# ORM - Object Relational Mapping

1. 定义：

ORM：也即对象 - 关系映射

1. 意义：

数据库常规操作下，操纵的对象针对数据库中的关系表。而ORM机制能让你的处理对象由这些表转换成Python中的表对象。这种机制的意义在于，它能够让你在不了解SQL语句的条件下使用数据库。

三、ORM方法论基于三个核心原则：

1. 简单：以最基本的形式建模数据。
2. 传达性：[数据库结构](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%BA%93%E7%BB%93%E6%9E%84/5507713" \t "https://baike.baidu.com/item/ORM/_blank)被任何人都能理解的语言文档化。
3. 精确性：基于数据模型创建正确标准化的结构。
4. [基于ORM思想的SQLAlchemy库](https://www.cnblogs.com/muchen/p/6882589.html)：

首先安装MySQL和SQLAlchemy(sqlalchemy)库，然后总体步骤如下：

1. 创建一个数据库；
2. 导入SQLAlchemy库相关内容 (详见下面代码展示部分)；
3. 定义表对象结构；
4. 建立数据库连接引擎；
5. 基于 4 中创建的引擎创建一个会话类型；
6. 定义会话对象；
7. 定义并初始化表对象；
8. 插入表对象；
9. 关闭会话。

参考文献：

1. <http://www.cnblogs.com/muchen/p/6882589.html>
2. <https://www.cnblogs.com/zhangyux/p/6284669.html>
3. <https://blog.csdn.net/qq562029186/article/details/81512791>
4. <https://blog.csdn.net/billvsme/article/details/50197197>
5. <https://zhuanlan.zhihu.com/p/27400862>
6. <https://www.jianshu.com/p/8d085e2f2657> (关于[SQLAlchemy](https://www.jianshu.com/p/0d234e14b5d3" \t "https://www.jianshu.com/p/_blank)操作介绍比较好的文章)
7. <http://www.bkjia.com/Pythonjc/1295436.html> （语法）
8. <https://docs.sqlalchemy.org/en/rel_1_0/orm/tutorial.html#working-with-related-objects>(官方)

# MVC—Model View Controller

# 定义 MVC框架：MVC全名是Model View Controller，是模型(model)－视图(view)－控制器(controller)的缩写，一种软件设计典范，用一种业务逻辑、数据、界面显示分离的方法组织代码，将业务逻辑聚集到一个部件里面，在改进和个性化定制界面及用户交互的同时，不需要重新编写业务逻辑。MVC被独特的发展起来用于映射传统的输入、处理和输出功能在一个逻辑的图形化用户界面的结构中。

* + 1. 控制器(controller)：负责转发请求，对请求进行处理，实现业务逻辑。

视图(View)：界面设计人员进行图形界面设计。

模型(Model)：程序员编写程序应有的功能（实现算法等）、数据库管理。实现业务逻辑与数据库分离，易于维护系统。

* + 1. MVC框架：整个框架分为实体层、数据访问层、业务基类层、Web请求转发层、控制层和视图层；

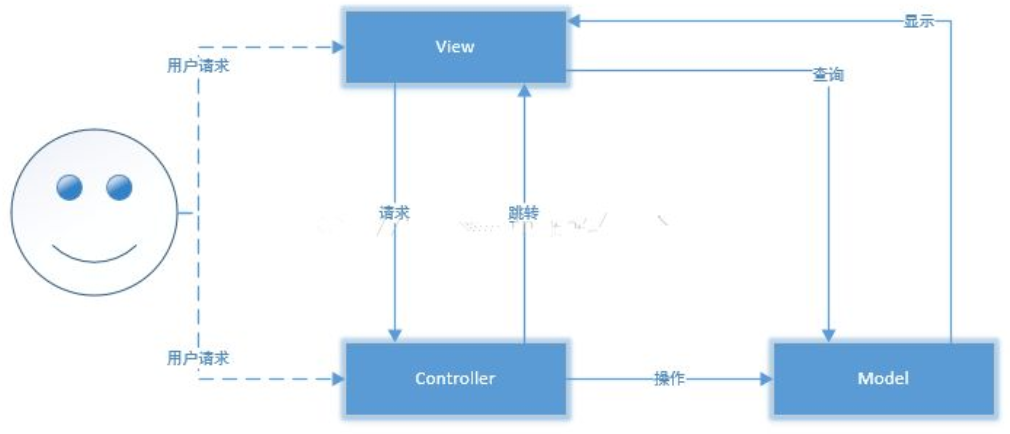


图1.1 MVC架构原理图

1. 意义  
    MVC是一种架构设计模式，一种设计理念。是为了达到分层设计的目的，从而使代码解耦，便于维护和代码的复用。
2. 关于Django架构中MVC思路的介绍

Django框架遵循MVC设计，并衍生出一个自己所属的专有名词：MVT。

MVT各部分的功能如下：

1. M全拼为Model，与MVC中的M功能相同，负责和数据库交互，进行数据处理。
2. V全拼为View，与MVC中的C功能相同，接收请求，进行业务处理，返回应答。
3. T全拼为Template，与MVC中的V功能相同，负责封装构造要返回的html。

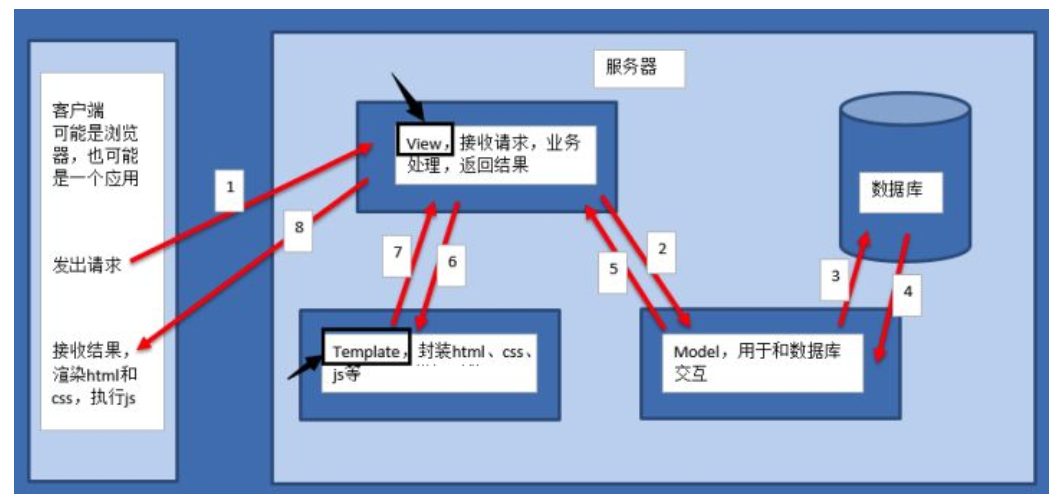


图3.1 Django模型中MVT架构思路

1. Django架构中模板系统介绍
   1. 定义：
      1. Django模板是一个string文本，它用来分离一个文档的展现和数据；
      2. 模板定义了placeholder和表示多种逻辑的tags来规定文档如何展现；
   2. 使用模板的步骤：
      1. 用模板代码创建一个Template对象，
      2. 使用一些给定变量context调用Template对象的render()方法。这将返回一个完全渲染的模板，它是一个string，其中所有的变量和块标签都会根据context得到值；

参考资料：

1. <https://www.cnblogs.com/JoeSnail/p/7724341.html>
2. <http://www.cnblogs.com/jyan/archive/2012/07/23/2604474.html>
3. [介绍Django框架中MVC设计思路的文章](https://zhuanlan.zhihu.com/p/43833483)
4. [Django中文手册](https://docs.pythontab.com/django/django1.5/" \l "template)
5. [Django模板系统介绍—关于HTML的设计](https://blog.csdn.net/zhangxinrun/article/details/8095118)
6. [Python mvc架构django搭建](https://blog.csdn.net/nuli888/article/details/51974143)
7. [Django框架介绍](https://blog.csdn.net/G_SANGSK/article/details/80542710)
8. [Django中使用SQLAlchemy的介绍](https://blog.csdn.net/scdxmoe/article/details/73294179)