



jQuery

day03



课程介绍

学习目标

Learning Objectives

1. 能够了解常见 `jQuery` 插件的使用套路
2. 能够了解插件的本质及实现方法
3. 能够掌握操纵表单的方法

人员管理

新增

姓名	手机	生日	操作	
小红	18888881111	1990年02月08日	编辑	删除
小黄	18888882222	1990年02月09日	编辑	删除
小蓝	18888883333	1990年02月10日	编辑	删除
小绿	18888884444	1990年02月11日	编辑	删除
小青	18888885555	1990年02月12日	编辑	删除



总结

1. 插件 + 表单 + 3个案例



入口函数

jQuery 中提供了更为简便的入口函数写法

测试代码

```
// 原生写法  
window.onload = function () {}  
// jq写法  
$(window).on('load', function () {})
```

- 页面资源加载完毕执行(包括图片、css等等)逻辑代码

jQuery 中提供了更为简便的入口函数写法

语法

```
// 完整写法  
$(document).ready(function () {})  
// 简化写法  
$(function () {})
```

- DOM载入完毕就会执行



总结

1. `window.onload` 和 `ready`哪个先执行?
`ready`
2. `window.onload` 和 `ready`哪个可以获取图片尺寸?
`window.onload`
3. 开发中用哪个?
具体情况具体分析

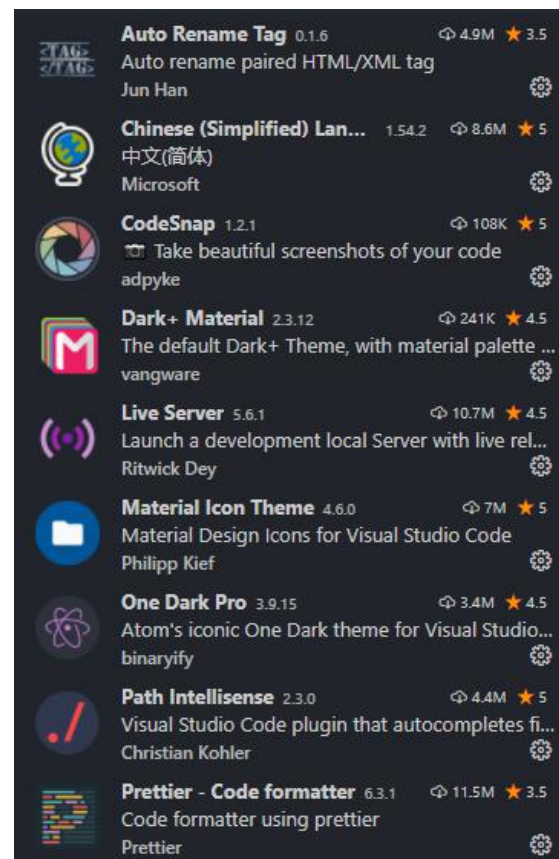
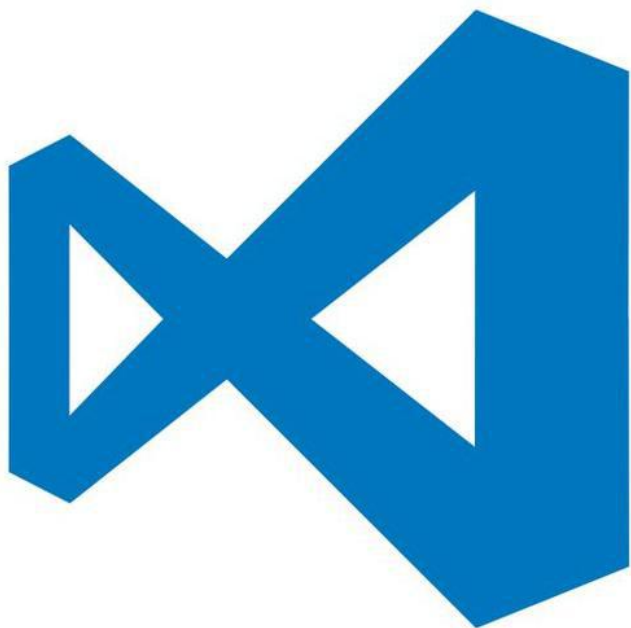


轮播图



轮播图插件

插件：为程序附加功能



jQuery 的轮播图插件 slick

- 下包：把插件下载到本地
- 导包：先导入 jQuery ，再导入插件 ，导入css(需要的话)
- 用包：根据文档(说明书)使用

导包

语法

```
<!-- 引入插件所需要的样式表 -->
<link rel="stylesheet" href="./assets/slick/slick.css">
<link rel="stylesheet" href="./assets/slick/slick-theme.css">

<!-- 先导入 jQuery -->
<script src="./assets/jquery/jquery-3.5.1.js"></script>
<!-- 在 jQuery 之后引入插件 -->
<script src="./assets/slick/slick.min.js"></script>
```

用包

语法

```
<div class="your-class">  
  <div>轮播元素一</div>  
  <div>轮播元素二</div>  
  <div>轮播元素三</div>  
</div>
```

语法

```
$('.your-class').slick()
```

用包

常用配置	含义	默认值	备注
autoplay	轮播图效果自动执行	false	true/false
arrows	是否显示翻页按钮	true	true/false
prevArrow	自定义 上一页 按钮	无	标签选择器
nextArrow	自定义 下一页 按钮	无	标签选择器
dots	是否显示指示器	false	true/false

语法

```
// 调用插件方法初始化
$('.your-class').slick({
  autoplay:true,
  arrows:true
  ...
})
```


用包

常用配置	含义	默认值	备注
autoplay	轮播图效果自动执行	false	true/false
arrows	是否显示翻页按钮	true	true/false
prevArrow	自定义 上一页 按钮	无	标签选择器
nextArrow	自定义 下一页 按钮	无	标签选择器
dots	是否显示指示器	false	true/false

- 默认样式不符合需求直接添加 **CSS** 覆盖即可
- 更多的配置参数可以查阅官方文档:<https://github.com/kenwheeler/slick>



总结

1. slick.js文件导入的位置是 **jQuery** 之前还是之后
?

之后

2. 是否需要把使用步骤和配置背下来?
不需要



懒加载插件

jQuery 的懒加载插件 lazyload

- 懒加载：图片用到了再去加载，常见于有大量图片的网页，比如电商



jQuery 的懒加载插件 lazyload

- 下包：把插件下载到本地
- 导包：先导入 jQuery ，再导入插件 ，导入css(需要的话)
- 用包：根据文档(说明书)使用

导包

语法

```
<!-- 先导入jQuery -->  
<script src="../jquery/jquery-3.5.1.js"></script>  
<!-- 再导入lazyload插件 -->  
<script src="../assets/jquery.lazyload.min.js"></script>
```

用包

语法

```
<!-- 图片 -->  

```

用包

语法

```
<!-- 图片 -->  
<img class="lazyload" data-  
original="./images/1.png" alt="" />
```

- 图片地址设置给 `data-original`

用包

语法

```
<!-- 图片 -->  
<img class="lazyload" data-  
original="./images/1.png" alt="" />
```

语法

```
// 找到希望懒加载的图片并调用lazyload方法  
$('.lazyload').lazyload()
```

- 官方文档:<https://github.com/tuupola/lazyload/tree/1.x>



总结

1. 使用懒加载插件之后，看不到的图片会不会加载？

不会

2. 图片较多的网页可以用这个提升加载速度吗？

可以



小兔鲜儿



需求

1. 轮播图
2. 图片懒加载



步骤

轮播图

需求：项目中整合轮播图效果，使用自定义的左右箭头，开启自动轮播，显示指示点并且居中

分析：

①：整合轮播图插件(slick)

下包，导包

②：调用插件方法

slick方法

③：调整配置或样式

随用随查，审查元素

```
<div class="xtx_banner">
  <ul>
    <li>
      <a href="javascript:;">
        
      </a>
    </li>
    ... 略
  </ul>
  <!-- 切换按钮 -->
  <a href="javascript:;" class="prev sprites">
</a>
  <a href="javascript:;" class="next sprites">
</a>
  <!-- 指示器 -->
  <div class="indicator"></div>
</div>
```

需求

1. 轮播图
2. 图片懒加载



步骤

图片懒加载

需求：轮播图下方的图片全部通过懒加载的方式载入

分析：

①：整合懒加载插件(lazyload)

下包，导包

②：调整图片地址

`data-original`

③：调用插件方法

`lazyload`

```

```



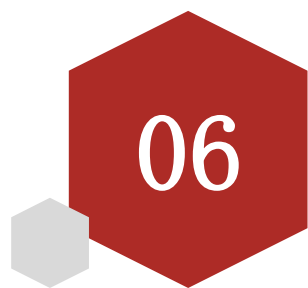

总结

1. 工作中所有的功能都需要自己手写吗?

不需要

2. 插件的使用方法，配置需要背下来吗?

不需要



全屏滚动

jQuery 的全屏滚动插件 fullpage



360安全浏览器

4亿网友共同信赖

更安全，更爽快，更智能，更时尚

jQuery 的全屏滚动插件 fullpage

- 下包：把插件下载到本地
- 导包：先导入 jQuery ，再导入插件 ，导入css(需要的话)
- 用包：根据文档(说明书)使用

导包

语法

```
<!-- 导入 fullpage 配套样式 -->  
<link rel="stylesheet" href="./fullpage/jquery.fullpage.css" />  
  
<!-- 导入 jQuery -->  
<script src="./jquery/jquery-3.5.1.js"></script>  
<!-- 在 jQuery 之后导入 fullpage 插件 -->  
<script src="./fullpage/jquery.fullpage.js"></script>
```

用包

语法

```
<div id="fullpage">  
  <div class="section">第一屏</div>  
  <div class="section">第二屏</div>  
  <div class="section">第三屏</div>  
</div>
```

- 每个区域必须有 `section` 这个类名

用包

常用配置	含义	默认值	备注
navigation	是否显示导航	false	true/false
navigationPosition	导航位置	right	left/right
anchors	每个区域的锚链接名	[]	在地址栏
afterLoad	区域加载完毕的回调函数，有2个参数	无	参数：锚链接、索引

语法

```
$('#fullpage').fullpage({  
  ....  
})
```

- 更多的配置参数可以查阅官方文档：<https://github.com/alvarotrigo/fullPage.js/tree/2.9.7>



总结

1. `fullpage`插件在使用的时候一定要传入配置吗?

不需要

2. `fullpage`插件的区域容器必须要有什么类名?

`section`



安全导航



360安全浏览器

4亿网友共同信赖

更安全，更爽快，更智能，更时尚

需求

1. 全屏滚动



步骤

全屏滚动

需求：整合全屏滚动效果，每次切屏完毕播放动画，支持来回滚动

分析：

①：整合fullpage插件

下包，导包

②：调用方法整合全屏滚动

fullpage

③：切屏完毕播放动画

afterLoad, current类, addClass

④：来回滚动

current类, removeClass

```
<!-- 外层容器 -->
<div id="fullpage" class="fullpage">
  <div class="section current"></div>
  <div class="section"></div>
  <div class="section"></div>
  <div class="section"></div>
  <div class="section"></div>
</div>
```



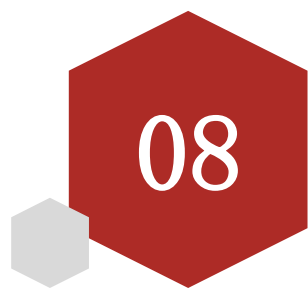
总结

1. 使用的回调函数叫什么名字?

`afterLoad`

2. 回调函数的第二个参数是什么?

索引



提交事件

form 标签本身具有提交数据的能力，但是现在基本不这么用

测试结构

```
<form>
  <input type="text" placeholder="用户名" />
  <br />
  <input type="password" placeholder="密码" />
  <br />
  <input type="submit" value="提交" />
  <button>提交</button>
</form>
```

- 点击提交按钮，输入区域点击回车都会触发表单提交
- 表单中的 **button** 默认就是提交按钮

现在比较流行在表单的 `submit` 事件中阻止默认行为，自己获取数据并提交

测试结构

```
$('form').submit(function (event) {  
    // 阻止默认行为  
    event.preventDefault()  
    // 阻止默认行为  
    return false  
})
```

- 返回false 或者 preventDefault 都可以阻止默认行为



总结

1. 点击提交按钮或者在输入区域按回车会触发表单的什么事件？

`submit`

2. 事件中`return`什么值可以阻止默认行为？

`false`



日期选择器

jQuery 的日期选择器插件 **datepicker**，让用户在不同的浏览器下可以用一致的方式来选择日期

- 下包：把插件下载到本地
- 导包：先导入 **jQuery**，再导入**插件**，导入**css**(需要的话)
- 用包：根据文档(说明书)使用



导包

语法

```
<!-- 导入日期选择器的样式 -->  
<link rel="stylesheet" href="./datepicker/datepicker.css" />  
  
<!-- 导入jQuery -->  
<script src="./jquery/jquery-3.5.1.js"></script>  
<!-- 导入日期选择器插件 -->  
<script src="./datepicker/datepicker.js"></script>  
<!-- 导入语言包 -->  
<script src="./datepicker/il8n/datepicker.zh-CN.js"></script>
```

- 默认显示的是英文，需要导入 中文语言包

用包

常用配置	含义	默认值	备注
autoPick	是否自动选择当前日期	false	true/false
autoHide	选择日期之后是否自动关闭	false	true/false
language	语言模式	空	需要结合语言包使用

语法

```
<!-- 准备html结构 -->
<input type="text" class="datepicker" />

// 调用插件方法
$('.datepicker').datepicker({
  ...
})
```

- 更多的配置参数可以查阅官方文档:<https://github.com/fengyuanchen/datepicker/blob/master/README.md>



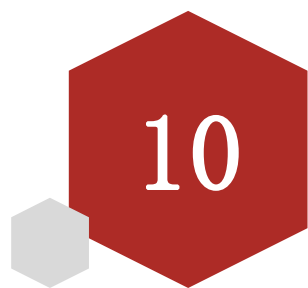
总结

1. 插件默认显示的语言是中文还是英文?

英文

2. 为了设置插件的语言为中文，需要导入什么?

中文语言包



表单验证

jQuery 的表单验证插件 `validate`，验证用户在表单中输入的内容

- 下包：把插件下载到本地
- 导包：先导入 `jQuery`，再导入 `插件`，导入 `css` (需要的话)
- 用包：根据文档(说明书)使用

10-表单验证插件

基于 jQuery 的表单验证插件，验证用户在表单中填写的内容。

昵称: 昵称不能为空!

密码: 密码只能为数字和字母且不能少于6位!

☐ 同意注册协议 必须同意该协议!

提交

导包

语法

```
<!-- 导入jQuery -->  
<script src="../jquery/jquery-3.5.1.js"></script>  
<!-- 导入validate插件 -->  
<script src="../jquery-validate/jquery-validate.js"></script>
```

用包

常用配置	含义	默认值	备注
onBlur	失去焦点时验证	false	true/false
onSubmit	提交表单时验证	true	true/false
sendForm	是否提交表单	true	true/false
valid	所有表单项验证通过执行的 回调函数	无	this是jq对象
invalid	至少一个表单项为通过验证时执行的 回调函数	无	this是jq对象
description	错误提示信息	无	Object

语法

```
$('#form').validate({  
  // 配置  
})
```

用包

自定义属性	含义	备注
data-required	验证表单项不能为空	不需要值
data-pattern	基于正则表达式验证	正则表达式

语法

```
<input
  type="password"
  name="password"
  data-required
  data-pattern=". {6,}"
/>
```

- 表单元素要放在 `form` 里面

用包

自定义属性	含义	备注
data-describedby	指定显示错误信息的标签	标签的id
data-description	指定错误信息的内容	和description中的属性对应

- 这2个属性需要结合 `description` 配置使用

语法

```
<input
  type="password"
  name="password"
  data-required
  data-describedby="password-error"
  data-description="password"
  data-pattern=". {6,}"
/>
<span class="error" id="password-
error"></span>
```

语法

```
$('#form').validate({
  description: {
    password: {
      required: '密码不能为空!',
      pattern: '密码不能少于6位!',
    },
  },
},
```



总结

1. 这一节学习的插件是用来验证什么的?
验证表单
2. 案例中校验密码长度用的是什么技术?
正则表达式



克隆

jQuery 中封装了克隆(复制)，节点的方法

语法

```
// 不带事件  
.clone()  
// 带事件  
.clone(true)
```

- 方法返回的还是 jQuery 对象
- 传入 true 事件也会一起克隆



总结

1. 克隆方法得到的是什么对象？

jQuery对象

2. 希望事件一起克隆需要传递什么参数？

true



获取dom对象

jQuery 封装了获取内部 dom 对象的方法

测试代码

```
// 播放视频  
video标签.play()  
// 暂停视频  
video标签.pause()  
// 重置表单  
form表单.reset()
```

- jQuery 并没有封装所有 dom 的方法

jQuery 封装了获取内部 dom 对象的方法

测试代码

```
// 播放视频  
$('video').trigger('play')  
// 暂停视频  
$('video').trigger('pause')  
// 重置表单  
$('form').trigger('reset')
```

- jQuery 通过trigger可以触发这些方法

jQuery 封装了获取内部 dom 对象的方法

语法

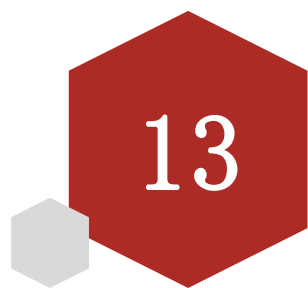
```
// get方法获取  
.get(索引)  
// 中括号获取  
[索引]
```

- 索引从 0 开始
- 获取到的是 dom 对象



总结

1. 可以使用 `[]` 还有一个什么方法来获取dom对象?
`get`方法
2. 索引从几开始?
从0开始



人员管理

人员管理

[新增](#)

姓名	手机	生日	操作	
小黄	18888882222	1990年02月09日	编辑	删除
小绿	18888884444	1990年02月11日	编辑	删除
小青	18888885555	1990年02月12日	编辑	删除

需求

- 新增
- 删除
- 编辑

人员管理			
新增			
姓名	手机	生日	操作
小黄	18888882222	1990年02月09日	编辑 删除
小绿	18888884444	1990年02月11日	编辑 删除
小青	18888885555	1990年02月12日	编辑 删除

需求

- 新增
 - 弹框及插件整合
 - 添加及关闭弹框
- 删除
- 编辑

人员管理			
新增	姓名	手机	生日
	小黄	18888882222	1990年02月09日
	小绿	18888884444	1990年02月11日
	小青	18888885555	1990年02月12日

步骤

新增 - 弹框及插件整合

需求：点击弹出新增框，整合日期及表单校验插件

分析：

①：点击新增按钮弹框带动画

`click , fadeIn`

②：整合日期插件

文档

③：整合表单校验插件

文档，非空，正则

`^(13[0-9]|14[01456879]|15[0-35-9]|16[2567]|17[0-8]|18[0-9]|19[0-35-9])\d{8}$`



需求

- 新增
 - 弹框及插件整合
 - 添加及关闭弹框
- 删除
- 编辑

人员管理			
新增	姓名	手机	生日
	小黄	18888882222	1990年02月09日
	小绿	18888884444	1990年02月11日
	小青	18888885555	1990年02月12日

步骤

新增 - 添加及关闭弹框

需求：点击提交新增数据到页面上，关闭弹框之后清空内容

分析：

①：关闭弹框之后，清空内容

`fadeOut` , `reset` , 抽取

②：表单校验通过之后

`valid`

③：新增数据，并且设置内容

克隆 , `val` , `append`

```
<tr>
  <td class="td-name">小红</td>
  <td class="td-mobile">18888881111</td>
  <td class="td-birth">1990年02月08日</td>
  <td>
    <button class="btn edit">编辑</button>
    <button class="btn del">删除</button>
  </td>
</tr>
```

需求

- 新增
 - 弹框及插件整合
 - 添加及关闭弹框
- 删除
- 编辑

人员管理			
新增	姓名	手机	生日
			操作
	小黄	18888882222	1990年02月09日
			编辑 删除
	小绿	18888884444	1990年02月11日
			编辑 删除
	小青	18888885555	1990年02月12日
			编辑 删除

步骤

删除

需求：点击删除，直接删除整行

分析：

①：删除按钮事件绑定

委托，click

②：tr播放淡出动画

parent，fadeOut

③：动画播放完毕，删除tr

回调函数，remove

④：tbody中没有tr之后，有坑!!

```
<tr>
  <td class="td-name">小红</td>
  <td class="td-mobile">18888881111</td>
  <td class="td-birth">1990年02月08日</td>
  <td>
    <button class="btn edit">编辑</button>
    <button class="btn del">删除</button>
  </td>
</tr>
```

需求

- 新增
 - 弹框及插件整合
 - 添加及关闭弹框
- 删除
- 编辑

人员管理			
新增	姓名	手机	生日
	小黄	18888882222	1990年02月09日
	小绿	18888884444	1990年02月11日
	小青	18888885555	1990年02月12日

需求

- 新增
 - 弹框及插件整合
 - 添加及关闭弹框
- 删除
- 编辑
 - 进入编辑状态
 - 保存此次修改

人员管理			
新增	姓名	手机	生日
	小黄	18888882222	1990年02月09日
	小绿	18888884444	1990年02月11日
	小青	18888885555	1990年02月12日

步骤

编辑 - 进入编辑状态

需求：点击提交，修改数据并关闭弹框

分析：

①：编辑按钮事件绑定

委托，click

②：获取这一行的内容

parent，find

③：修改弹框标题和内容，并显示

fadeIn，val

④：新增功能受影响了没？

```
<tr>
  <td class="td-name">小红</td>
  <td class="td-mobile">18888881111</td>
  <td class="td-birth">1990年02月08日</td>
  <td>
    <button class="btn edit">编辑</button>
    <button class="btn del">删除</button>
  </td>
</tr>
```

需求

- 新增
 - 弹框及插件整合
 - 添加及关闭弹框
- 删除
- 编辑
 - 进入编辑状态
 - 保存此次修改

人员管理			
新增	姓名	手机	生日
	小黄	18888882222	1990年02月09日
	小绿	18888884444	1990年02月11日
	小青	18888885555	1990年02月12日

步骤

编辑 - 保存此次修改

需求：点击编辑，弹出编辑框，并填入这一行的内容

分析：

①：点击提交校验表单

valid

②：修改被编辑tr的数据

共享tr，修改tr

③：区分新增和编辑的逻辑

判断

④：新增功能受影响了没？





总结

1. 是否需要记忆插件的使用步骤?
不需要，随用随查
2. 弹框的出现和隐藏动画分别是什么?
fadeIn , fadeOut
3. remove方法删除的是自己，还是子元素?
自己



表单序列化

jQuery 中封装了快速获取表单数据的方法，叫做序列化

语法

```
$('#form').serialize()
```

- 表单元素要有 `name` 属性才可以获取到 `value` 值
- 获取到的数据格式是 `name1=value1&name2=value2` 的 字符串



总结

1. `serialize`方法要能够获取到 `value` 值，表单元素要有什么属性？
`name`属性
2. 获取到的数据是什么格式？
字符串， `name1=value1&name2=value2...`



插件机制

插件是 **jQuery** 提供的扩展机制，本质是往 **jQuery** 原型对象上添加方法

语法

```
jQuery.fn.extend({  
  插件名 (参数) {  
    // 逻辑  
  }  
})
```

- **jQuery** 是 **\$** 的别名
- **jQuery** 内部也是通过这种方式添加方法



总结

1. 通过 `jQuery.fn.extend` 是往什么地方添加功能?
原型
2. 导入 `jQuery` 之后，会多2个全局变量，一个是 `$` 一个是什么?
`jQuery`



工具方法

jQuery 除了封装了大量的 DOM 操作外，还提供了一些工具方法，这些方法通过 \$ 或 jQuery 直接调用

语法

```
// 遍历数组
$.each(数组, function (下标, 值) {})  
  
// 遍历并返回新数组
$.map(数组, function(值) {  
    // 返回新的值  
})
```

- 不仅仅只有这2个方法
- 逐步被 ES6 及更高级的版本新增特性取代



总结

1. jQuery提供的工具方法只有2个?

有很多

2. jQuery的工具方法和ES6及更高版本的新特性更推荐用哪个?

后者



版本差异

最新版本 3.6.0

最新版本 3.x

- 3: 大的版本
- .x: bug修复, 小的更新



你的电脑遇到问题，需要重新启动。

我们只收集某些错误信息，然后为你重新启动。(完成100%)

如果你想了解更多信息，则可以稍后在线搜索此错误：INVALID_AFFINITY_SET

jQuery 主要有3个大的版本

- 1.x: 兼容低版本的ie , 最后一个版本是1.12.4
- 2.x: 不兼容低版本ie , 不在更新
- 3.x: 不兼容低版本ie , 更新中
- 兼容信息查看:<https://jquery.com/browser-support/>
- 历史版本下载:<https://code.jquery.com/>



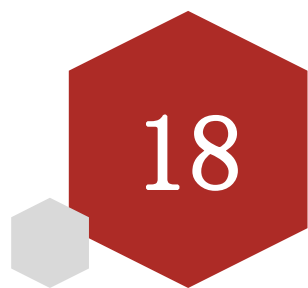
总结

1. 大的版本一共有几个？

3个

2. 不考虑兼容性的话用哪个？

最新版本



多库共存

如果我们也要用 `$` 变量，可以通过 `jQuery` 封装的方法来起别名

语法

```
// 给$起别名 为 _  
let _ = $.noConflict()  
// 查看jquery的版本  
console.log($.fn.jquery)
```

- 导入其他版本的 `jQuery` 需要放在调用方法之后

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



总结

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

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

 [百度一下](#)

 在 Google 上搜索，或者输入一个网址



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“ \times ”相同
/	除	参看小学二年级，与“ \div ”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）

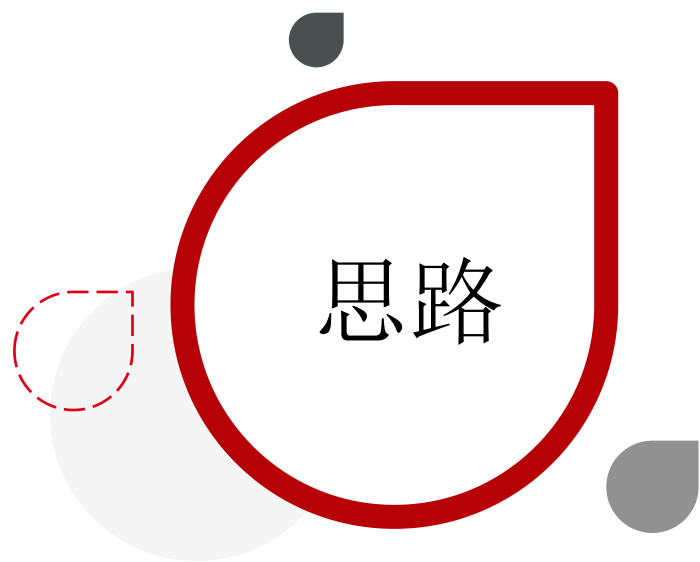
④：百位的计算：数值 / 10 / 10 % 10

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



思考

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

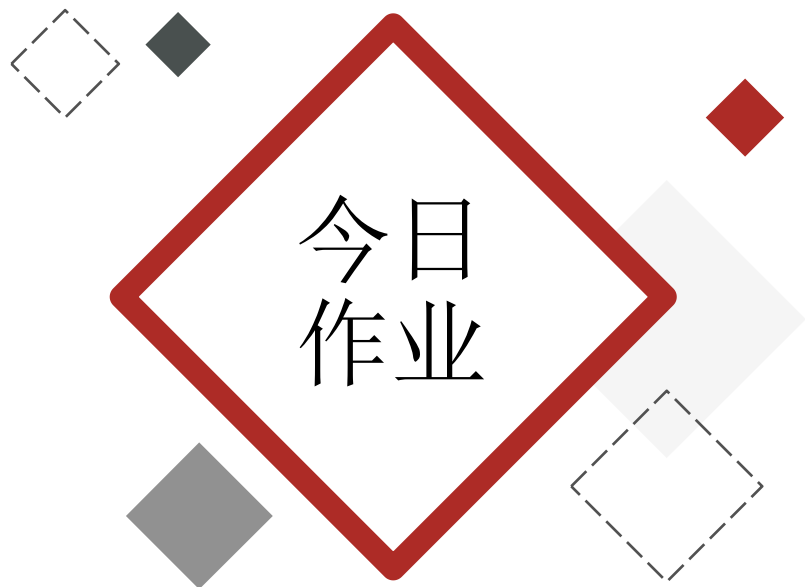


1. 文本居中对齐
2. 根据实际情况调整文本位置，红色色值
3. 力求美观、简洁、大方



总结

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

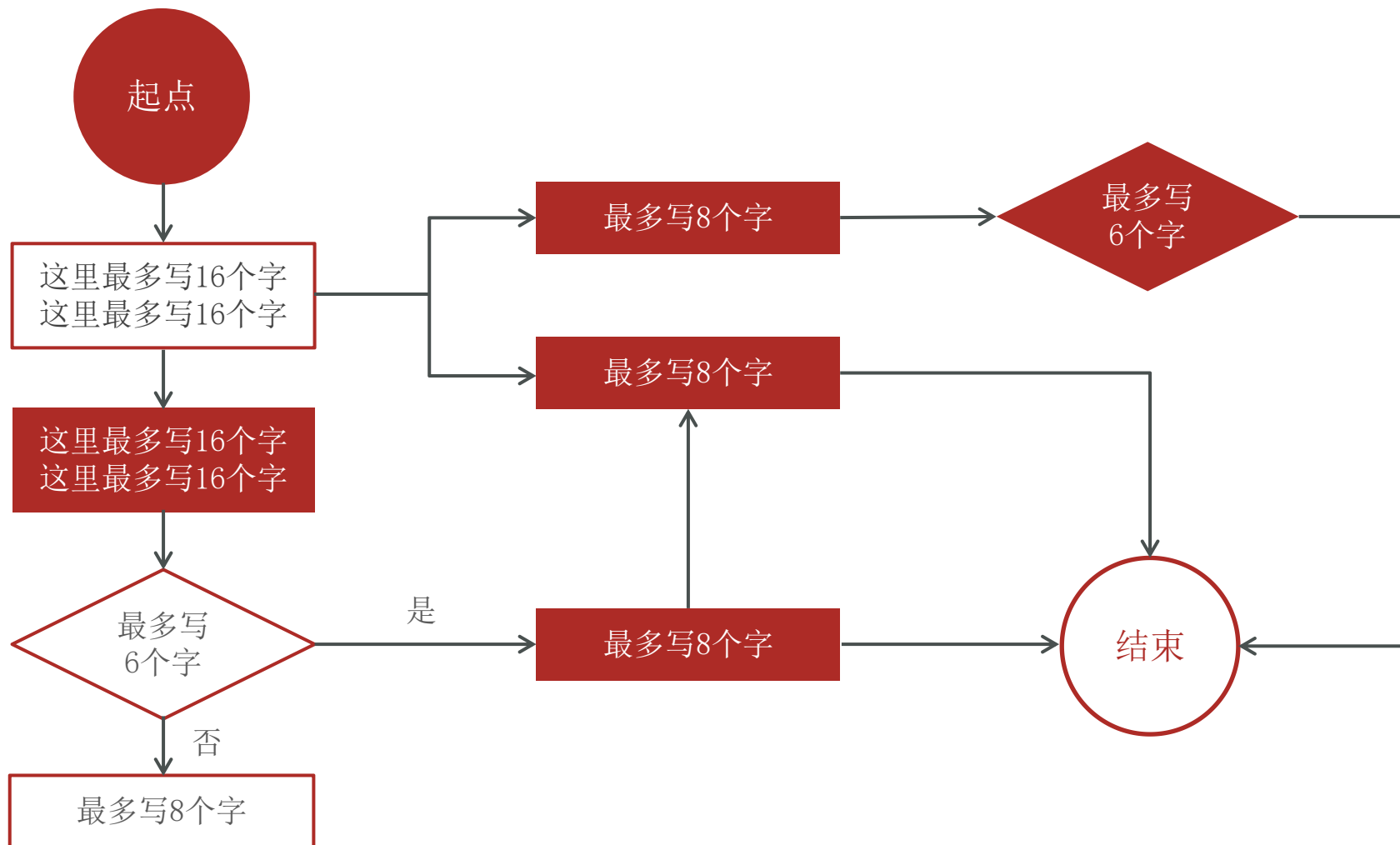
代码样式（非母版，需要请复制，如有红色请用色值#AD2B26）

```
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。

填写标题

填写标题

填写标题

填写标题

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。

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。

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。

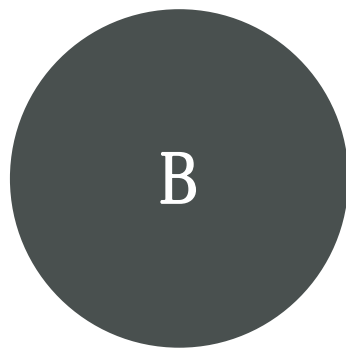


其他的版式样式



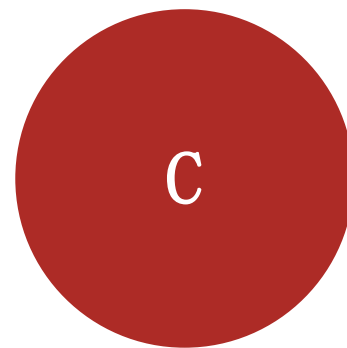
填写段落标题

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。

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