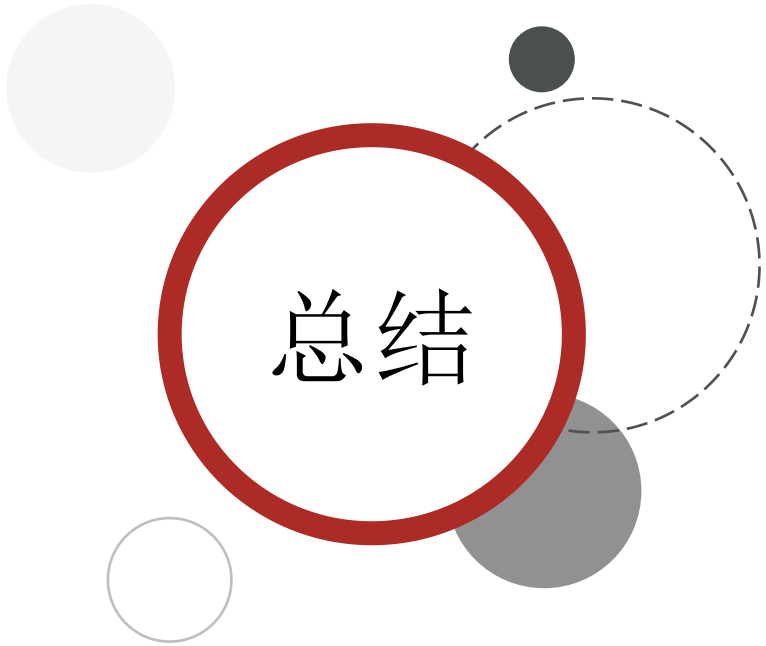
②：修改文本

text方法

③：判断并提示用户

0 到 10

```
<div class="input-num">
        <!-- 减号 -->
        <button> - </button>
        <!-- 内容 -->
        <span>0</span>
        <!-- 加号 -->
        <button> + </button>
</div>
```



1. last-child 和 first-child叫做什么选择器?

   伪类选择器

2. span里面的文本设置和取值用的是什么方法?

   text

10 过滤方法

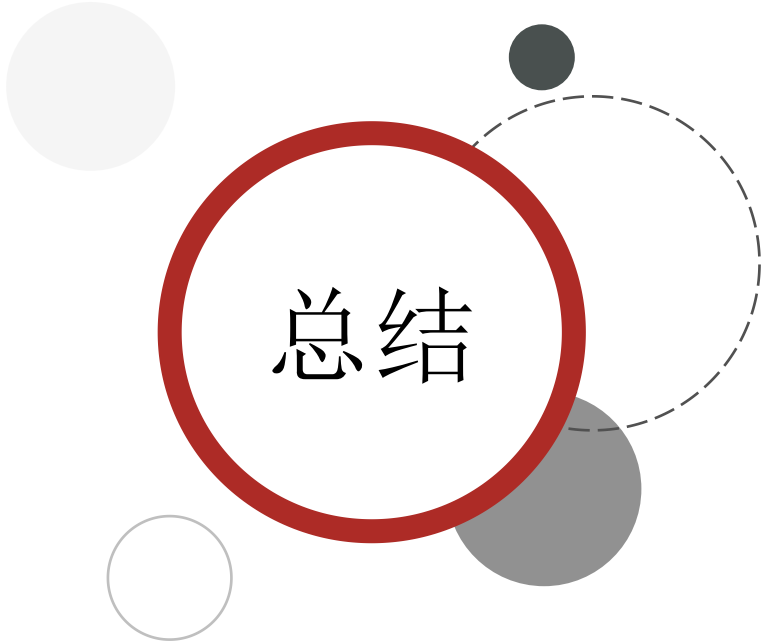
过滤方法

jQuery 中封装了过滤方法,对 jQuery 对象中的 dom 元素再次筛选

语法

```
//  匹配的第一个元素
.first()
// 匹配的最后一个元素
.last()
// 根据索引匹配元素
.eq(索引)
```

- eq 方法的索引从0开始
- 返回的是 jQuery 对象



1. first方法和last方法筛选出来的分别是第几个元素？

   第一个和最后一个

2. eq方法的索引从几开始?

   0

3. 这三个方法的返回值是什么对象

   jQuery对象

样式操纵

样式操纵

jQuery 中对样式的操作进行封装 ， 可以设置或者获取样式

语法

```
// 1. 键值对设置
.css('样式名','值')
.css('backgroundColor','pink')
.css('color','red')
.css('width','200px')
.css('height',200)
```

- 数值类的样式省略单位,默认会使用 px

样式操纵

jQuery 中对样式的操作进行封装 ， 可以设置或者获取样式

语法

```
// 2. 对象方式设置
.css(对象)
.css({
  backgroundColor:'pink',
  color:'red',
  width:'200px',
  height:200
})
```

- 数值类的样式省略单位,默认会使用 px

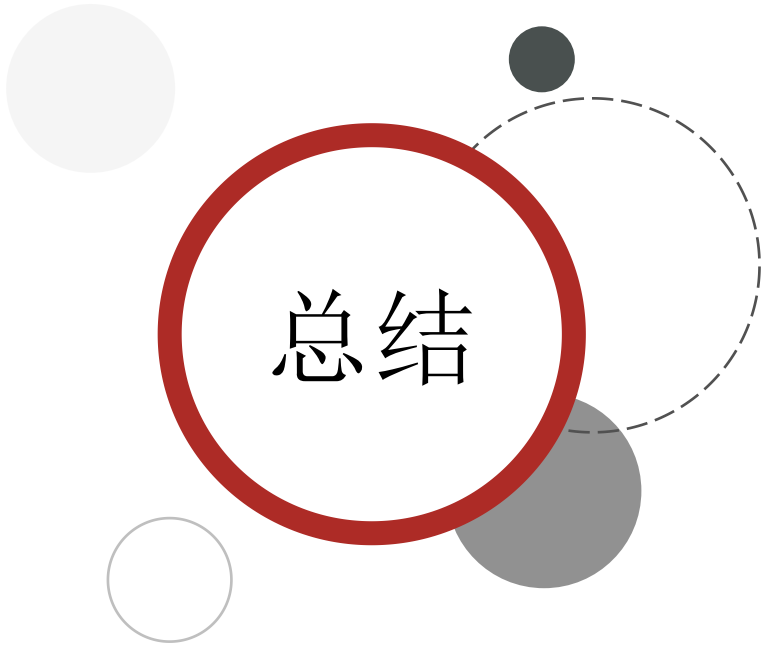
样式操纵

jQuery 中对样式的操作进行封装 ， 可以设置或者获取样式

语法

```
// 3. 样式获取
.css('样式名')
.css('width')
```

- 数值类的样式省略单位,默认会使用 px
- 获取样式需要传递样式名



1. 设置数值类的样式时省略单位,默认值是什么?

   px

2. css方法设置的样式在元素什么位置?

   行内

3. css方法取值时是否需要传递参数?

   需要

属性操纵

属性操纵

jQuery 中对属性的操作进行封装 ， 可以设置、获取和删除属性

测试代码

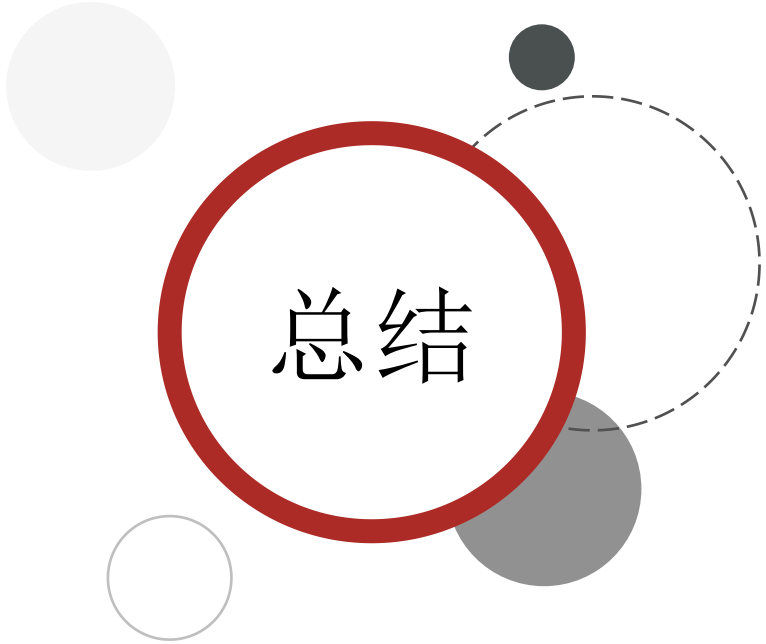
```
<a href="https://www.baidu.com">黑马程序员</a>
<img src="logo.png" info="黑马程序员" />
```

属性操纵

jQuery 中对属性的操作进行封装 ， 可以设置、获取和删除属性

语法

```
// 1.赋值
.attr('属性名','值')
// 2.取值
.attr('属性名')
// 3.删除属性
.removeAttr('属性名')
```



1. attr方法赋值操作时需要传递几个参数?

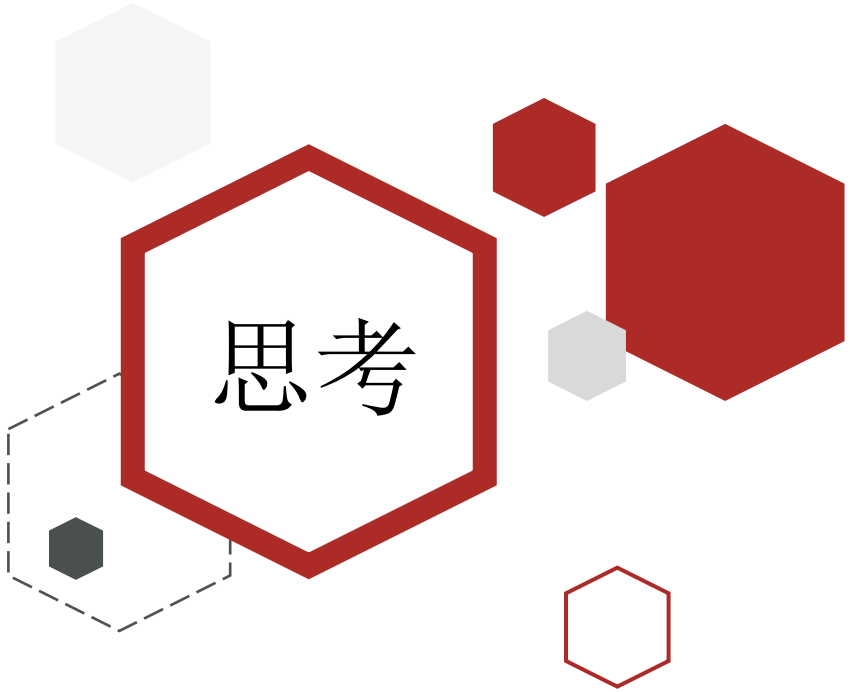
   2个参数

2. attr方法传递一个参数的作用是什么?

   属性取值

3. 删除属性用什么方法?

   removeAttr



1. 如何通过jQuery中改变页面结构?(√)
2. 如何知道动画播放完毕?(√)
3. 如何让动画延迟播放?
4. 如何获取元素高度?

13 图片切换

图片切换

图片切换

需求

- 箭头缩放
- 图片切换

步骤

箭头缩放

需求：两侧箭头鼠标移入之后变大，鼠标移出之后还原

分析：

①：鼠标移入和移出事件

mouseenter ， mouseleave

②：更改箭头缩放

transform ， scale

```
        <!-- 图片 -->
        <img class="cover" src="./images/1.png" alt=
"" />
        <!-- 左箭头 -->
        <a href="javascript:void(0)" class="left">
          <img src="./images/prev.png" alt="" />
        </a>
        <!-- 右箭头 -->
        <a href="javascript:void(0)" class="right">
          <img src="./images/next.png" alt="" />
        </a>
```

图片切换

需求

- 箭头缩放
- 图片切换

步骤

图片切换

需求：点击左右箭头分别切换上一张和下一张图片，第一张时隐藏左箭头，最后一张时隐藏右箭头

分析：

① ：默认隐藏左箭头

display

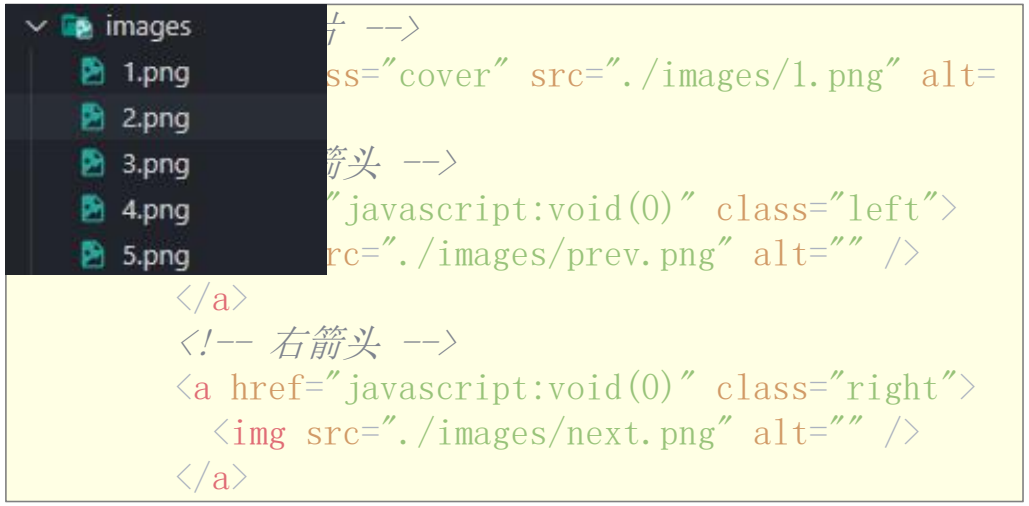
② ：绑定点击事件

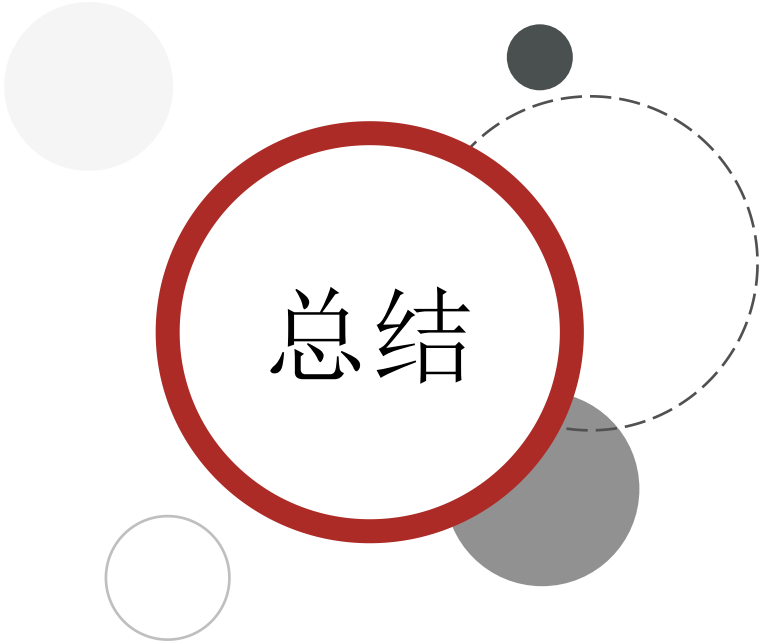
③ ：更改图片显示

src属性，索引

④：更改箭头显示状态

判断 ， display





1. 为transform设置什么属性可以调整元素缩放?

   scale

2. 使用什么方法修改元素属性?

   attr

3. 使用什么方法修改元素样式?

   css

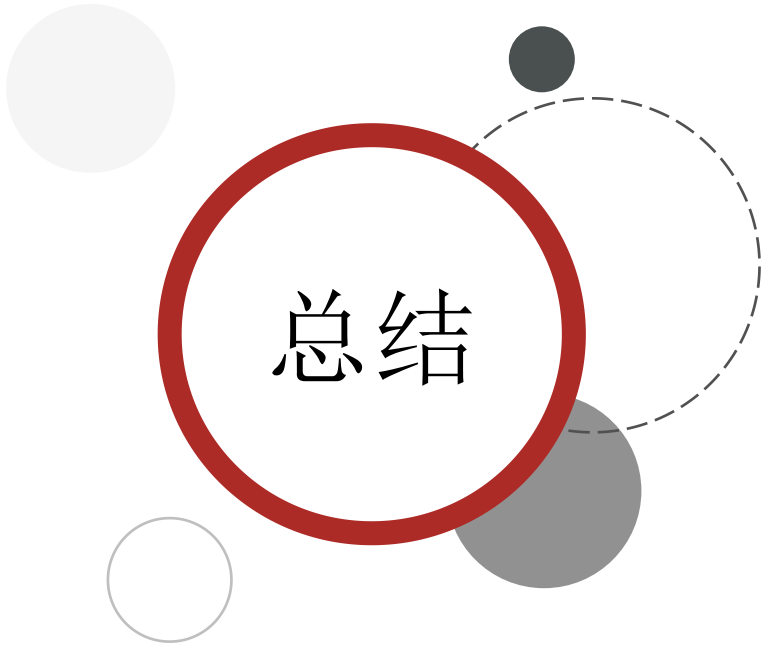
操纵value

操纵value

jQuery 中封装了操纵表单元素value属性的方法，可以取值和赋值

语法

```
// 1. 赋值
.val('参数')
// 2. 取值
.val()
```



1. val方法操纵的是元素的什么属性？

   value属性

2. 获取value是否需要传递参数?

   不需要

15 查找方法

查找方法

jQuery 中封装了查找元素的方法，可以基于元素的结构关系查找新的元素

语法

```
// 1. 父元素
.parent()
// 2. 子元素
.children()
// 3. 兄弟元素
.siblings()
// 4. 后代元素
.find('选择器')
```

- find方法方法需要传入选择器

```
<div class="container">
 <h4>校区列表</h4>
 <ul class="campus">
  <li class="bj">北京校区</li>
  <li class="sh">上海校区</li>
  <li class="gz">广州校区</li>
  <li class="sz">深圳校区</li>
 </ul>
</div>
```

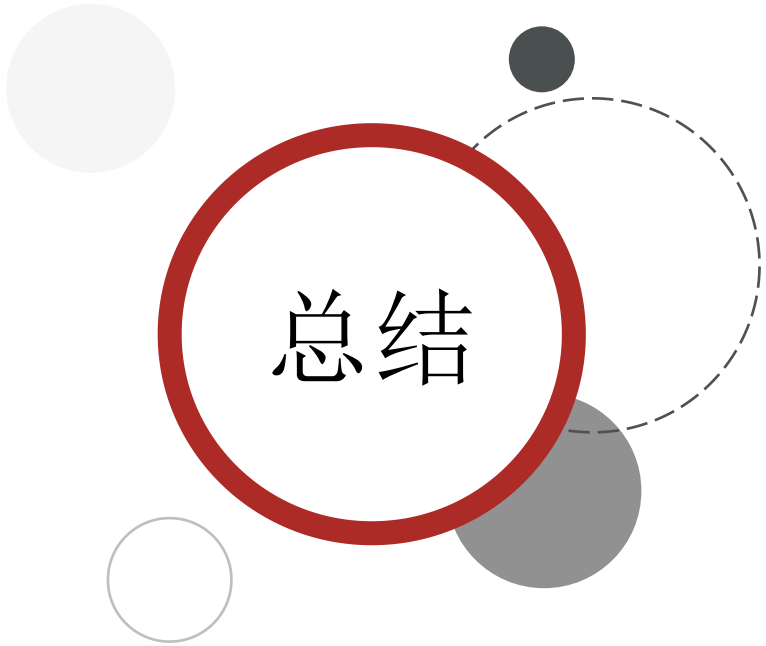
查找方法

jQuery 中封装了查找元素的方法，可以基于元素的结构关系查找新的元素

语法

```
// 1. 父元素
.parent()
// 2. 子元素
.children('选择器')
// 3. 兄弟元素
.siblings('选择器')
// 4. 后代元素
.find('选择器')
```

- find方法方法需要传入选择器
- children 、sibings 方法支持传入选择器



1. parent方法是否需要传递参数?

   不需要

2. 获取子元素的方法是什么?

   children

3. siblings方法一定要传入选择器吗?

   不一定

4. find方法是否可以获取到子元素?

   可以

16 操纵类名

操纵类名

测试代码

```
$('选择器').css({
  backgroundColor: 'orange'
})
```

操纵类名

测试代码

```
$('选择器').css({
  backgroundColor: 'orange',
  color: 'white',
  width: '100px',
  height: '100px',
  border: '1px solid black'
})
```

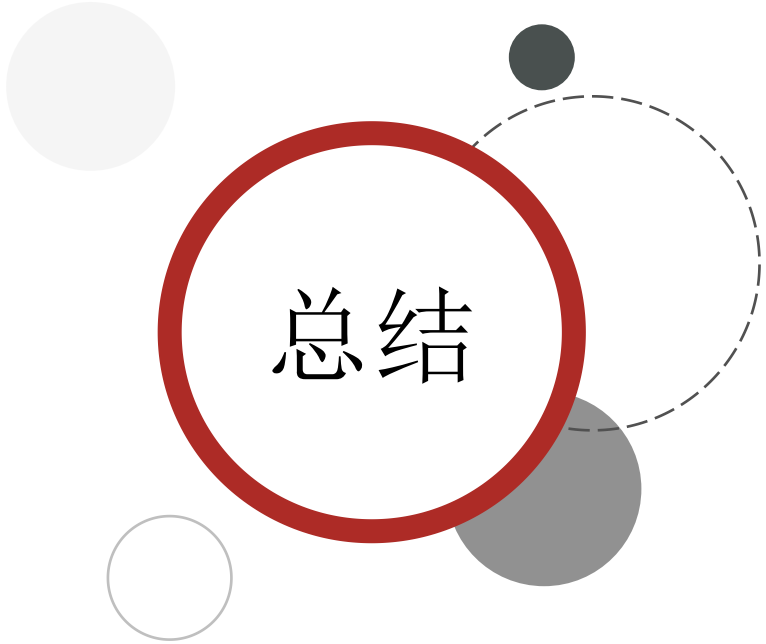
操纵类名

jQuery 中封装了为网页元素添加、移除、检测、切换类名的方法。

语法

```
// 1. 添加类名
.addClass('类名')
// 2. 移除类名
.removeClass('类名')
// 3. 判断类名 返回布尔值
.hasClass('类名')
// 4. 切换类名
.toggleClass('类名')
```

- 参数都是需要操纵的类名



1. 这一节学习的方法是否需要传递参数？

   需要

2. hasClass方法的返回值是什么？

   布尔值：true(存在)、false(不存在)

3. addClass是添加，removeClass是移除，切换类名的方法叫做？

   toggleClass

事件进阶

事件进阶

jQuery 中封装了更为灵活的 on/off、one 方法处理 DOM 事件

测试代码

```
// 绑定点击事件
$('选择器').click(function () {})
// 绑定获得焦点事件
$('选择器').focus(function () {})
```

事件进阶

jQuery 中封装了更为灵活的 on/off、one 方法处理 DOM 事件

测试代码

```
// 可以绑定input事件吗?
$('选择器').input(function () {})
// 解绑事件?
// 一次性事件?
```

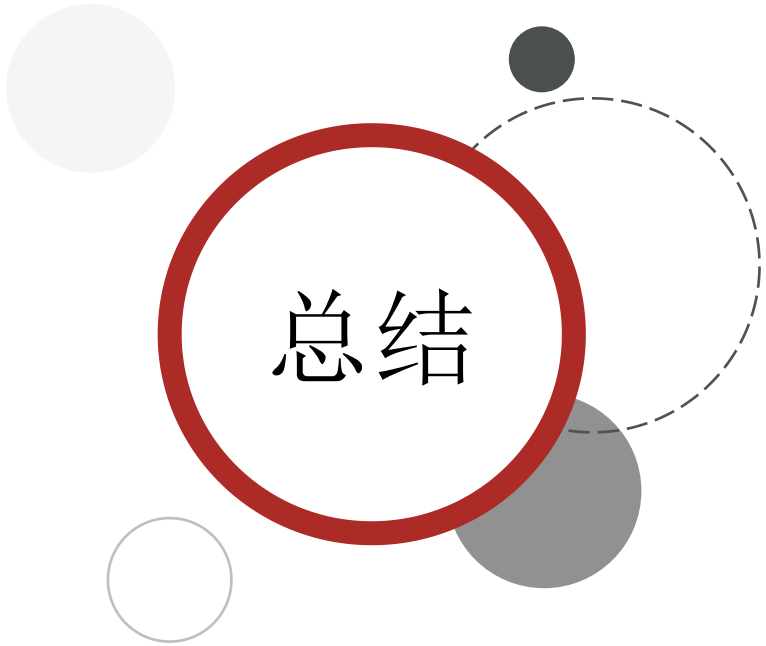
事件进阶

jQuery 中封装了更为灵活的 on/off、one 方法处理 DOM 事件

语法

```
// 1. 注册事件
.on('事件名', function(){})
// 2. 移除指定事件
.off('事件名')
// 3. 移除所有事件
.off()
// 4. 注册一次性事件
.one('事件名', function(){})
```

- on ， one 方法回调函数中的 this 是触发事件的 dom 元素



1. 使用 on 方法进行事件绑定时第一个参数是什么?

   事件名

2. 使用 off 方法是否可以省略参数?

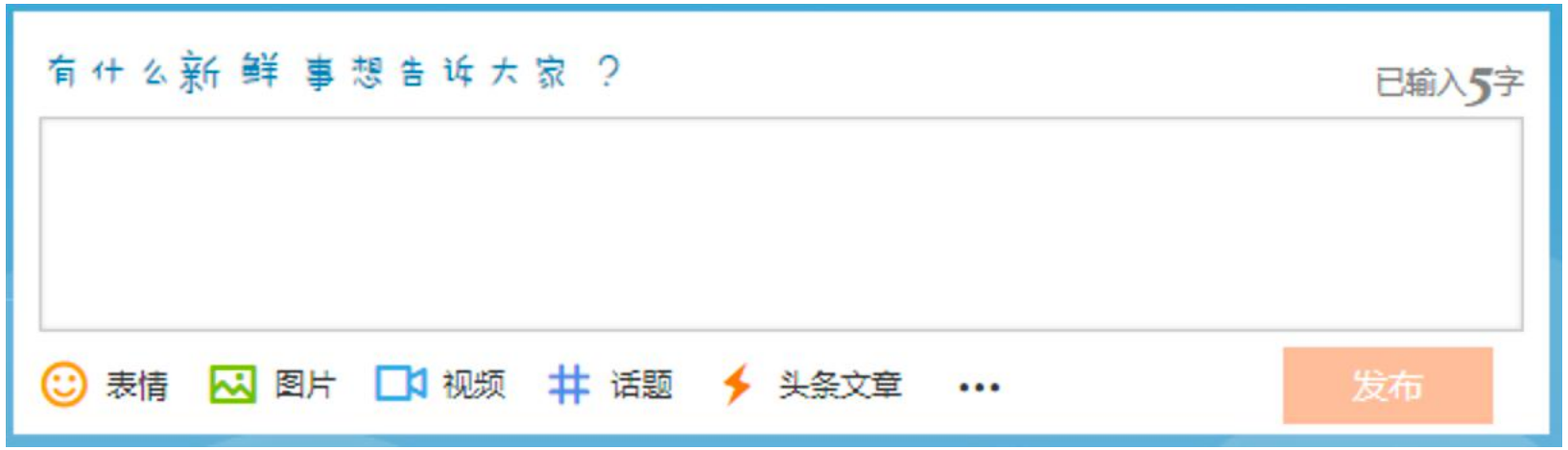
   可以

3. 一次性事件通过什么方法绑定?

   one

18 输入统计

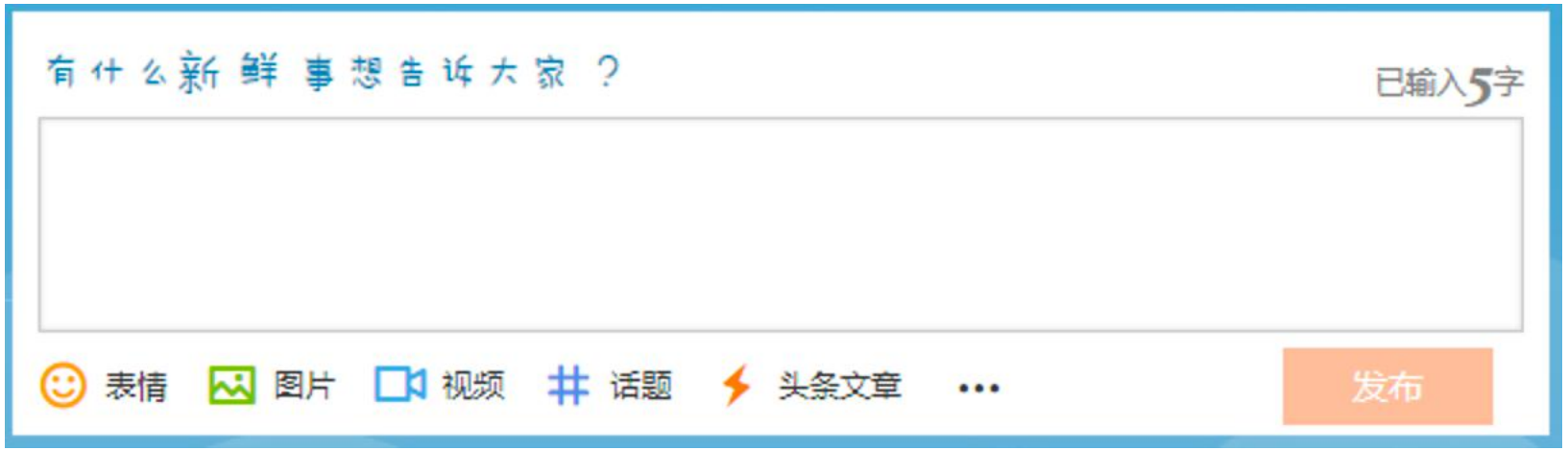
输入统计



输入统计

需求

1. 初始状态
2. 高亮效果
3. 字数统计
4. 启用禁用



步骤

初始状态

需求：默认把统计字数设置为0，发布按钮设置为禁用状态

分析：

①：设置字数为0

text

②：添加禁用类名(disabled)

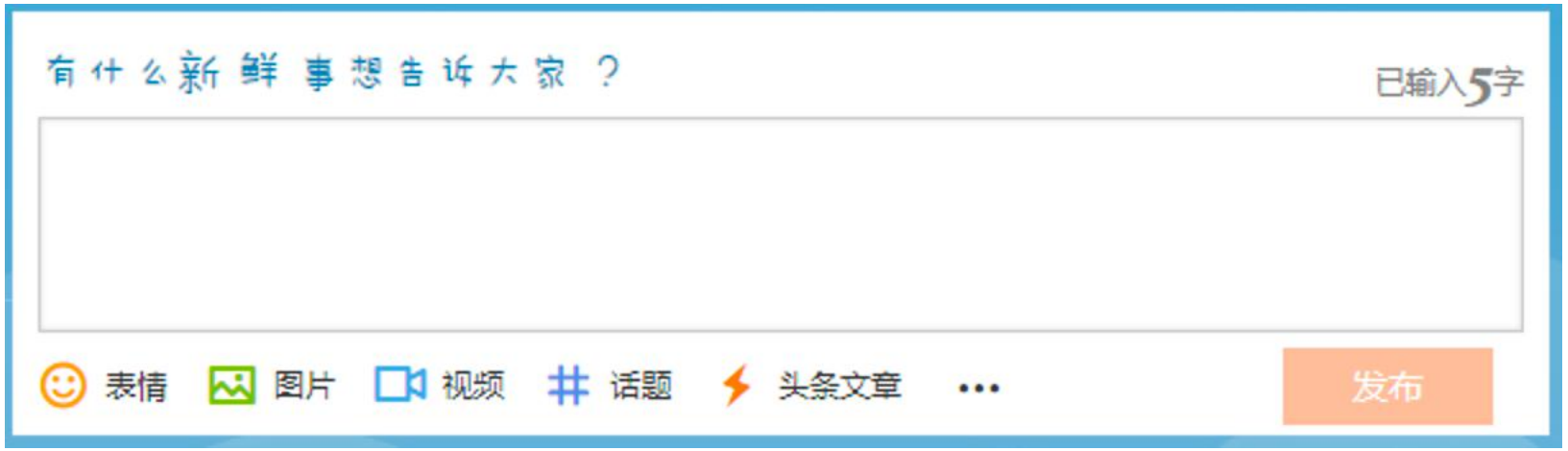
addClass

```
    <!-- 字数统计区域 -->
    <p class="words">
      已输入
      <span>5</span>
      字
    </p>
    <!-- 发布按钮  -->
    <a href="javascript:;" class="publish_btn">发布
</a>
```

输入统计

需求

1. 初始状态
2. 高亮效果
3. 字数统计
4. 启用禁用



步骤

高亮效果

需求：文本域获取焦点时父元素高亮，失去焦点时移除高亮效果

分析：

①：绑定获取焦点和失去焦点事件

on

②：获取到文本域的父元素

parent

③：添加和移除类名(actived)

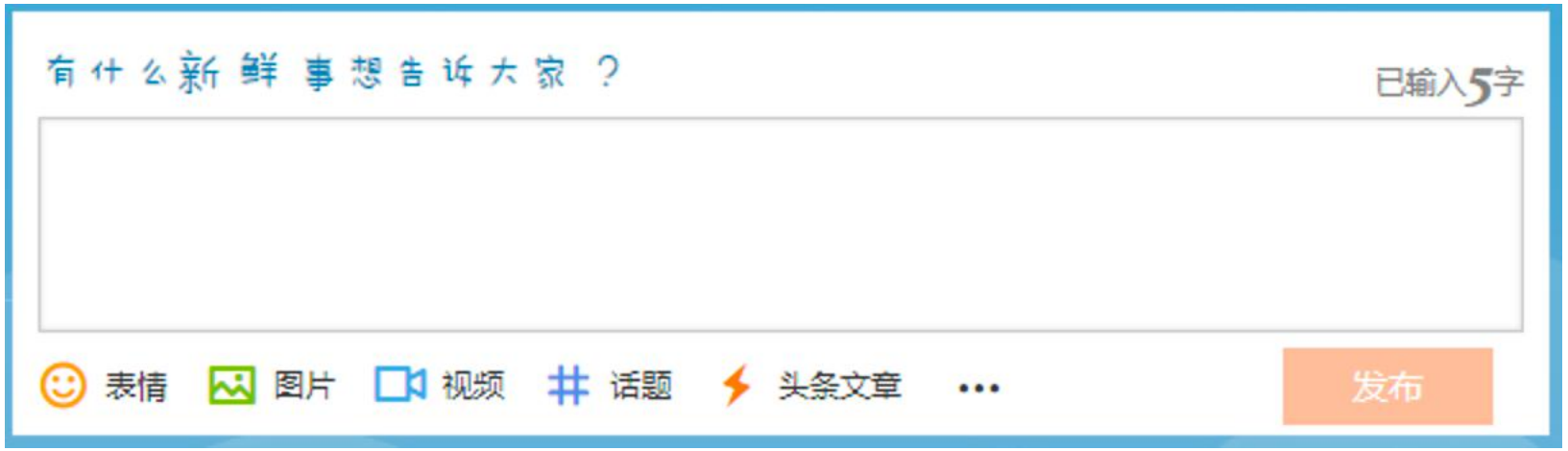
addClass , removeClass

```
<!-- 字数统计区域 -->
<p class="words">
  已输入
  <span>5</span>
  字
</p>
<!-- 输入区域 -->
<div class="input-box actived">
  <textarea></textarea>
</div>
<!-- 发布按钮  -->
<a href="javascript:;" class="publish_btn">发布</a>
```

输入统计

需求

1. 初始状态
2. 高亮效果
3. 字数统计
4. 启用禁用



步骤

字数统计

需求：文本域输入文字时，同步获取输入的文字个数并显示出来

分析：

①：绑定输入事件

on ， input

②：获取文字个数

val ， length

③：渲染页面

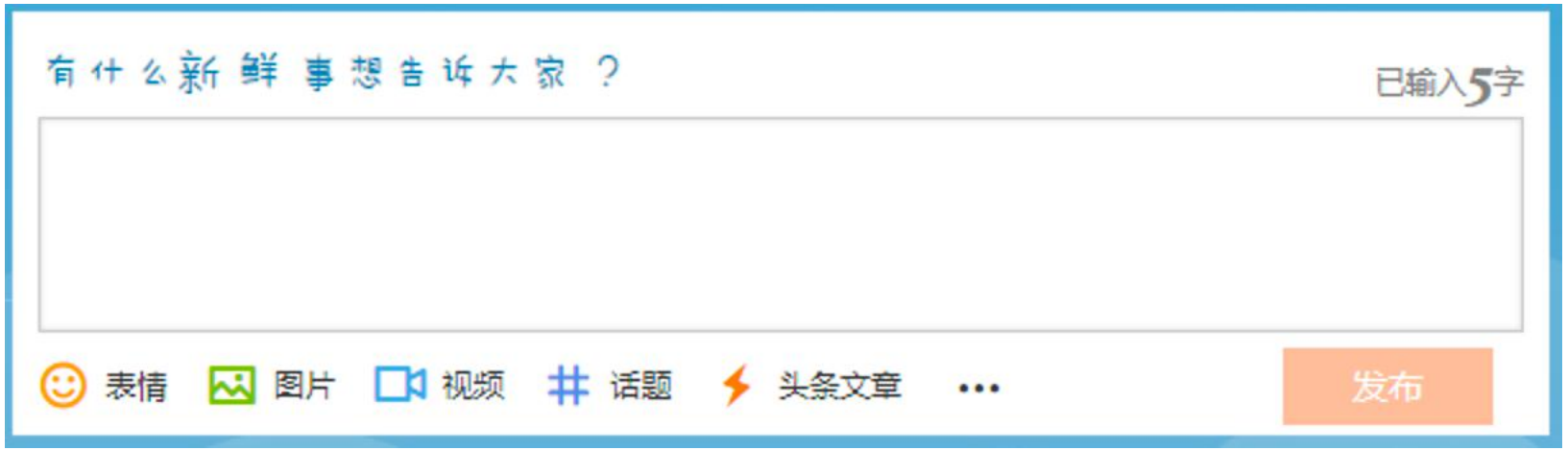
text

```
<!-- 字数统计区域 -->
<p class="words">
  已输入
  <span>5</span>
  字
</p>
<!-- 输入区域 -->
<div class="input-box actived">
  <textarea></textarea>
</div>
<!-- 发布按钮  -->
<a href="javascript:;" class="publish_btn">发布</a>
```

输入统计

需求

1. 初始状态
2. 高亮效果
3. 字数统计
4. 启用禁用



步骤

启用禁用

需求：输入内容为空时禁用发布按钮，不为空时启用发布按钮

分析：

①：绑定输入事件

on ， input

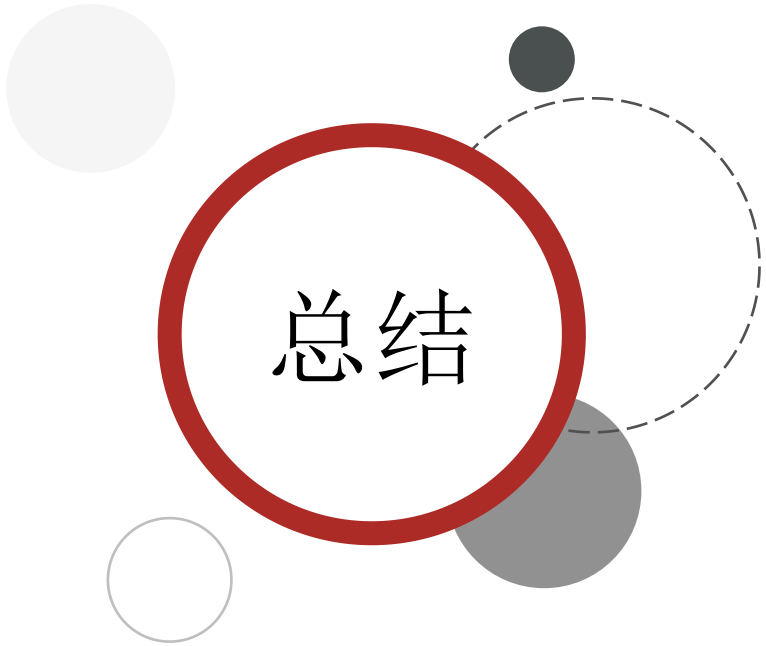
②：判断内容是否为空(长度)

val ， length

③：添加或者移除类名(disabled)

addClass ， removeClass

```
<!-- 字数统计区域 -->
<p class="words">
  已输入
  <span>5</span>
  字
</p>
<!-- 输入区域 -->
<div class="input-box actived">
  <textarea></textarea>
</div>
<!-- 发布按钮  -->
<a href="javascript:;" class="publish_btn disabled">发布
</a>
```



1. focus 和 blur 事件必须通过on方法进行绑定吗?

   不是

2. 哪个方法可以用来获取父元素?

   parent

3. input事件的触发是否需要失去焦点?

   不需要

19 触发事件

触发事件

jQuery 中如何通过代码的方式触发绑定的事件

测试代码

```
// 1. 初始状态
$('.words span').text(0)
$('.publish_btn').addClass('disabled')
```

测试代码

```
// 3. 字数统计
$('.input-box textarea').on('input', function () {
  let length = $(this).val().length
  // 优化
  $('.words span').text(length)
  if (length === 0) {
    $('.publish_btn').addClass('disabled')
  } else {
    $('.publish_btn').removeClass('disabled')
  }
})
```

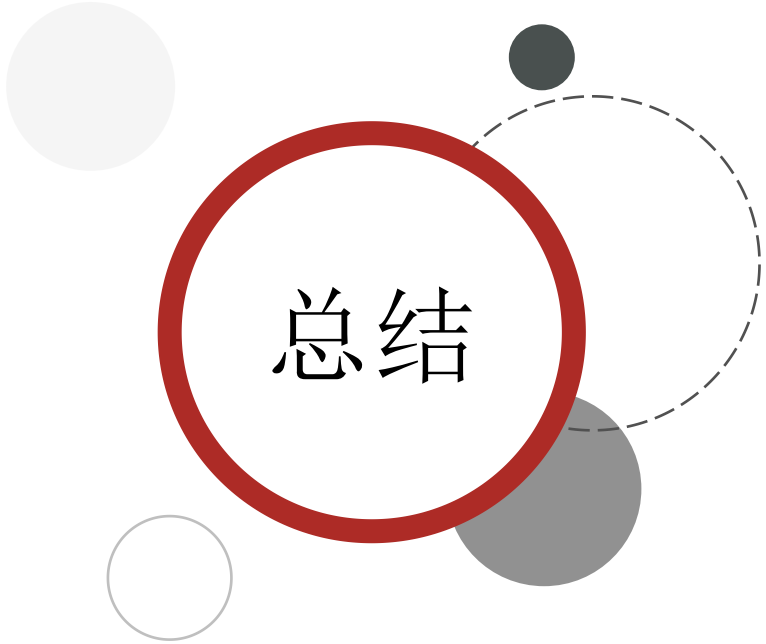
触发事件

jQuery 中如何通过代码的方式触发绑定的事件

语法

```
// 1. 直接触发
.事件名()
// 2. trigger触发
.trigger('事件名')
// 3. 触发自定义事件
.trigger('自定义事件')
// 4. 注册自定义事件
.on('自定义事件',function(){})
```

- 自定义事件是一种进阶用法，目前了解使用方法即可



1. 调用click方法是否可以触发点击事件?

   可以

2. trigger方法只能用来触发原生事件?

   不是

3. 自定义事件是否可以通过鼠标点击来触发?

   不可以

20 登录切换

登录切换

登录切换

需求

1. 账号&安全切换
2. 安全&手机切换

登录切换

步骤

账号&安全切换

需求：点击某种登录方式，切换到对应的登录界面

分析：

①：账号&安全登录点击高亮(active)

click , addClass , siblings , removeCla

②：显示登录界面

attr , siblings

```
<!-- 切换登录场景 -->
<div class="multi-type">
  <a data-label="#account" class="label active">账号登录
</a>
  <a data-label="#secure" class="label">安全登录</a>
  <a data-label="#phone" class="icon"></a>
</div>
<!-- 登录界面 -->
<div class="login-type">
  <!-- 账号登录 -->
  <div class="account" id="account">内容省略</div>
  <!-- 安全登录 -->
  <div class="secure" id="secure">内容省略</div>
  <!-- 手机号登录 -->
  <div class="phone" id="phone">内容省略</div>
</div>
```

登录切换

需求

1. 账号&安全切换(√)
2. 安全&手机切换

登录切换

步骤

安全&手机切换

需求：点击右上角图标，根据当前状态在手机登录和安全登录之间切换

分析：

①：点击判断是否高亮

click , hasClass

②：(未高亮)图标点击高亮(active)

addClass , siblings , removeClass

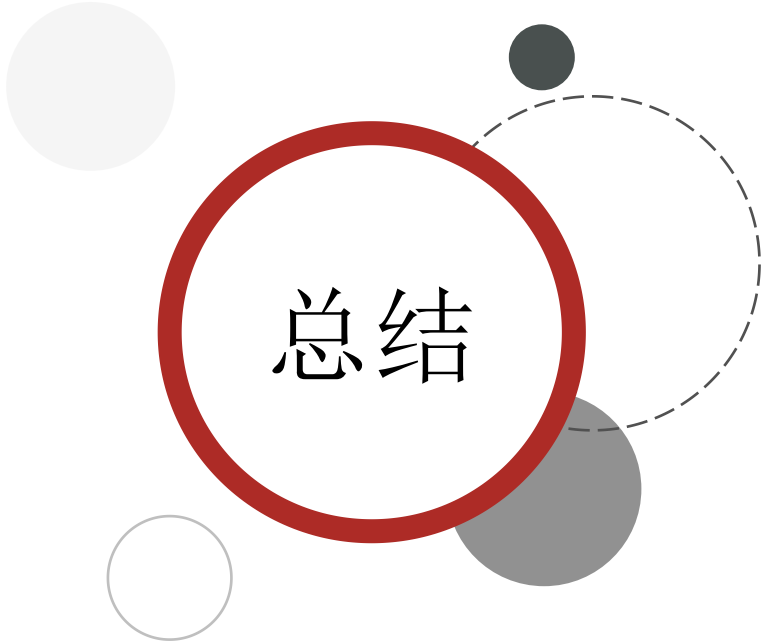
③：(未高亮)显示登录界面

attr , siblings

④：(已高亮)切换到安全登录

trigger

```
<!-- 切换登录场景 -->
<div class="multi-type">
  <a data-label="#account" class="label active">账号登录
</a>
  <a data-label="#secure" class="label">安全登录</a>
  <a data-label="#phone" class="icon"></a>
</div>
<!-- 登录界面 -->
<div class="login-type">
  <!-- 账号登录 -->
  <div class="account" id="account">内容省略</div>
  <!-- 安全登录 -->
  <div class="secure" id="secure">内容省略</div>
  <!-- 手机号登录 -->
  <div class="phone" id="phone">内容省略</div>
</div>
```



1. 自定义属性是否可以通过attr方法获取到?

   可以

2. 判断类名是否存在的方法是?

   hasClass

3. 通过trigger方法触发事件时，参数传递什么？

   事件名

window事件绑定

window事件绑定

使用 jQuery 为window对象绑定事件

测试代码

```
// 滚动
window.onscroll = function () {}
//  点击
window.onclick = function () {}
```

window事件绑定

使用 jQuery 为window对象绑定事件

测试代码

```
// 滚动
$('window').scroll(function () {})
//  点击
$('window').click(function () {})
```

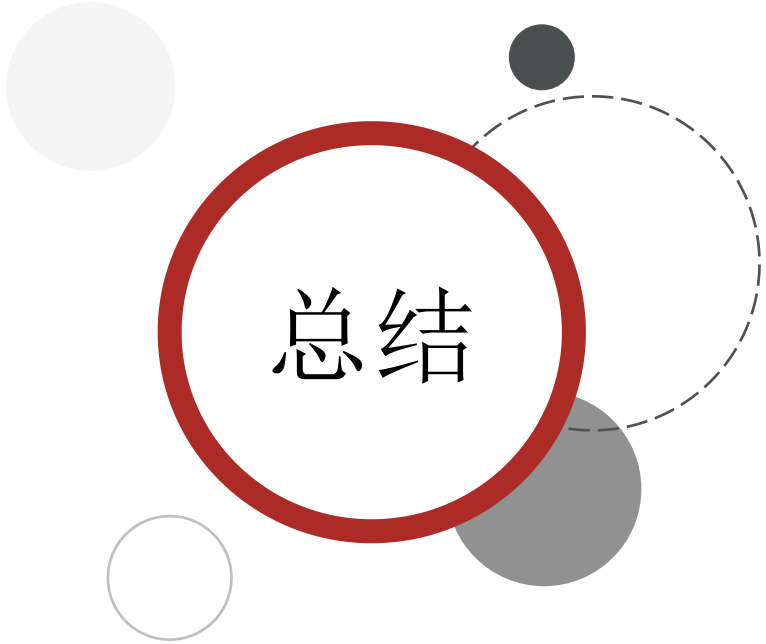
window事件绑定

使用 jQuery 为window对象绑定事件

语法

```
// 滚动
$(window).scroll(function () {})
//  点击
$(window).click(function () {})
```

- 直接传入 window 对象，不需要写成选择器



1. $方法中传入什么可以为window绑定事件?

   window对象

输入标题

输入二级标题，可根据实际情况删除

目录

Contents

- 强调的目录内容
- 根据实际情况调整目录的位置
- 规范使用母版，不可随意调整哦
- 如果内容较少要调整位置
- 不能偏差太大，灵活但要遵守规范
- 尽量不要有回行

学习目标

Learning Objectives

1. 重点文字颜色
2. 此内容上下居中对齐
3. 可根据实际情况微调位置和字体大小
4. 想调整位置也可以用回车试试

一级标题

- 设置二级名称
- 设置二级标题

仅有一级标题

二级标题

- 运算符：对常量或者变量进行操作的符号
- 表达式：用运算符把常量或者变量连接起来符合java语法的式子就可以称为表达式。

  不同运算符连接的表达式体现的是不同类型的表达式。

举例说明

```
int a = 10;
int b = 20;
int c = a + b;
```

+：注释的背景颜色

a + b：代码的背景颜色

一级标题

ElasticSearch是一个搜索服务器

搜索就是查询

select * from xxx → 关系型数据库

表格样式（非母版，使用请复制）

符号	作用	说明
+	加	参看小学一年级
-	减	参看小学一年级
*	乘	参看小学二年级，与“×”相同
/	除	参看小学二年级，与“÷”相同
%	取余	获取的是两个数据做除法的余数

注意事项

/ 和 % 的区别：两个数据做除法，/ 取结果的商，% 取结果的 余数。

整数操作只能得到整数，要想得到小数，必须有浮点数参与运算。

案例

案例标题

需求：键盘录入一个三位数，将其拆分为个位、十位、百位后，打印在控制台

分析：

①：使用Scanner键盘录入一个三位数

②：个位的计算：数值 % 10

123 除以 10（商12，余数为3）

③：十位的计算：数值 / 10 % 10

123 除以 10 （商12，余数为3，整数相除只能得到整数）

12 除以 10 （商1，余数为2）

④：百位的计算：数值 / 10 / 10 % 10

123 / 10 / 10 % 10（123 / 10 得到12，12 / 10 得到1，1 % 10 得到 1）

练习

输入练习的标题

需求：键盘录入一个三位数，将其拆分为个位、十位、百位后，打印在控制台

分析：

①：使用Scanner键盘录入一个三位数

②：个位的计算：数值 % 10

123 除以 10（商12，余数为3）

③：十位的计算：数值 / 10 % 10

123 除以 10 （商12，余数为3，整数相除只能得到整数）

12 除以 10 （商1，余数为2）

④：百位的计算：数值 / 10 / 10 % 10

123 / 10 / 10 % 10（123 / 10 得到12，12 / 10 得到1，1 % 10 得到 1）

步骤

步骤标题

需求：键盘录入一个三位数，将其拆分为个位、十位、百位后，打印在控制台

分析：

①：使用Scanner键盘录入一个三位数

②：个位的计算：数值 % 10

123 除以 10（商12，余数为3）

③：十位的计算：数值 / 10 % 10

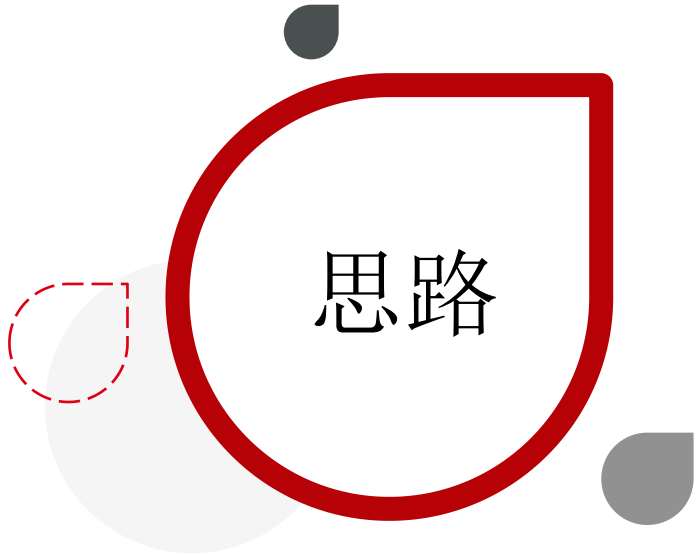
123 除以 10 （商12，余数为3，整数相除只能得到整数）

12 除以 10 （商1，余数为2）

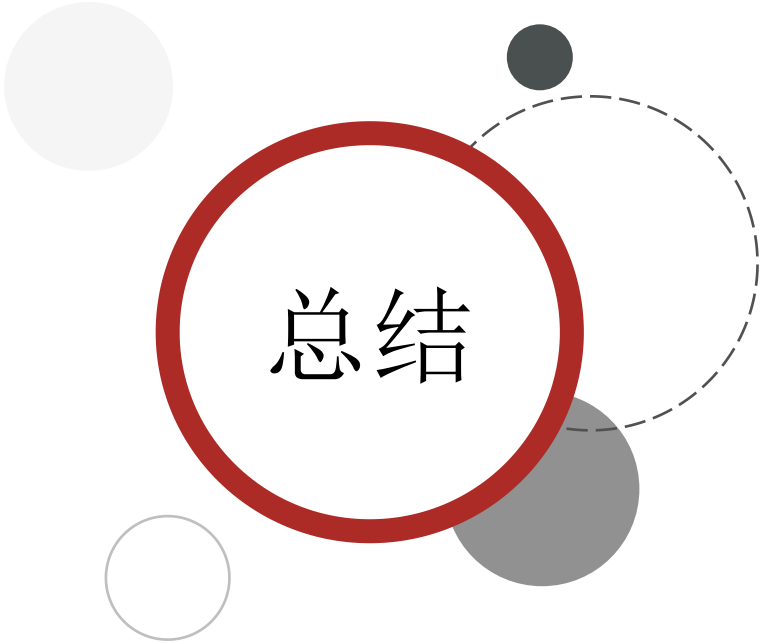
④：百位的计算：数值 / 10 / 10 % 10

123 / 10 / 10 % 10（123 / 10 得到12，12 / 10 得到1，1 % 10 得到 1）

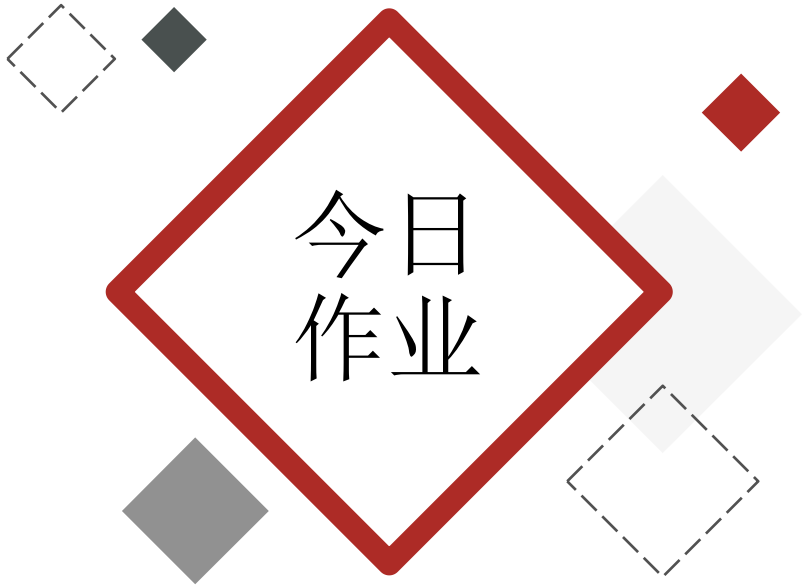
1. 文本居中对齐
2. 根据实际情况调整文本位置，红色色值
3. 力求美观、简洁、大方



1. 文本居中对齐
2. 根据实际情况调整文本位置，红色色值
3. 力求美观、简洁、大方



1. 文本居中对齐
2. 根据实际情况调整文本位置，红色色值
3. 力求美观、简洁、大方



1. 文本居中对齐
2. 根据实际情况调整文本位置，红色色值
3. 力求美观、简洁、大方

代码的样式

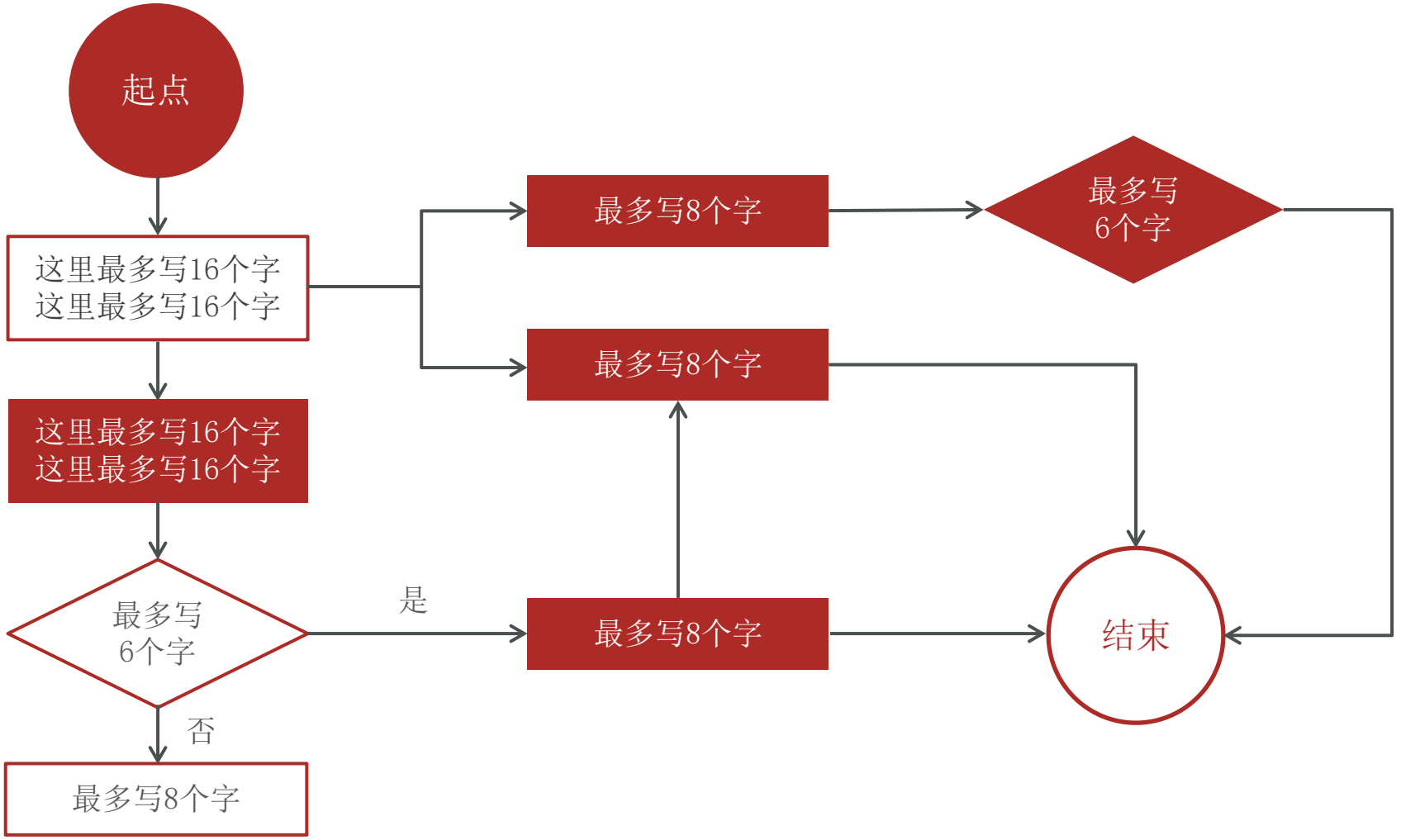
代码样式（非母版，需要请复制，如有红色请用色值#AD2B26）

```
public class Test {
    public static void main(String[] args) {
        int a = 10;
        int b = 20;
        System.out.println(a > b || a < b && a > b);
    }
}
```

流程图使用规范（非母版，请复制使用，规范需删除）

流程图样式

使用规范：
1、只有90度横向和竖向，没有45度的走向。
2、起点和结束为固定样式。
3、流程中重点用实色块，非重点用线框。
4、字数多和字数少的矩形高度不同，做了区分。



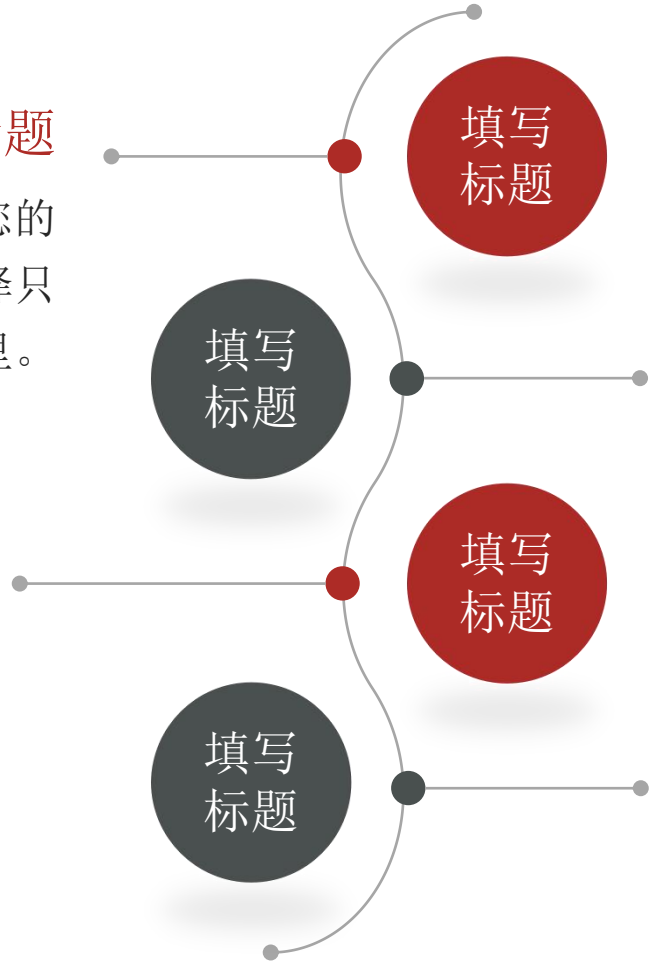
其他的版式样式（此版式有动画）

填写段落标题

您的内容打在这里，或者通过复制您的文本后，在此框中选择粘贴，并选择只保留文字。您的内容打在这里。

填写段落标题

您的内容打在这里，或者通过复制您的文本后，在此框中选择粘贴，并选择只保留文字。您的内容打在这里。



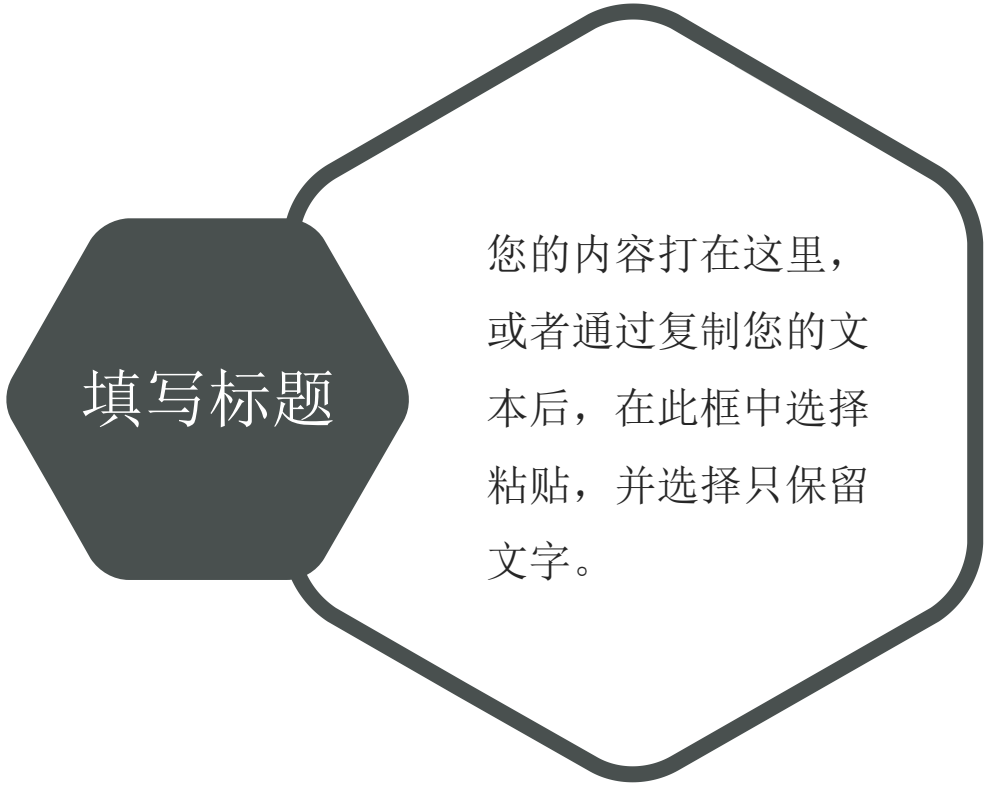
填写段落标题

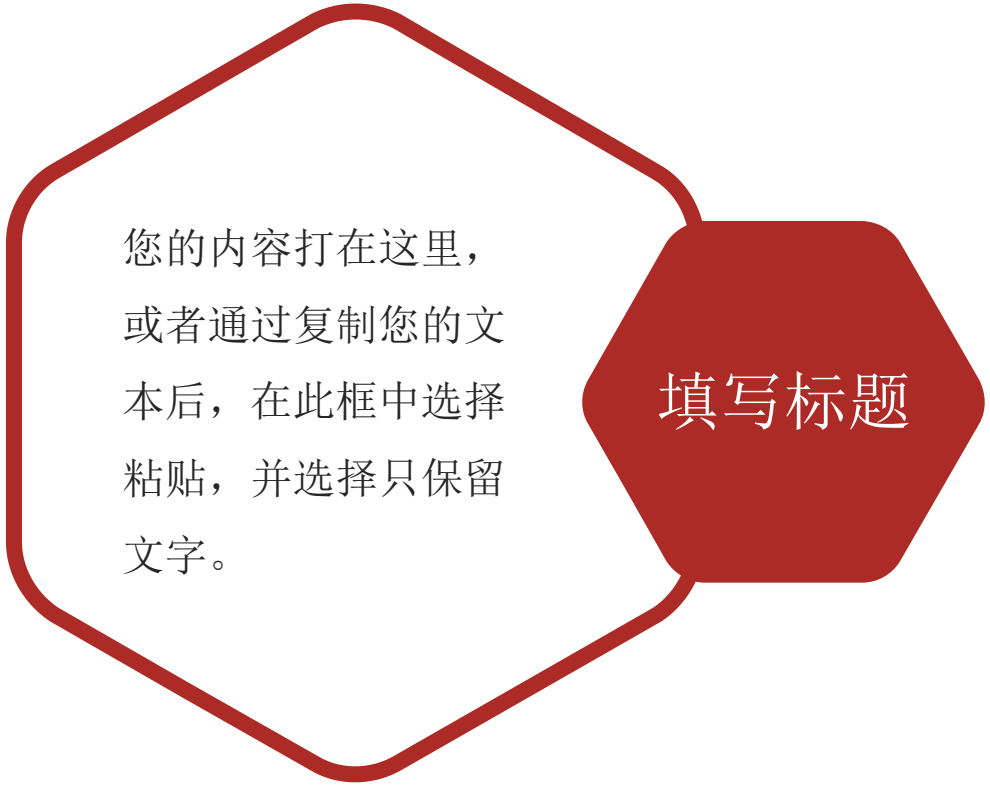
您的内容打在这里，或者通过复制您的文本后，在此框中选择粘贴，并选择只保留文字。您的内容打在这里。

填写段落标题

您的内容打在这里，或者通过复制您的文本后，在此框中选择粘贴，并选择只保留文字。您的内容打在这里。

其他的版式样式


填写标题
您的内容打在这里，或者通过复制您的文本后，在此框中选择粘贴，并选择只保留文字。


您的内容打在这里，或者通过复制您的文本后，在此框中选择粘贴，并选择只保留文字。
填写标题

其他的版式样式

填写段落标题

您的内容打在这里，或者通过复制您的文本后，在此框中选择粘贴，并选择只保留文字。

填写段落标题

您的内容打在这里，或者通过复制您的文本后，在此框中选择粘贴，并选择只保留文字。

填写段落标题

您的内容打在这里，或者通过复制您的文本后，在此框中选择粘贴，并选择只保留文字。

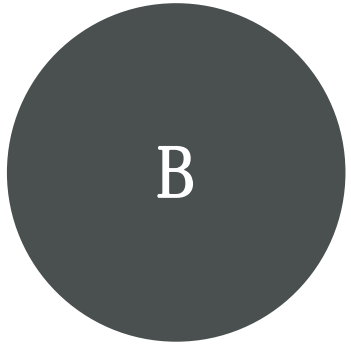
填写段落标题

您的内容打在这里，或者通过复制您的文本后，在此框中选择粘贴，并选择只保留文字。

其他的版式样式

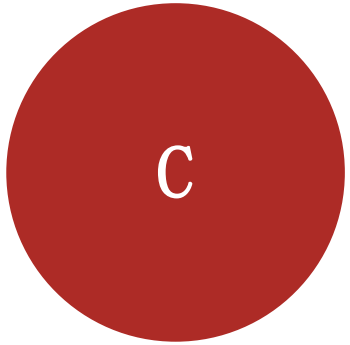
填写段落标题

您的内容打在这里，或者通过复制您的文本后，在此框中选择粘贴，并选择只保留文字。



填写段落标题

您的内容打在这里，或者通过复制您的文本后，在此框中选择粘贴，并选择只保留文字。



填写段落标题

您的内容打在这里，或者通过复制您的文本后，在此框中选择粘贴，并选择只保留文字。

规范使用说明

1. 字体为 阿里巴巴普惠体
2. 字号 正文一级标题 24号；正文二级标题 18号；正文文字 16号；代码字号14号；流程图字号14号；注释、注意事项字号14号。字号可根据实际内容的多少进行微调，原则上按规范使用。
3. 标准色及色值

红色色值 AD2B26　　黑色色值 49504F

传智教育旗下高端IT教育品牌