# quick start

## CMS 网站管理系统简介

CMS其实是一个很广泛的称呼，从一般的[博客程序](http://baike.baidu.com/view/1705040.htm)，[新闻](http://baike.baidu.com/view/14325.htm)发布程序，到综合性的[网站](http://baike.baidu.com/view/4232.htm)管理程序都可以被称为[内容](http://baike.baidu.com/view/748656.htm)管理系统

## 分类

根据不同的需求，CMS有几种不同的[分类](http://baike.baidu.com/view/330961.htm)方法。比如，根据[应用](http://baike.baidu.com/view/220910.htm)层面的不同，可以被划分为：

* 重视[后台](http://baike.baidu.com/view/179243.htm)管理的CMS
* 重视[风格](http://baike.baidu.com/view/136139.htm)设计的CMS
* 重视[前台](http://baike.baidu.com/view/1334232.htm)发布的CMS

就CMS本身被设计出来的出发点来说，应该是方便一些对于各种网络[编程语言](http://baike.baidu.com/view/552871.htm)并不是很熟悉的用户用一种比较简单的方式来管理自己的网站。这虽然是本身的出发点，但由于各个CMS系统的原创者们自己本身的背景与对“[简单](http://baike.baidu.com/view/66543.htm)”这两个字的理解程度的不同，造成了群雄纷争的局面。

简而言之，CMS就是可以让你不需要学习复杂的建站技术，不需要学习太多复杂的[HTML](http://baike.baidu.com/view/692.htm)语言，你就能够利用CMS构建出一个风格统一功能强大的专业网站。

### 功能

CMS具有许多基于[模板](http://baike.baidu.com/view/160044.htm)的优秀设计，可以加快[网站](http://baike.baidu.com/view/4232.htm)开发的速度和减少开发的成本。CMS的功能并不只限于[文本](http://baike.baidu.com/view/300107.htm)处理，它也可以处理图片、[Flash](http://baike.baidu.com/view/7641.htm)动画、声像流、图像甚至电子邮件档案。

CMS还分各个[平台](http://baike.baidu.com/view/58664.htm)脚本种类的。

内容管理系统是企业信息化建设和[电子政务](http://baike.baidu.com/view/2056.htm)的新宠，也是一个相对较新的市场，对于内容管理，业界还没有一个统一的定义，不同的机构有不同的理解：

Gartner Group 《加特纳集团》认为内容管理从内涵上应该包括企业内部内容管理、Web内容管理、[电子商务](http://baike.baidu.com/view/757.htm)交易内容管理和企业外部网(Extranet)信息共享内容管理（如[CRM](http://baike.baidu.com/view/4799.htm)和SCM等），Web内容管理是当前的重点，e-business《电子商务》和XML是推动内容管理发展的源动力。

### 网站模板

市场上有很多公司自行开发的**CMS**系统模板，几个比较大的网站上也经常能够看到一些国外（欧美，韩国）的模板，当中比较有代表性的，在美国有较大影响力的当属怪兽模板了。起源于美国，随即登陆亚洲市场，瞬间风靡全球的网站建设的崭新模式--怪兽网页模板的CMS网站模板，包括了[WordPress](http://baike.baidu.com/view/23618.htm)，joomla，magento，drupal，mambo等目前市场上最流行的模板。

中国也有一些好的开源项目如：[jeecms](http://baike.baidu.com/view/2313124.htm)、kooboo、wojilu

### 系统开发

内容管理系统是一个很泛的概念：从商业门户网站的新闻系统到个人的Weblog都可以称作[发布](http://baike.baidu.com/view/856622.htm)系统。

* 框架型：本身不包含任何应用实现，只是提供了底层框架，具体应用需要一定的二次开发，比如Cocoon，Vignette；
* 应用型：本身是一个面向具体类型的应用实现，已经包含了新闻/评论管理，投票，论坛，WIKI等一些子系统。比如：postNuke xoops等；

但无论如何，在发布系统选型之前，首先了解自己的实际需求是最重要的：想根据现成系统将自己的需求硬往上照搬是非常不可取的。访问量，权限控制和各种功能 需求。每个模块和功能自己都比较清晰一点以后，你会发现其实每个环节都有比较成熟的实现，如果没有：你的需求太特殊，或者可以尝试分解成更小的系统组合实 现。

内容管理系统被分离成以下几个层面：各个层面优先考虑的需求不同。

* 后台业务子系统管理（管理优先：内容管理）：新闻[录入](http://baike.baidu.com/view/902375.htm)系统，[BBS](http://baike.baidu.com/view/66.htm)论坛子系统，全文[检索](http://baike.baidu.com/view/677662.htm)子系统等，针对不同系统的方便管理者的内容录入：所见即所得的编辑管理界面等，清晰的业务逻辑：各种子系统的权限控制机制等
* Portal系统（表现优先：模板管理）：大部分最终的输出页面：网站首页，子频道/专题页，新闻详情页一般就是各种后台子系统模块的各种组合，这种发布组合逻辑是非常丰富的，Portal系统就是负责以上这些后台子系统的组合表现管理；
* 前台发布（效率优先：发布管理）：面向最终用户的缓存发布，和搜索引擎spider的URL设计等……

内容管理和表现的分离：很多成套的CMS系统没有把后台各种子系统和Portal分离开设计，以至于在Portal层的模板表现管理和新闻子系统的内容管 理逻辑混合在一起，甚至和BBS等子系统的管理都耦合的非常高，整个系统会显得非常庞杂。而且这样的系统各个子系统捆绑的比较死，使得后台的模块很难改 变。但是如果把后台各种子系统内容管理逻辑和前台的表现/发布分离后，Portal和后台各个子系统之间只是数据传递的关系：Portal只决定后台各个 子系统数据的取舍和表现，而后台的各个子系统也都非常容易插拔。

内容管理和数据分发的分离：需要要Portal系统设计的时候注意可缓存性（Cache Friendly）性设计：CMS后台管理和发布机制，本身不要过多考虑"效率"问题，只要最终页面输出设计的比较Cacheable，效率问题可通过更前端专门的缓存服务器解决。

此外，就是除了面向最终[浏览器](http://baike.baidu.com/view/7718.htm)用户外，还要注意面向搜索引擎友好(Search engine Friendly)的URL设计：通过URL REWRITE转向或基于PATH\_INFO的参数解析使得动态网页在链接（URI）形式上更像静态的目录结构，方便网站内容被搜索引擎收录