# Javascript 设计模式

第4天课堂笔记（本课程共4天）

[Javascript 设计模式](#_Toc17208)

[复习](#_Toc3911)

[节流模式](#_Toc24409)

[委托模式](#_Toc6358)

[迭代器模式](#_Toc23540)

[MVC](#_Toc15816)

[诞生](#_Toc26866)

[后端语言中的实践](#_Toc16121)

[前端语言中实践](#_Toc6659)

[MVC模型图](#_Toc1651)

[MVC实战](#_Toc18855)

# 复习

## 节流模式

对重复性操作，只执行最后一次操作，取消前面的操作，以提高性能

返回顶部

Icon浮层

## 委托模式

将多个对象或者类的请求委托给另外一个对象统一处理，减少请求的

1 优化事件数量

2 对未来元素的事件绑定

3 jquery中的delegate

4 内存外泄

5 请求分发

## 迭代器模式

Dom迭代器

# MVC

## 诞生

是Xerox PARC（施乐帕罗奥多研究中心）在二十世纪八十年代为编程语言Smalltalk－80发明的一种软件设计模式，已被广泛使用。后来被推荐为Oracle旗下Sun公司Java EE平台的设计模式，并且受到越来越多的使用ColdFusion和PHP的开发者的欢迎

## 后端语言中的实践

Model（模型）表示应用程序核心（比如数据库记录列表）。

是应用程序中用于处理应用程序数据逻辑的部分。通常模型对象负责在数据库中存取数据。

View（视图）显示数据（数据库记录）。

是应用程序中处理数据显示的部分。通常视图是依据模型数据创建的

Controller（控制器）处理输入（写入数据库记录）

是应用程序中处理用户交互的部分。通常控制器负责从视图读取数据，控制用户输入，并向模型发送数据。

## 前端语言中实践

页面的本事是数据信息的读取，有以下特点

对于机器可读性好的数据，人很难读取

对于人很容易理解的数据（图形化界面），机器很难读取

所以前端即是一个将机器可读性好的数据转化成人类可读性好的数据的一个过程，在这个过程中分成三个部分，机器可读性好的数据（model），人类可读性好的数据（view），以及这里各个部分的交互过程（controller）

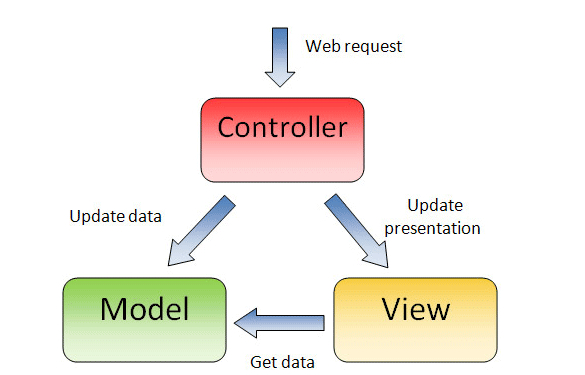
前端MVC实践中分层

Model，页面数据的存储操作

View，渲染出给人看的页面操作

Controller，页面的交互对视图的更改以及对数据的更改

## MVC模型图



View层通过model渲染数据，所以view层可以访问model层

Controller层可以对model写入读取数据，所以controller可以访问model层

Controller层可以对view层打开弹层，浮层等交互，所以controller可以访问view层

在前端的框架中，很多框架是基于MVC模式实现的，比如backbone

它将MVC进行了一些改造，比如他允许Model层可以访问View层，实现数据层的更新通知view层视图的渲染

## MVC实战

通常我们一个模块对应一个控制，一个模型，一个视图，那么如果将页面所有模块中的视图，控制器，模型放在一起，那么我们逻辑比较混乱，通常我们为了管理方便，将一个模块的控制器，模型，视图，放在一个文件内管理，根据他们的父模块不同，要根据父模块作为文件夹设置

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |