目录

[前沿 1](#_Toc612376)

[第一章 物联网背景下的PHP概况 3](#_Toc612377)

[1.1 物联网系统的体系架构 3](#_Toc612378)

[1.2 感知层的技术选择 5](#_Toc612379)

[1.3 传输层的技术选择 6](#_Toc612380)

[1.4 应用层的技术选择 7](#_Toc612381)

[1.5 PHP技术的局限性 7](#_Toc612382)

[1.6 小结 7](#_Toc612383)

[第二章 Web服务器开发环境的搭建 9](#_Toc612384)

[2.1 Apache服务器的安装和使用 10](#_Toc612385)

[2.1.1 Ubuntu下Apache服务器的安装 10](#_Toc612386)

[2.1.2 Ubuntu下Apache服务器的设置 11](#_Toc612387)

[2.2 MySQL服务器的安装和使用 12](#_Toc612388)

[2.2.1 Ubuntu下MySQL服务器的安装 12](#_Toc612389)

[2.2.2 MySQL管理工具MySQL Workbench的安装和使用 13](#_Toc612390)

[2.3 PHP运行库的安装和扩展 15](#_Toc612391)

[2.3.1 Ubuntu下的PHP运行库的安装 15](#_Toc612392)

[2.3.2 常用的PHP运行库扩展程序 16](#_Toc612393)

[2.4 小结 16](#_Toc612394)

[第三章 几种常用的Web开发工具 17](#_Toc612395)

[3.1 XDebug开源的PHP调试工具 17](#_Toc612396)

[3.1.1 Linux下XDebug的安装 17](#_Toc612397)

[3.2 Vistual Code开源的跨平台程序编辑器 18](#_Toc612398)

[3.2.1 Linux下Vistual Code的安装 19](#_Toc612399)

[3.2.2 Vistual Code中PHP相关插件的安装 19](#_Toc612400)

[3.3 Eclipse for PHP Developers全能的程序编辑器 21](#_Toc612401)

[3.3.1 Linux下Oracle Java的安装 22](#_Toc612402)

[3.3.2 Linux下Eclipse for PHP Developers的安装 23](#_Toc612403)

[3.4 Chrome DevTools实用的前端调试工具 24](#_Toc612404)

[3.5 小结 25](#_Toc612405)

# 前沿

狄更斯在他的双城记中第一句话就说：这是一个最好的时代，也是一个最坏的时代。

对于程序员来说确实是这样，回顾2016年可以说是前端和服务端齐头并进的时代，是新旧更替的时代，是众多新技术被广泛验证的时代，如经过2016年一整年洗礼后的React几乎成为了企业开发的标准配置，老牌的前端框架Bootstrap依然老骥伏枥，最新的Bootstrap 4已经进入了Alpha阶段，为2017年的发力做最后的准备，又如PHP 7.0在2015年下半年发布了RC版，在半年时间里收集问题、迭代更新，于2016年1月6日正式发布了7.0.2版本，使得PHP的整体性能提高了至少一倍以上，同时PHP的5.6也得到了长期的支持，最新的PHP 5.6.29在2016年12月8日正式发布，进一步巩固了PHP在服务的地位，此外阿里在2016年10月14日的杭州云栖大会上发布了基于MySQL数据库的AliSQL，在普通环境下其性能在MySQL的基础上提高了70%，而在特定环境下性能可以提高近百倍。

对于程序工程师来说，除了技术的变化，岗位的变更也在悄悄的进行，原先的前端工程师主要责任是负责Web页面的开发，但是到了2016年前端工程师熟悉的JavaScript不仅可以编写Web编写，还可以进行跨平台的应用开发，从大屏到小屏都能胜任，做到程序的一次编写就可以适用多种屏幕，甚至在性能方便不输于采用Java或者Object-C所开发的原生应用，此外原先负责后台应用的工程师也随着大数据、云计算普及也向着数据分析、数据整合、以及数据挖掘发展，特别是越来越多的中小型的创业企业通过运用类似与ThinkPHP这样基于PHP语言的快速开发框架从而验证项目的可行性，以确定项目的未来走向。

因此可以看出2016年中每一天我们的身边都在进行了翻天覆地的变化，技术的发展速度远远超过了个人学习的速度，可能几天不刷论坛，当再进入时就会冒出几个新名词，再过几天网上已经炒的沸沸扬扬，因此没有一本书或者作者能够一直站在技术的最前端，那么有没有一个全栈技术能够经久不衰，任凭外面技术如果的变化，但是它一直都能解决大部分企业的问题？答案是肯定的，采用Bootstrap+PHP+MySQL就可以。

本书从实际工程应用入手，通过实验过程和实验现象讲解系统间的内在联系，并且通过研究各系统的设计原理和设计缘起，讲解在进行系统开发时的思考方法，由浅入深、循序渐进的方式深刻剖析了如HTML 5的语义设计思想、PHP 5中的面向对象的构建理念，以及MySQL数据库中的范式的意义，本书不同于传统讲解前端或者后端的书籍，把两者之间进行割裂，读者在阅读后依然没有一个完善的体系在心中，本书想通过真实的项目把作者的实践中的所思所想、学习方法，以及相对较为完善的体系与各位读者分享，从而能够快速适应这个变化的时代。另外，本书的内容均来自作者科研、教学和项目实践，因此所涵盖的内容是作者多年来经验的总结。

本书适用读者

目前国内对于各项技术相对入门的内容都有完善的书籍，因此本书在编写时对于一些相对基础的内容加入的较少，主要是以最新的、最稳定的技术为切入点，讲述在具体项目中的开发思路，以及对许多技术中相对原理性，但是又一般在很多书籍中不涉及的内容进行项目的阐述，因此在阅读本书时需要有一定的HTML5、CSS3、PHP、MySQL基础，或者这些技术有一定的了解，主要面向的读者有一下四种：

1、对Web开发有兴趣，并想深入了解原理的读者

2、基于Web的移动互联网开发人员

3、在大中专院校进行Web相关研究的老师

4、有志从事Web开发的在校学生

本书内容与组织结构

本书是我们团队编写的关于Web开发系列丛书中的一本，目的是帮助有HTML或者PHP相关基础的工程师能够通过学习本书更加深刻的理解各系统的运作机理，并把这些应用于实际项目中，从而提高项目的可维护性和可扩展性，全书共分为8个章节。

第1章介绍了物联网环境中PHP如何定义，并且给出了物理网的体系架构和模块通信方式。

第2章介绍了解Web全栈系统的开发环境的搭建。

第3章介绍了在生产环境中Web全栈系统开发所用到的工具，以及这些工具的配置方法和使用方法，从帮助工程师提高开发效率。

第4章探讨了在面向对象程序设计思想下的PHP程序设计方法，并且讨论了在程序运行时内存的分配和回收机制，帮助工程师深入理解系统的工作原理。

第5章探讨了在HTML5中语义标签的使用方法和设计理念，然后讨论利用面向对象的概念来设计CSS3样式表，最后讲解面向对象的JavaScript的设计思路。

第6章探讨了响应式框架Bootstrap的设计原理和设计思路，并基于此讨论在前端开发中的开发规范，以及封装方法，然后在Bootstrap的基础上开发一个基于学生工作过程管理系统的登录界面。

第7章讨论了数据库设计时的几种范式的作用，并且讨论、设计出基于学生工作过程的数据库，最后探讨通过常用数据库操作指令来完成对数据库的操作。

第8章探讨了基于最新ThinkPHP 5.x框架的应用开发方法，并且在其中学习ThinkPHP的设计原理和实现过程，从而使得工程师能够更加的灵活运用该框架设计出高效的系统。

# 第一章 物联网背景下的PHP概况

毛主席在其《矛盾论：矛盾的特殊性》中说到针对问题不能“只见树木，不见森林”，头疼医头脚疼医脚。学习一项技术也是这样，需要充分了解这项技术所处的环境，以及应用场景，当前该技术最新的发展情况如何，有哪些已经是成熟的方法被广泛使用，以及这项技术的优缺点有哪些，只有这样全面的了解过后再深入的学习时就能够有的放矢、循序渐进的开展，并构建起自己完整的知识体系，从而为后续学习打下坚实的基础。不同于以往的小规模商业试用和科学研究，随着NB-IOT、LoRa等物联网技术在运营商和大型企业的推动下众多的物联网项目犹如雨后春笋般在国内落地开花，而PHP以其自身简单、易用、稳定，同时还具有成熟的生态环境等优势在物联网项目大量落地的今天也占得了一席之地。物联网项目不同于其他软件项目，它是大系统的集成，既包含了软件又包含了硬件，因此需要众多的技术相互配合才能完成。因此，本章首先想通过以构建一张完整的物联网技术架构，讲解PHP在其中是如何做到承上启下，然后以此为基础为各个功能子模块提供实现的解决思路，最后通过PHP局限性的说明给读者一个辩证的思路，从而在后续的项目开展中能够对技术的选择有清晰的认识。

本章内容

* 物联网系统的体系架构
* 感知层的技术选择
* 传输层的技术选择
* 应用层的技术选择
* PHP技术的局限性

## 1.1 物联网系统的体系架构

“物联网”这词语最初的时候是一个外来词汇，其完整的英文名叫“Internet Of Things”（简称：IOT），顾名思义就是很多物体连成的网络，所以叫做物联网。随着人工智能等技术在国家层面的大力扶持与推广，2018年下半年又出现了一个新的词汇——AIoT，即人工智能（AI）与物联网（IOT）相结合。虽然这几年各种新技术不断涌现，各个单独模块的具体实现上通过采用各种更为先进的技术进行不断改进，而物联网的基本体系架构依然没有发生根本性的改变，因此只要深入的了解物联网系统的体系架构和运作模式，读者就可以脱离纷繁复杂的技术局限，根据自身的技术特点和擅长把各类具体的实现手段进行自由组合，从而实现一个或简单或复杂的物联网系统。

要学习一个技术首先需要了解该技术最初时是用来解决什么问题，下面通过三个故事来讲述物联网的源起。

1、故事一：1991年剑桥大学特洛伊计算机实验室的科学家们为了解决去楼下倒咖啡经常空手而归的问题，在楼下咖啡壶旁安装了一个摄像头，通过图像捕捉技术实时分析咖啡是否煮好，从而帮助实验室的科学家们可以随时监控咖啡的情况，而不用经常白跑一趟。到了1993年，这套咖啡监控系统被搬到的互联网，有近240万人点击浏览了这个监控网站，甚至有更多的人希望能亲眼目睹这个已经成为网红的“特洛伊咖啡壶”。虽然现在看来这是一个很小的项目，但可以说它包含了物联网体系架构中的大部分内容，而随后的不论市场开发还是基于应用都可以找到这个咖啡壶的影子。

2、故事二：1995年微软的创始人比尔盖茨在他的新书《未来之路》中描述了物联网的基本雏形，即当时的互联网已经实现了计算机和计算机之间的联网，但是还没有实现万物互联的情况，但他相信随着技术的不断发展，未来一定可以实现。可以说比尔盖茨在他的这本预言书中所描绘的物联网情景正在被逐步实现。

3、故事三：1998年英国工程师凯文·艾什顿发现宝洁公司的不少零售店的库存信息与销售状态不符，于是在宝洁公司的一次演讲中首次提出了“物联网”的概念。随后宝洁公司与MIT成立了自动识别研究中心，主要研究RFID技术以及智能包装系统。当时，艾什顿对物联网的定义相对也更具针对性，即把各类商品通过射频识别等技术与互联网相连接，从而实现智能识别与智能管理。而MIT的自动识别研究中心提出的物联网概念就相对宽泛，即在互联网的基础上，利用RFID、无线传感器网络、数据通信等技术，构造一个覆盖世界上万事万物的“物联网”，在这个网络中，物品之间能够彼此“交流”，而无需人为干预。

从上面三个故事不难看出，不论最初的特洛伊咖啡壶还是之后的MIT的自动识别研究中心，他们所构建的物联网都有一个相同的体系架构，即三层架构，分别是感知层、网络层和应用层，如图1-1所示，具体如下：

1、感知层：主要负责获取和控制各类传感器和智能设备的数据，如小到简单的温湿度传感器，大到复杂的智能设备，如机械臂等。这部分的实现主要通过各类智能硬件来完成，其本质就是在传统的嵌入式系统上再加上一个网络模块，从而实现网络通信，并且把以往保存在本地的设备运行过程数据，通过格式化的数据传递到远端服务器。

2、网络层：主要负责把感知层的数据上传给远端应用层，以及把应用层的计算结果下发至感知层。这部分可以根据具体需求的选择是采用广域网的方式（俗称“上云”）还是采用局域网的方式，所谓城域网其本质也是局域网的一种。局域网和和广域网除了本身设备的成本之外，在使用上广域网设备还需要向运营商支付一笔网络使用费，但网络的维护由各个运营商来完成，而局域网则不需要网络使用费，但是网络的持续维护需要项目建设方来完成。

3、应用层：可以说是这个物联网系统中的计算大脑，它把从感知层、用户操作等多方面收集到的数据进行计算、分类、统计、识别等一系列操作，并把一部分结果下发至感知层，从而实现这个物联网系统从底至上再到底的闭环。应用层所包含的技术最为复杂，也是延展性最大的部分，可以说所有和计算机相关的技术都能够应用到该部分中，小到一个设备管理系统，大到包括人工智能、移动互联网、机器人等应用。

如果再要把物联网进行形象的比如，那么就可以和人进行类比，感知层就相当于人体的各个运动部件和感觉器官，而网络层则是人体内的各个传导神经和中枢神经，网络层把从感觉器官获取的信息传递到后面的应用层，而应用层就是人体的大脑，用于对收集到的数据进行计算，并反馈到运动部件进行控制。

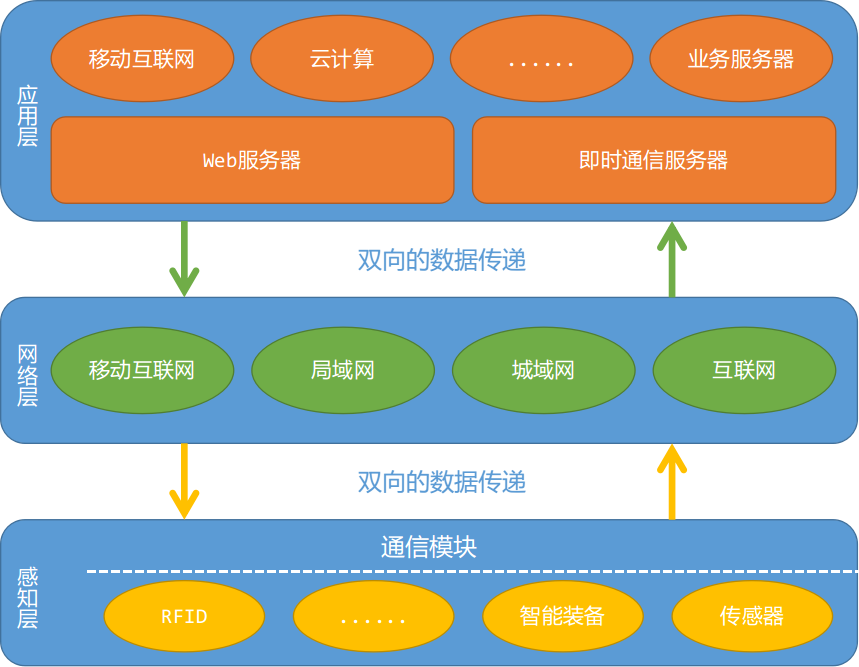


图1-1 物联网体系架构

## 1.2 感知层的技术选择

感知层的相关技术主要偏向于嵌入式系统与应用，要讲解这部分的技术可以从两个层面进行说明，分别是硬件层面和软件层面这两个方向进行，具体如下：

1、硬件层面： 根据不同的应用领域，硬件的选择大致可以分为两大类，分别是低频小数据量的信息采集与控制和高频大数据量的信息采集与控制。针对低频小数据量的信息采集与控制的这一类应用经常会采用ARM公司的Cortex-M系列相关内核的SOC芯片，其中最常用的内核为Cortex-M3和Cortex-M0。所谓SOC芯片就是片上系统，简单的理解就是在一个芯片内部集成的控制器、存储器、外围通信接口等众多内容，类似与一个完整的电脑主机，而ARM公司本身并不生产芯片，只设计相关的芯片内核，然后把这些设计的芯片内核授权给各个芯片生产商，各芯片厂商再根据所授权的IP进行例如存储器、外围通信的扩展与裁剪，最后生产出自己的芯片，所以在选择芯片时会发现即使是相同内核，如Cortex-M3，也会有众多的厂商生产应用于不同领域的各类芯片，类似于很多电脑主机，虽然都是相同的CPU，但有的内存大些，有的硬盘大些，形成中众多不同类型的电脑。目前，常见芯片厂商有意法半导体（STMicroelectronics）、恩智浦（NXP）、德州仪器（TI）等，这些公司的所生产的芯片都有各自的特点和应用领域，甚至在某些细分行业的设备研发和生产中由于一些芯片公司常年的深耕，慢慢形成了一定的行业规则，即只使其中一家或几家所生产的芯片。而针对高频大数据量的信息采集与控制就需要更具不同的应用场景，以及研发成本综合考虑是使用FPGA、GPU或Cortex-A相关芯片，然而往往为满足要求会采用多芯片的组合的解决方案来实现低成本、高性能，例如控制用Cortex-A或Cortex-M系列，而计算则使用FPGA或GPU等。此外，在研发硬件设备时还需要考虑后续的应用场景，以及是否需要出口到其他国家，因为很多场景的产品都有其国标，以及出口目标国的标准，产品必须符合这些标准才能够进行量产和售卖。

2、软件层面： 根据不同的需求，软件可以分为三大类，分别是无操作系统软件、实时操作系统（RTOS）软件和分时操作系统（OS）软件，其中无操作系统软件即裸机程序，这在Cortex-M系列芯片上使用的最为常见，它通过直接配置和读写寄存器就能很好的完成各项任务，包括传感器的读取，数据的收发等工作。而当任务要求具有较高响应或程序体量不断扩大时就需要考虑引入实时操作系统，该操作系统能够以足够快的速度处理各类请求，其处理的结果又能在规定的时间之内进行响应，同时还控制所有实时任务协调一致的运行。目前实时操作系统较为常用的有RTThread、FreeRTOS、UCOS-II/III等，其中RTThread是国人自主研发的嵌入式实时操作系统，并且经过近10年的发展，目前已经非常稳定，并且各类组件和文档都非常的丰富和完善，而FreeRTOS和UCOS-II/III则是在国际上使用非常多的操作系统，其中UCOS-II/III最早引入到国内，有专门的书籍讲解该操作系统的使用，但该操作系统只能用于学习和非盈利，如果商用就需要支付一定的费用，FreeRTOS则不同，不论商用还是学习都不收取任何的费用，但资料相对较少。有了操作系统还需要一个小屏幕进行人机交互，目前针对RTOS的UI库有三种，一种是由德国Segger开发的emWin，该库开发简单，但界面相对较老，属于Win95/98的风格，此外还有由Draupner Graphics开发的TochGFX，该UI库效果非常炫酷，但是开发难度较大，以及RTThread开发商自主研发的Persimmon（柿饼），该UI库开发简单，效果也相对现代。除了实时操作系统，另外一种就是分时操作系统，也是最常说的操作系统，嵌入式领域中针对民用的其实只有Linux一种选择，而军用和航空级的也还有一些其他选择，该操作系统通常都应用于Cortex-A系列的芯片上，用于处理如音视频或大数据量的网络通信等内容的处理。

## 1.3 传输层的技术选择

传输层的相关技术主要偏向用网络通信，要讲解这部分的技术可以从三个层面进行说明，分别是感知层的网络出口、网络通信、应用层的网络入口，具体如下：

1、感知层的网络出口： 目前感知层的网络通信按照研发方式的不同可以分为两种方式，第一种方式是购买现成的网络通信模块，通过主芯片的串行通信接口把要发送的数据发送给网络通信模块，由该模块进行数据的发送和接收，采用这种方式的优势是开发简单，周期也相对较短，当通信模块发生改变时几乎不需要修改任何程序就可以直接使用，但劣势也很明显，就是会增加设备的硬件成本。第二种方式是通过自行对协议栈软件进行研发和实现，常用的协议栈软件有Z-Stack、LwIP等，采用这种方式优点是硬件成本较低，并且可以进行深度定制，同时还可以可以充分利用网络芯片的其他硬件资源，同样劣势也很明显，即开发周期长，对研发人员有较高的要求，同时程序的稳定性也有待后续的验证。

2、通信网络： 目前通信网络的分类主要针对的是无线通信，而如果采用有线通信方式，那么通常需要使用RJ-45或者RS-485等方式连接至中心机房后再传递至云端。而无线通信则可以分为两大类，分别是广域网通信和局域网通信。其中如果采用广域网通信，那么理论上就有三种解决方案，即GSM、NB-IOT、eMTC，其中NB-IOT和eMTC都具有低功耗的特性，同时NB-IOT和eMTC可以直接在GSM网络进行建设，这也是运营商主推这些技术的原因，目前要使用该技术主要的方式是和各个本地运营商进行谈判，并购买相应的物联网卡，以及从运营商或者相关渠道购买通信模块，然后集成到自己的硬件中，然后通过AT指令就可以和服务器交互。目前各个运营商也都在实验多模通信方式，未来在一个基站上能够同时实现GSM、NB-IOT、eMTC通信，形象的比喻就是手机里的全网通。这里需要注意的是，很多基站为了保证数据通信的效率都会定期清除长时间没有数据交互的通道，因此采用GSM、NB-IOT或eMTC时需要定期发送心跳包，以保证数据链路的存在。而如果采用局域网方式那么就要根据通信范围的大小进行选择，例如通信范围为一个城市，那么可以采用LoRa通信技术来实现，通常来说低功耗一般很难覆盖远距离，而远距离一般都是功耗高，而LoRa最大特点就是在同样的功耗下传播距离是其他无线方式传的3-5倍。因为LoRa的传输距离是以公里为单位，因此通过少量的中继就可以覆盖很广的范围，而如果是通信范围是为一个区域，那么可以采用Zigbee或者蓝牙4.0+等方式来实现组网和数据通信，虽然采用这两个技术的模块单独一个只有几十米到几百米的通信距离，但是可以通过中继的方式实现无线的扩展。此外，对于局域网通信来说，不论是采用LoRa这样的远距离通信，还是Zigbee、蓝牙这样的短距离通信，要实现与服务器的交互都必须要通过一个网关来实现，这就要求开发人员对每个节点的分布和每个节点有单独的编号和统一的管理，以便在远端分辨出不同的设备。

3、应用层的网络入口： 数据一旦需要传递到远端服务器，那么就有两种方式可以选择，一种是HTTP通信，另外一种则是Socket通信。其中HTTP通信是一种短连接的通信方式，通信双方交互完数据后就会把数据链路断开，如果需要再次通信，那么就要与服务器重新链接，此外由于建立链接的过程需要经过多次握手通信，设备的功耗也相应提高，所以这种通信更适合不频繁的数据操作，例如用户交互等。要采用这种这种方式通常只需要架设Web服务器，如Apache或Nginx服务器，再配合本书后面讲的基于MVC架构的ThinkPHP软件库就可以实现。而Socket通信则是一种长连接的通信方式，通信双方的链路始终保持接通，方便了数据的实时传递，也免去了多次握手的情况，因此这种方式更适合对数据实时性要求较高或者要求低功耗的通信，其软件可以采用基于PHP的高性能异步通信解决方案，如Workerman和Swoole，其中Workerman的特点就是开发容易、能满足一般的开发需求，部署也相对简单，由于采用纯PHP进行开发，因此效率上略逊于Swoole，而Swoole采用纯C语言进行扩展，因此内存管理、通信协议、效率上略高于Workerman，但是开发相对复杂，周期也相应的有所增加。在实际的物联网项目开发中，HTTP和Socket都会同时用到，因此不论是HTTP服务器还是Socket的服务器都需要进部署才能满足各种网络通信。

## 1.4 应用层的技术选择

应用层的技术选择最为复杂，根据不同的需求可以说是千变万化，但总体上来说由三部分组成，分别是业务服务器、数据服务器，以及前端展示。其中业务服务器主要完成从Web服务器和Socket服务器传递来的数据进行计算和分析，以本书讲解的内容为基础，简单的构建就是把Socket服务器和ThinkPHP进行整合，并在其中完成相应的业务计算，以及数据反馈，而复杂构架则可以引入更多的分层服务器，从而实现更加复杂的内容。而对于数据库服务器来说，通常采用MySQL作为基础，如果项目的体量很大，以及要求响应速度较高，那么就可以再配合Hadoop等分布式的数据库，以及Redis这类内存数据库来提高相应速度。最后的前端展示，则是可以采用移动端与Web端来完成，其中移动端除了采用原生方式进行开发外，还可以采用目前最为流行的H5方式进行实现，其技术方案可以从Angular+Ionic、React Native、Weex中进行选择，而Web端的逻辑可以采用JQuery、Angular或Vue等方式来实现，而UI则有众多的选择，这里推荐Bootstrap、Amaze UI、NG-ZORRO或Element，这些都是较为现代的UI框架。

## 1.5 PHP技术的局限性

再好的东西只要寻找都能找出问题，就像PHP一样，经过20多年的发展与完善，使得它具有开发快速、配置简单、性能稳定、容易部署、生态完善等一系列优点，虽然不论何时都不断有人指出它的各种缺点，但依然阻止不了它成为Web领域中使用最为广泛的语言之一。但为了在进行技术选型时能否充分的辩证思维，也需要了解该技术的局限性，使得在项目实施时可以对对可能出现的问题和瓶颈也有一个合理的预判。PHP的诞生之初就已经决定这一门语言的局限应，即它所面向的技术领域就是Web应用，并且追求的是入门简单、快速开发，这些定位就决定的PHP具有如下三点问题：

1、页面级的资源回收：PHP属于脚本语言，采用的是解释运行机制。这种运行机制使得当PHP页面被解释并执行完成后所有的相关资源都会被回收。也就是说，PHP语言本身就不支持对象驻留等操作，因此不论变量是全局变量，还是静态成员，又或者是局部变量都会在页面执行完毕后被清空。

2、单线程、非异步的运行模式：由于历史和最初设计的原因使得该模式始终存在，因此PHP从根本上就无法应对成千上万大规模的并发请求。

3、共享的全局配置文件：这样的方式使得PHP在部署时无法进行个性化的配置和使用。

其实PHP还有很多其他问题，如函数命名不规范、安全机制相对欠缺等问题，但总体来说PHP作为一个快速验证市场或小规模应用的项目来说非常合适，同时完善的生态系统也为PHP提供许多经得起实验考验的框架，这些能否帮助开发人员快速的进行开发，减少开发中的问题，而对于大型项目来说PHP作为非核心的前端承载也非常合适。

## 1.6 小结

通过本章的学习，了解了PHP在物联网背景的应用场景和使用方式，并且深入了解了物联网中感知层、网络层以及应用层常用的技术。在本章中首先详细讲解了物联网的体系架构，以及该架构中各模块之间通信方式，其次详细了解感知层中硬件部分的基本选型方式和软件的开发模式，然后有了解了在传输层中常用的几种通信方式的特点，以及如何利用PHP实现这些数据的接收，接下来又讲解了应用层中使用的一般技术和几种常用的软件框架，最后讲了PHP技术的局限性来帮助读者能否辩证的看待问题。通过这些知识的学习读者基本可以对物联网中的各个模块的运行方式有了一个清晰的认识，由于物联网本身就是多种技术的组合，因此本章内容也只能起到引路的作用，就像提到的各种问题和各种框架，其实都可以通过其他方式进行弥补，以及替代，但所涉及的内容又会超出本章，甚至本书的内容，因此本章只能做到临渊羡鱼,不如退而结网。

# 第二章 Web服务器开发环境的搭建

古人有云：“不积跬步无以至千里，不积小流无以成江海”。软件开发也是这样，要运行一个复杂系统就必须要从一砖一瓦开始搭建起来。默认情况下计算机是无法识别PHP程序，因此想让计算机能够运行PHP程序，就需要在计算机上搭建一个PHP程序的运行环境，这个运行环境包括三个部分，分别为Web服务器、PHP解析器、数据库服务器，这三个部分有机的结合支撑了这个世界上所有的Web应用，包括网站、短链接的APP等，当然随着应用不断的复杂，这里面内容也会跟着发生巨大的变化，但概括来说整个系统运行分为以下六步，如图2-1所示，

第一步：用户通过HTTP向Web服务器发送请求，这个请求可以理解为在浏览器中输入一条网址。

第二步：Web服务器接收到请求后，就会去读取请求路径下的代码文件。

第三步：如果Web服务器发现代码为HTML这样的静态文件，那么就会直接解析，并且还给用户，如果发现代码文件不是静态文件，那么就会把该文件交给PHP解析器进行解析，同时等待PHP解析器把解析好的静态数据，并返回给Web服务器。

第四步：PHP解析器收到Web服务器提交的任务就会开始解析PHP文件，并把结果以静态数据的形式返回给Web服务器。

第五步：PHP解析器在解析的过程中很多时候会对数据库进行读写操作，从而使页面数据动态化，也就是为什么不同用户登陆到某个论坛显示的用户信息都是不同的。

第六步：当所有代码和数据都解析完成后PHP解析器会把这些数据变为静态数据返回给Web服务器，Web服务器在把数据返回给用户，从而看到的就是动态页面。

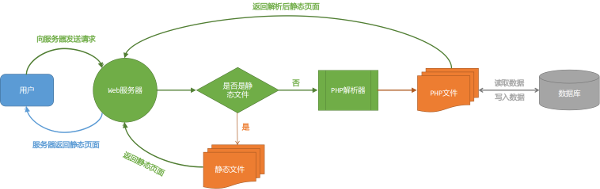


图2-1 Web服务器运行流程

目前世界上90%以上的Web服务器都是采用基于Linux操作系统，因此本书也是基于Linux的一个非常流程的发行版Ubuntu进行讲解，从而帮助读者可以快速的把书本内容应用于实际中。在Ubuntu系统中搭建完整的PHP开发环境的工具称之为LAMP，这里所说的LAMP不是一个工具，而是取四个工具名称的首字母的统称，这四个工具分别为Linux操作系统、Apache服务器、MySQL服务器、PHP，除了Linux操作系统，本章讲讲解在Ubuntu下如何安装另外三个工具。

本章内容

* Ubuntu下的Apache服务器的安装和使用
* Ubuntu下的MySQL服务器，及管理工具的安装和使用
* Ubuntu下的PHP的安装和使用

## 2.1 Apache服务器的安装和使用

目前被广泛使用的Web服务器有三种，分别是Apache、IIS、Nginx，虽然Apache已经有超过20年的历史，几乎可以说是伴随着Web发展一路走来，虽然在计算机领域里面20年可以算是老爷爷了，但是折不妨碍Apache成为Web服务器领域的领头羊，即使在今天有IIS和Nginx冲击的情况，依然不妨碍它成为全世界使用最为广泛的Web服务器，市场占有率超过50%。在国内如淘宝、优酷、新浪等知名网站都采用Apache或基于Apache的优化版作为其Web服务器，Apache的图标如图2-2所示。



图2-2 Apache的图标

### 2.1.1 Ubuntu下Apache服务器的安装

在Ubuntu下安装Apache服务器是非常简单的，因此在Ubuntu的软件源中已经包含了Apache服务器，所以只需要在终端里输入一条指令，并根据提示操作就可以完成Apache服务器的安装，当安装完服务器再在终端中执行apache2 -v，如果能返回Apache版本就说明安装成功，如图2-3所示，具体指令如下。

1 sudo apt-get install apache2

2 apache2 -v

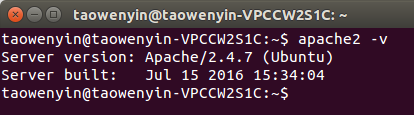


图2-3 Apache安装验证

当安装完Apache服务器时，读者可能会发现在终端里显示一个错误，如图2-4所示。这个错误的意思是没有给服务器指定域名，因此只需要在配置文件中为Apache服务器设置域名就可以，具体步骤如下。

第一步：在终端中执行下面的指令，将当前目录切换到/etc/apache2目录下。

1 cd /etc/apache2/

第二步：在终端中执行下面的指令，打开apache2.conf文件，并在最后添加ServerName localhost:80，如图2-5所示。

1 sudo gedit apache2.conf

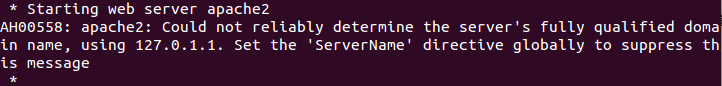


图2-4 安装Apache的错误

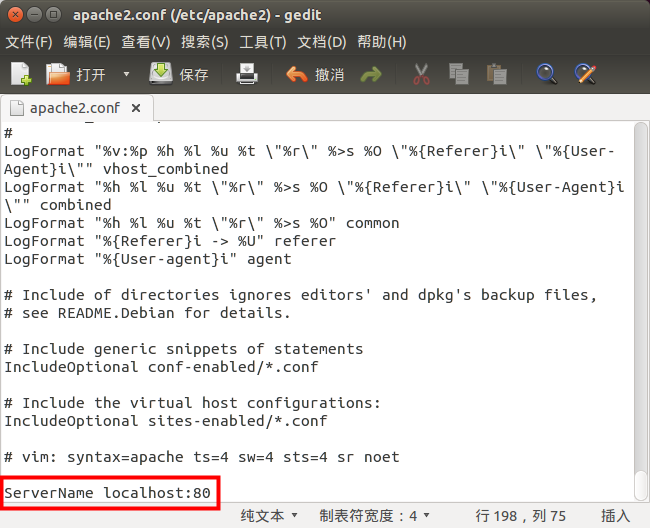


图2-5 为Apache设置ServerName

第三步：在终端中执行下面的指令来重启服务器，此时就不会看到错误产生。

1 sudo service apache2 restart

第四步：打开浏览器，在地址栏中输入localhost，如果看到Apache的主页，就表示安装完成，如图2-6所示。

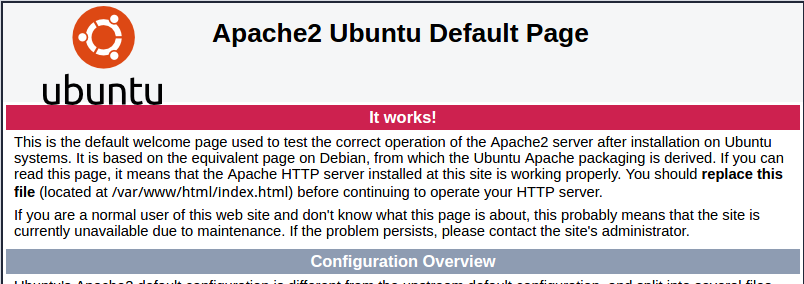


图2-6 验证Apache设置成功

### 2.1.2 Ubuntu下Apache服务器的设置

Apache服务器的主目录是经常要修改的内容，因此本节将讲述如何通过修改配置文件来实现Apache主目录位置的修改。Apache主目录位置的修改主要是修改/etc/apache2目录下面的apache2.conf文件和sites-available/000-default.conf文件。其中apache2.conf主要是修改主目录的位置，而sites-available/000-default.conf则是修改域名重定向时所进入的子目录，例如当前设置Apache主目录位置为/var/www/,而000-default.conf则设置localhost域名指向/var/www/html,那么当在浏览器中输入localhost时实际上读取的是/var/www/html下的文件，所以000-default.conf指定的目录一定是apache2.conf中主目录下的子目录。具体如下：

第一步：在终端中执行下面的命令打开，打开/etc/apache2/apache2.conf文件，并把164行中的/var/www/改成主目录的位置，然后保存并关闭。

1 sudo gedit /etc/apache2/apache2.conf

第二步：在终端中执行下面的命令打开，打开/etc/apache2/sites-available/000-default.conf文件，并把12行中的/var/www/html改成主目录下的子目录，当然也可以和主目录写的一样，那么此时就直接读取主目录下的文件，然后保存并关闭。

1 sudo gedit /etc/apache2/sites-available/000-default.conf

第三步：在终端中执行下面的命令重启Apache服务器，然后在浏览器中输入localhost就可以看到当前的目录已经切换到所指定的目录下。

1 sudo service apache2 restart

## 2.2 MySQL服务器的安装和使用

目前单机版的应用越来越少，即使是笔记本这类应用都已经从原来系统自带的笔记本变成如印象笔记、有道云笔记等类似的云笔记,而在游戏领域，原先的PlayStation和XBox上基本都是单机版的游戏，而最新的PS4和Xbox One上的游戏也由单机版向网络版发展的趋势，这些基于网络的应用都是通过有线或者无线的方式把数据存储在远端的服务器上，可以让用户随时随地的对数据进行增、删、改、查的操作，而保存这些数据的服务器就是数据库服务器，当然数据库的种类现在也非常多，有基于Key-Value数据类型的Redis数据库，有基于分布式文件存储的数据库MongoDB等，而本节讲述的是目前被广泛使用的关系型数据库MySQL，该数据库和Apache服务器一样，是开源、免费的软件，但他同样有着悠久的历史，并被一次次在大型系统中证明其高度的稳定和快速的性能，至于阿里巴巴在2016年8月9日云栖大会北京峰会上发布了基于MySQL开发了一个全新的分之AliSQL，用于进一步提升数据服务器的运营能力，MySQL的图标如图2-7所示。



图2-7 MySQL的图标

### 2.2.1 Ubuntu下MySQL服务器的安装

在Ubuntu下安装可以采用和Apache相同的方法，因为系统默认的软件源已经添加了MySQL的软件源，但该软件源中的MySQL并不是最新版的，因此本节采用从MySQL官网下载最新的软件源更新包，来更新系统自带的软件源，然后在通过命令行的方式进行安装。

进入MySQL软件源下载页面（http://dev.mysql.com/downloads/repo/apt/），在页面的最下方就是软件源的更新文件，如图2-8所示。要下载该软件源更新包需要先在MySQL官网进行注册账号，注册完成后就可以下载。

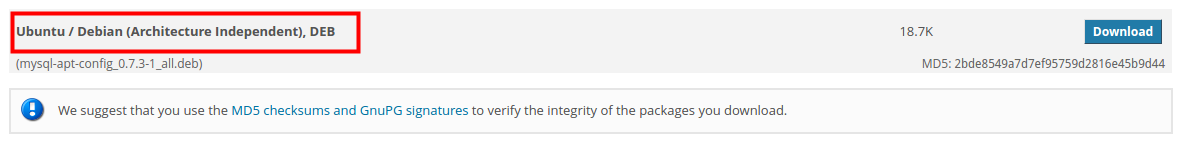


图2-8 MySQL源更新文件

下载下来后的文件是以.deb结尾的文件，该后缀名结尾的文件是Ubuntu下标准的安装包，类似于Windows下的.msi,因此只需要执行双击该文件就可以进行安装，安装的过程非常简单，只需要根据提示完成就可以，在安装过程中会弹出一个对话框用于设置MySQL的Repository，如图2-9所示，只需要在下拉列表中选择OK，然后根据提示点击前进就可以完成。当配置完成完后只需要在终端中执行更新软件源的指令，然后执行安装MySQL Server的指令就可以完成整个最新版MySQL数据库的安装，在MySQL数据库的安装过程中会要求输入MySQL的登陆密码，而用户名是默认的，为root，如图2-10所示，具体安装指令如下。

1 sudo apt-get update

2 sudo apt-get install mysql-server

安装完成后通过下面的三条终端指令可以检查MySQL Server的状态，同时还可以停止MySQL Server和启动MySQL Server，具体指令如下。

1 # 检查MySQL Server的状态

2 sudo service mysql status

3 # 停止MySQL Server

4 sudo service mysql stop

5 # 启动MySQL Server

6 sudo service mysql start

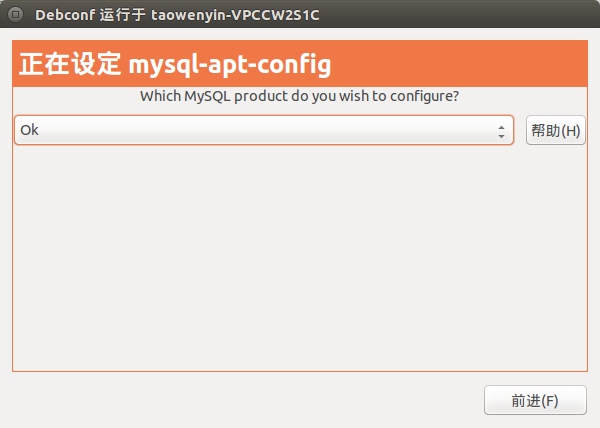


图2-9 MySQL源更新配置

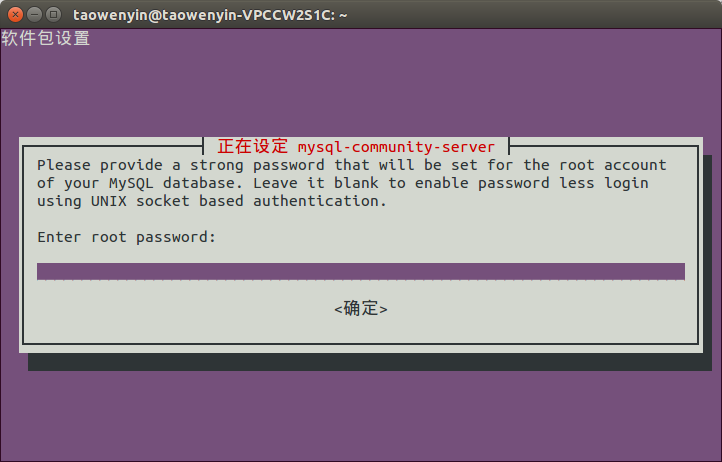


图2-10 MySQL登陆密码设置

### 2.2.2 MySQL管理工具MySQL Workbench的安装和使用

MySQL安装完成后，就可以在其中创建数据库、数据表，以及数据，当目前所有的操作都是基于命令行的方式，虽然这种方式很重要，熟练运用SQL语言是每个程序开发人员必备的素质，但是在实际的运营中，采用命令行的方式其效率就相对较低，因此需要一种GUI的方式来管理数据库，常用数据库管理软件有Navicat for MySQL和phpMyAdmin，其中Navicat for MySQL的性能和跨平台性确实能够很好的提高运营的效率，但是由于是收费软件，因此本文也不推荐，另外一款是phpMyAdmin是一款基于Web的MySQL管理工具，同时也是免费的，有兴趣的读者也可以尝试，而本文推荐的则是MySQL自己开发一款管理工具——MySQL Workbench，这款工具兼具了免费、高性能特点，可以和MySQL无缝对接。

MySQL Workbench的安装一定是在前面已经更新了软件源的基础上进行安装，虽然默认软件源中的包含了该软件，但是版本相对较老。MySQL Workbench的安装和MySQL Server的安装没有什么区别，也只需要在终端中输入下面的指令就可以完成。

1 sudo apt-get install mysql-workbench-community

安装完成，在软件列表就可以找MySQL Workbench的图标，如图2-11所示。点击该图标就可以打开该软件。首先进入MySQL Workbench的登陆界面，如图2-12所示。图中的标记1是系统自动创建的连接本地MySQL Server的快捷方式，但用户点击时就会弹出标记2的对话框，用于输入登陆的密码，当密码正确时就会进入MySQL的管理界面，此外标记3用于创建数据库的E-R图，同时还能通过E-R生成对应的SQL文件。当用户通过MySQL Workbench登陆数据库成功后，就可以看到图2-13的所示的主界面，在该界面中标记1是标签栏，用于切换不同的MySQL Server，标记2则是工具栏，可以用来打开数据库的SQL文件、创建数据库等功能，标记3用来查看该MySQL Server中包含哪些数据库、数据表，而标记4则是用于执行SQL语句，最后的标记5则显示查询结果的集合。



图2-11 MySQL Workbench的图标

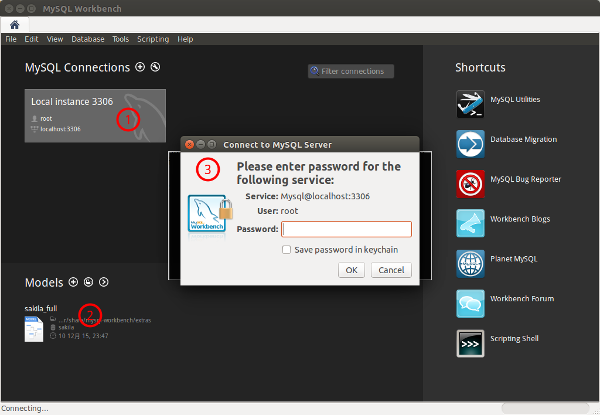


图2-12 MySQL Workbench的登陆界面

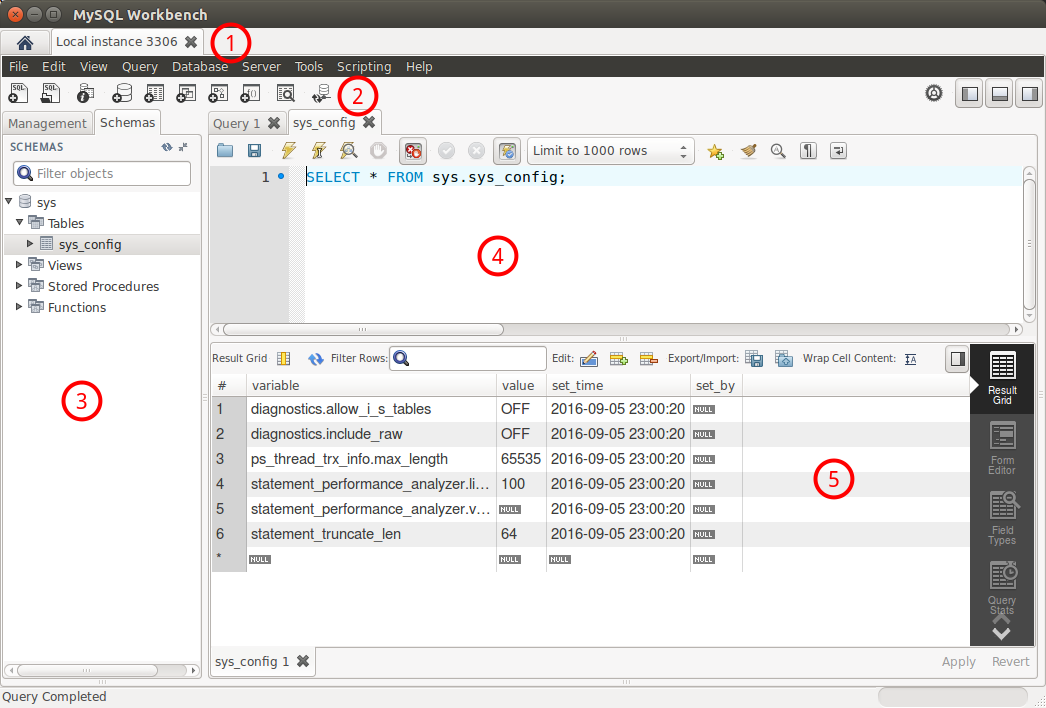


图2-13 MySQL Workbench的主界面

## 2.3 PHP运行库的安装和扩展

在讲如何安装PHP运行库之前先给各位读者讲讲为什么本书会选择以PHP为主要的编程语言，其实要实现动态网站，能用的解决方案还是非常多的，比如Python、Node.js、Ruby、Java、C#等都是非常好的选择，有的甚至在某些性能方便还超过了PHP，但是自从PHP在1994年被发明以来，经过了BBC、Facebook、百度、淘宝等大型网站的应用都充分证明的PHP的可靠性和稳定性，此外虽说PHP已经有20多年的历史，但是老而弥坚，2016年1月最新发布7.0版本性能是原先的一倍多，可见PHP团队的坚持和坚守，此外在20多年的历史中全世界又有数百万开发人员在利用PHP进行系统的开发，也因此产生了无数的第三方的标准库，供后来的开发人员使用，俗话说“大树底下好乘凉”，这句话用在PHP上面最为贴切，因为目前PHP所能解决的问题已经可以覆盖了大部分Web应用的需求，此外PHP的语法相对比较简单，且完全支持面向对象，经过这么多年其学习资料也相对丰富，使用的场景也非常广泛，因此除了可以应用于实际的项目开发中外，也是非常好的学习对象。

此外PHP其自身的特别也促使了PHP能否在20多年的发展中被世界所接受。首先，PHP对于数据库的支持非常好，能够被应用于几乎全部的数据库，所以可以很方便进行系统的部署。其次，PHP自身也是开源的软件,可以免费的获取并应用于商业应用，而不需要额外在付费,减低了开发的成本。最后，PHP是运行在Web Server上的脚本语言，只要服务器上安装的PHP脚本解析器就可以在任何平台上进行使用，如Linux、Windows等都可以运行，当然从目前的应用来说还是Linux上更多写。所以基于上面这些原因本书选择以PHP为主要开发语言，PHP的图标如图2-14所示。

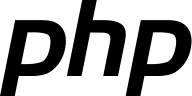


图2-14 PHP的图标

### 2.3.1 Ubuntu下的PHP运行库的安装

PHP在作者执笔时，最新的版本是5.6，此时有读者会到官网上发现有7.X的版本，此版本和5.X是完全不同的版本，相互并不兼容，但是目前从使用情况来说还是5.X使用的更为广泛，因此采用5.6的版本，而如果使用系统默认的软件源安装的话，PHP的版本只有5.5，但是本书力求最新，因此使用和MySQL类似的策略，通过添加第三方软件源的方法来安装最新的PHP，要安装最新的PHP软件源只需要在终端中执行下面两条指令就可以完成软件源的添加和更新。

1 sudo add-apt-repository ppa:ondrej/php

2 sudo apt-get update

当PHP的软件更新完成后，需要在终端里输入一条指令，并根据提示操作就可以完成PHP运行环境的安装，同时还会为Apache Server添加PHP模块，因为默认情况Apache Server是不包含PHP模块，也就无法把.php文件交给PHP解析器进行解析，当安装完服务器再在终端中执行php -v，如果能返回PHP版本，如果版本号为5.6.X就说明安装成功，如图2-15所示，具体指令如下。

1 sudo apt-get install php5.6

2 php -v

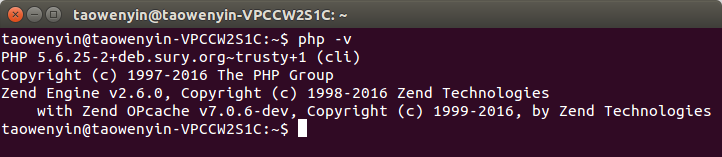


图2-14 PHP安装验证

### 2.3.2 常用的PHP运行库扩展程序

PHP的运行库扩展的目的是为了让PHP具有更加丰富的功能，在默认情况下安装的PHP只包含PHP的内核，即让服务器能够读取和解析PHP文件，但是此时的PHP连数据库都无法链接和读取，因为此时的PHP还不包括数据库的功能，因此需要为其添加扩展，而PHP功能的扩展采用插件式的方式，通过基于PHP内核的开发来完成，所以要让PHP能较为完整的服务于应用开发，那么就需要通过安装PHP扩展来完成。常见的扩展包括以下五种：

1、MySQL数据库扩展的安装。该扩展的作用顾名思义就是帮助PHP能够对MySQL数据库进行增、删、改、查的操作，具体指令如下：

1 sudo apt-get install php5.6-mysql

2、字符串多编码扩展的安装。默认情况下PHP内部所支持的文本编码方式只有ISO-8859-\*、EUC-JP、UTF-8，而采用其他编码方式的文本都没办法在PHP程序上正确显示，因此通过安装此编码让PHP更多的文本编码方式，如简体中文、繁体中文等，具体指令如下：

1 sudo apt-get install php5.6-mbstring

3、加密和解密扩展的安装。在编写代码程序时，除了要保证代码的高性能之外，还有一点是非常重要的，那就是程序的安全性。除了PHP自带的几种加密函数外，还可以通过安装PHP加密扩展库Mcrypt来获取额外20多种的加密和解密的算法，具体指令如下：

1 sudo apt-get install php5.6-mcrypt

4、图形扩展的安装。在使用论坛时很多时候需要输入验证码，单这个验证码是随机的，并不是实现安排好的图片，并且每张验证码的图片均不相同，即使内容相同，但背景或字母摆放的位置也不尽相同，此时就需要使用图形库来进行随机的绘制和显示，具体指令如下：

1 sudo apt-get install php5.6-gd

5、HTTP通信扩展的安装。在用PHP开发系统时，特别是微信系统，很多时候都需要使用HTTP和微信接口进行通信，但是默认情况下PHP是不支持HTTP通信，因此需要安装CURL扩展来帮助PHP实现HTTP通信，具体指令如下：

1 sudo apt-get install php5.6-curl

## 2.4 小结

通过本章的学习，了解了动态网站通信的基本过程，了解了常用LAMP系统的几个组成部分，同时还掌握了在Ubuntu系统下Apache服务器的安装和简单配置，及使用方法，此外掌握了如何在Ubuntu系统下安装MySQL数据库服务器，以及图形化数据库管理器MySQL Workbench的基本使用，最后还掌握了Ubuntu系统下PHP运行库的安装，以及常用扩展组件的作用及安装方法。在学完本章内容后，读者可以自行安装完整的LAMP系统，以供后续内容的使用。

# 第三章 几种常用的Web开发工具

古人有云：“工欲善其事，必先利其器”。只有装备了高等级的武器，那么上阵杀敌才能又快又准。学习PHP编程也是这样，虽然使用系统自带的记事本也可以进行编程，但是效率非常低，而且容易出错。所以本章将介绍几款常用PHP程序编辑和调试软件，熟练掌握这些软件的使用可以大幅提高编写程序的效率。

本章内容

* XDebug 开源的PHP调试工具的使用
* Vistual Code 开源的跨平台程序编辑器的使用
* Eclipse for PHP Developers 全能的程序编辑器的使用
* Chrome DevTools 实用的HTML、CSS、JavaScript调试工具的使用

## 3.1 XDebug开源的PHP调试工具

在生活中如果做错了事可能经常是没办法挽回的，就像泼出去的水一样，但是在程序的世界就不一样了，我们可以通过调试工具来反复的验证我们的程序，直到它没有问题为止，而PHP的调试工具就是本节要讲述的XDebug。XDebug是一个开放源代码的PHP程序调试工具，可以用来跟踪，调试和分析PHP程序。要安装XDebug如果采用从源码编译的方式相对较为复杂，而且对于PHP开发人员来说也没有必要了解XDebug的编译方式和源码内容，因此本节采用通过命令行的方式来安装XDebug，XDebug的图标如图3-1所示。



图3-1 XDebug的图标

### 3.1.1 Linux下XDebug的安装

XDebug的安装不同于其他应用软件的安装可以通过APT进行，因为XDebug并不是一个应用程序，而只是PHP的扩展程序，因此就需要使用安装PHP扩库社区库的方式来进行安装。PHP扩库社区库的英文全程叫The PHP Extension Community Library（简称PECL），通过PECL可以很方便的安装XDebug。要安装PECL其实非常的简单，由于系统的软件源中已经包含了PECL，因此只需要执行APT执行就可以完成，当安装完成后在执行PECL执行安装XDebug就可以完成整个XDebug的安装，具体指令如下。

1 sudo apt-get install php-pear

2 sudo pecl install xdebug

但是在通过PECL安装XDebug时可能会出现错误，如图3-2所示。该错误的意思是因为缺少phpize这个程序，而phpize默认是不会在安装PHP的时候被安装的，因此需要单独安装该程序，phpize被包含在PHP开发版里面，所以需要通过下面指令安装PHP开发版即可同步安装phpize，然后再重新通过pecl重新安装XDebug就可以正常完成整个XDebug的安装，具体指令如下。

1 sudo apt-get install php5.6-dev

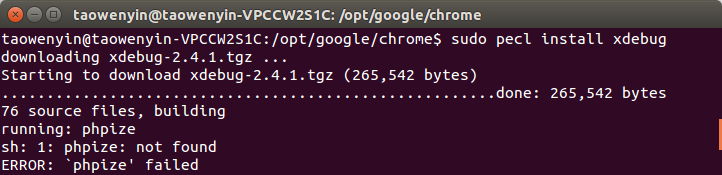


图3-2 XDebug的安装错误

当XDebug安装完成后，PECL会自动把XDebug的扩展文件xdebug.so放在/usr/lib/php/20131226目录下，但是此时PHP还无法使用XDebug，需要把扩展文件的路径配置到php.ini文件中，这里需要注意的是php.ini所在的目录是etc下，默认情况下普通用户是无法进行修改的，因此需要在用编辑工具修改php.ini文件的同时使用管理员权限打开该文件，具体执行如下：

1 cd /etc/php/5.6/apache2

2 sudo gedit php.ini

打开php.ini配置文件后，在文件的最下面添加如下XDebug的配置信息。

1 [Xdebug]

2 zend\_extension="/usr/lib/php/20131226/xdebug.so"

此外还要在该配置信息中再添加七个常用的配置信息，通过这些配置信息的添加可以增强XDebug调试PHP的性能，具体如下：

1、允许XDebug跟踪函数调用，具体配置信息如下：

1 xdebug.auto\_trace=1

2、允许XDebug跟踪函数参数，具体配置信息如下：

1 xdebug.collect\_params=1

3、允许XDebug跟踪函数返回值，具体配置信息如下：

1 xdebug.collect\_return=1

4、允许XDebug跟踪函数调用的信息输出文件目录，具体配置信息如下：

1 xdebug.trace\_output\_dir="自己指定输出目录"

5、打开XDebug的性能分析器，具体配置信息如下：

1 xdebug.profiler\_enable=1

6、XDebug的性能分析器的分析文件的存放位置

1 xdebug.profiler\_output\_dir="自己指定输出目录"

## 3.2 Vistual Code开源的跨平台程序编辑器

2016年3月中旬Vistual Code（以下简称VSCode）发布了正式版本1.0，笔者用过很多的代码编辑软件，例如VIM、Emacs、Sublime Text、Atom等，但使用一圈下来从实用性、方便性、性能等方面来说笔者还是更加喜欢VSCode。VSCode是微软发布第一款能够运行在Windows、Linux、MAC上面的跨平台的程序编辑器，在过去微软的另一款集成开发环境（以下简称IDE）Vistual Studio只能运行在Windows，而无法在其他操作系统中运行，而现在微软把代码编辑功能从Vistual Studio中剥离，进而开发了VSCode，VSCode支持几乎所有的语言，并且可以通过的使用可以很方便的实现代码提示，VSCode的图标如图3-1所示。

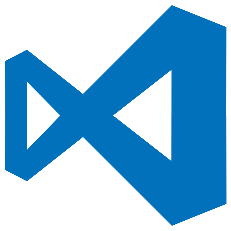


图3-1 VSCode的图标

### 3.2.1 Linux下Vistual Code的安装

进入VSCode的官网（https://code.visualstudio.com/），其会自动的识别当前系统的类型，例如笔者所使用的是Ubuntu，那么网站就会提供2中安装包，一个是基于Fedora的RPM包，还有一种是基于Ubuntu的DEB包，因此这里只需要把DEB包下载下来，并且执行下面的命令或者直接双击就就可以进行安装，如果使用命令行安装那么需要把package.deb换成下载后的文件名，如图3-2所示。当安装完成后就可以在系统的程序目录下找到Vistual Code。

1 sudo dpkg -i package.deb



图3-2 执行命令行安装VSCode

### 3.2.2 Vistual Code中PHP相关插件的安装

VSCode之所以能在极短的时间内在GitHub上获得17000多Star，同时也得到了全世界程序员的认可，其中一个很重要的原因就是VSCode可以通过插件扩展的方式来支持任何的语言，并且每天有无数的程序员在为其添加新的功能，大到微软自己有独立的团队编写插件，小到有人专门为VSCode编写文件图标的插件，甚至国内的游戏制作团队Cocos也在为其编写开发插件，以适应Coscos2D-JS的开发。

目前原生的不安装任何插件的VSCode已经为开发人员提供了如代码高亮、括号匹配、代码智能提示，以及代码片段等功能，已经可以满足基本的开发需求，此外还可以通过插件扩展更多的功能。在这里本书推荐2个插件，如图3-3所示。一个是PHP Debug，可以让VSCode实现PHP代码的调试，另外一个是Crane-PHP Intellisense，可以让VSCode实现整个项目的代码智能感知，从提供更强大的代码提示。

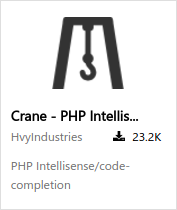
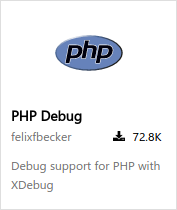


图3-3 两个PHP开发常用的插件

**1、PHP Debug插件的安装与使用**

在Vistual Code中安装PHP Debug插件是非常简单的一件事，只需要通过以下八个步骤就可以完成。

第一步：保证你的电脑处于联网状态，这个非常重要，因为插件的安装其实是从网络上下载，再有VSCode自动安装完成。

第二步：在VSCode的配置文件中添加PHP的可执行路径，因为只有这样PHP Debug才能把PHP和XDebug进行链接，代码如下：

1 // 指向可执行的 php。

2 "php.validate.executablePath": "/usr/bin/php"

第三步：安装XDebug，这个在本章的最后一节会进行详细的介绍，是真正用于PHP调试的工具，而我们的PHP Debug插件只是在VSCode和XDebug之间以建立一个沟通的桥梁。

第四步：在PHP的配置文件php.ini中添加XDebug的配置项，代码如下。其中zend\_extension后面的地址需要读者根据自己的安装目录进行修改。

1 zend\_extension=path/to/xdebug

第五步：在PHP的配置文件php.ini中打开PHP远程调试，配置代码如下所示。

1 [XDebug]

2 xdebug.remote\_enable = 1

3 xdebug.remote\_autostart = 1

第六步：重启你的Web服务器，如果你和笔者一样使用的是Linux操作系统，可以在终端中使用下面的代码进行服务器的重启。

1 service httpd restart

第七步：通过phpinfo()打印来验证你的配置是否正确，如果在打印结果的最下方出现如图3-4所示，那么就表示配置成功。

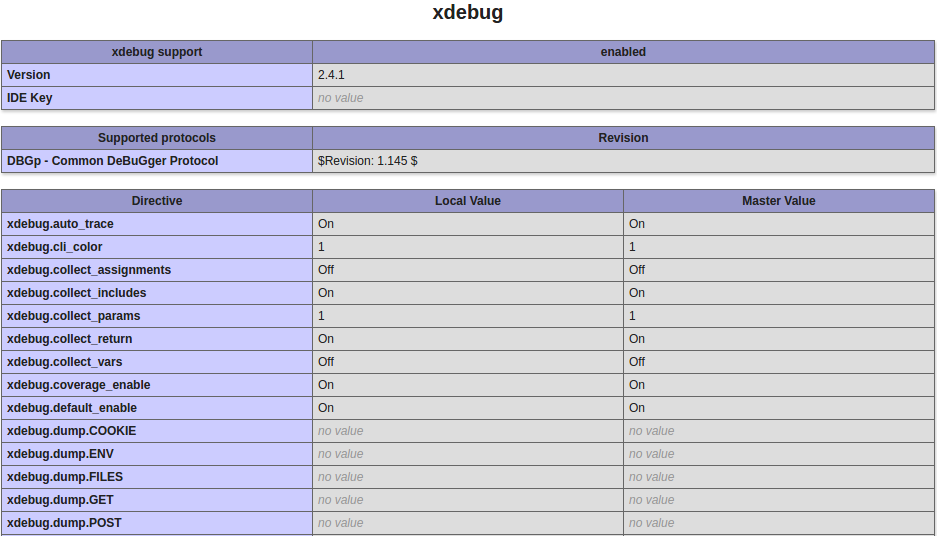


图3-4 XDebug的配置信息

第八步：这一步非常重要，用于安装VSCode插件，不过也非常的简单，只要按下键盘上的F1建，并在顶部的输入框中输入如下代码，等待几分钟，就可以完成了整个PHP Debug插件的安装，接下来就可以通过左侧导航栏中的“调试”界面进行代码的调试。

1 ext install php-debug

**2、Crane-PHP Intellisense插件的安装与使用**

和安装安装PHP Debug插件一样，Crane-PHP Intellisense插件的安装也是非常简单的一件事，只需要通过一下七个步骤就可以完成。

第一步：在键盘上按下F1建，并在顶部的输入框中输入如下代码，等待几分钟，就可以完成了整个Crane-PHP Intellisense插件的安装。

1 ext install crane

第二步：使用VSCode打开一个PHP的项目。

第三步：检查右下角的状态栏是否有Crane状态，此时可以看到Crane检索的所有PHP文件，此时当调用类中方法时就会自动弹出方法名称，如图3-5所示。

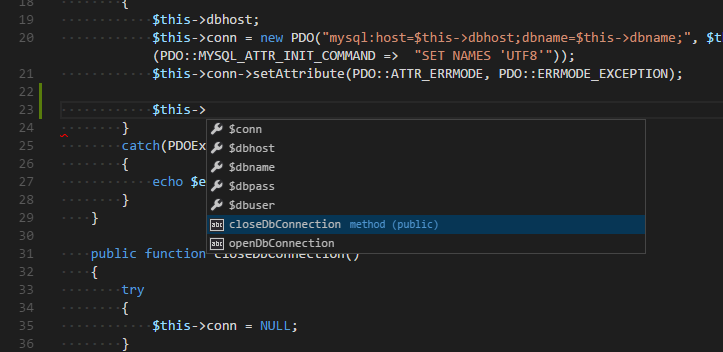


图3-5 Crane的使用

## 3.3 Eclipse for PHP Developers全能的程序编辑器

Eclipse和VSCode类似，也是一个基于插件的模式，且开放源代码程序编辑器，但由于其包含了软件开发所有需要几乎所有的功能，因此通常它为集成开发环境（简称：IDE）。Eclipse最初是在1999年由IBM发起，并和多家公司共同完成的用于开发另外一种编程语言——Java的工具，在当时市面上没有比较好的Java开发工具，相对来说只有Borland公司的JBuilder比较好用，但由于价格昂贵，不是一般程序员能够使用的，因此IBM在2001年的时候把Eclipse彻底开源，同时把该项目捐给了Eclipse基金会，用现在流行的语言说，Eclipse就是当时的爆款，而且是全球的爆款，其免费、开源、IBM做背书、由Java代码编写、平台的一致性等特点彻底俘获了所有程序员的心，目前Eclipse最新版为Eclipse Neon，其图标如图3-6所示。



图3-6 Eclipse的图标

随着Eclips的发展，已由原先开发Java发展到现在几乎可以开发任何的语言，如果说Vistual Studio是Windows平台下最好的集成开发环境，那么Eclipse就是在跨平台领域下最好的集成开发环境。不过和Vistual Studio一样，由于其功能非常的丰富，因此体积也越发的臃肿，其中许多的功能对于一个初学者来说还暂时用不到。在IBM开发Eclips时把Eclipse定义为一个程序框架，通过基于该框架的开发可以把Eclipse扩展到任意的编程语言，有的公司甚至基于Eclipse开发自己的IDE，如PHP中著名的公司Zend，该公司就是基于Eclipse开发了自己的集成开发环境Zend Studio，而我们这里所说的Eclipse for PHP Developers也是基于Eclipse开发的PHP插件，使得Eclipse可以支持PHP的开发，与Zend Studio不同的是Eclipse for PHP Developers是免费的，而且功能也能满足一般开发的需求。

### 3.3.1 Linux下Oracle Java的安装

由于Eclipse的使用需要依赖于Java，但Linux系统默认没有安装Java环境，因此在安装Eclipse for PHP Developers之前首先要把Java环境安装好。因为笔者使用的是目前非常流行的Linux发行版——Ubuntu系统，所以可以通过以下五个步骤就可以完成。

第一步：保证你的电脑处于联网状态，这个非常重要，因为软件的安装其实是从网络上下载，再由系统的软件管理器进行自动安装完成。

第二步：添加Oracle Java软件源。因为Ubuntu默认不包含Oracle Java的软件源，所以需要执行下面的命令来添加对应的软件源，结果如图3-7所示。

1 sudo add-apt-repository ppa:webupd8team/java

2 sudo apt-get update

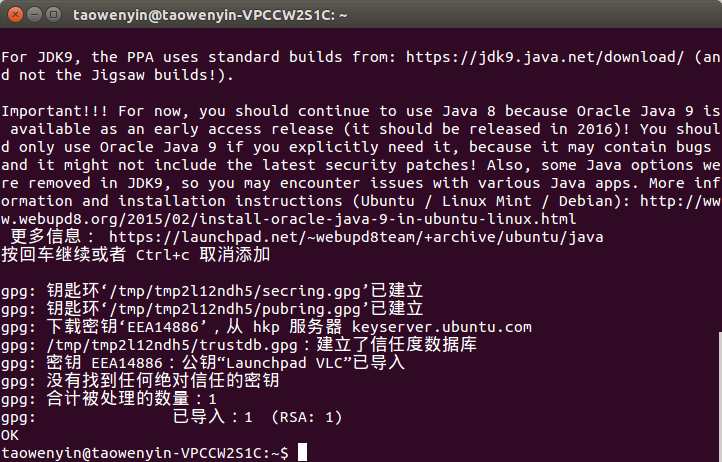


图3-7 Oracle Java软件源的添加

第三步：安装Oracle Java。添加完软件源后，安装Java是非常简单的一件事，只需要执行下面的命令就可以完成Oracle Java的下载与安装。

1 sudo apt-get install oracle-java8-installer

第四步：执行下面的命令把Java8设为默认Java环境

1 sudo apt-get install oracle-java8-set-default

第五步：在终端中输入java或javac，并且打印如图3-8所示的内容，则表示Java环境已经安装完成。

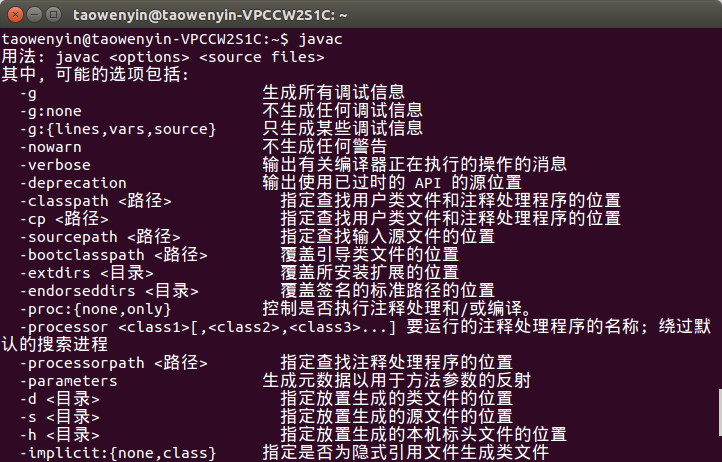
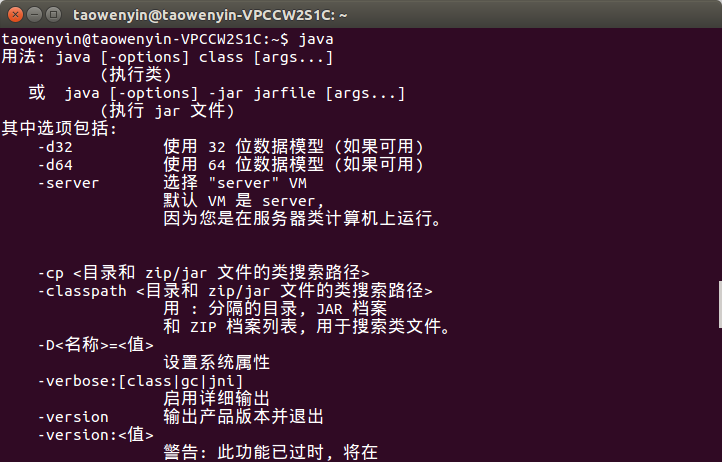


图3-8 验证Java安装完成

### 3.3.2 Linux下Eclipse for PHP Developers的安装

当进入Eclipse的官网（https://www.eclipse.org/downloads/），其会自动的识别当前系统的类型，例如笔者所使用的是Ubuntu，那么网站就会提供一个压缩包，在Neno版之前Eclipse官网都会直接提供针对不同开发语言的Eclipse程序件，但到了最新的Neno版，官网除了原先提供的内容外，还提供提供了一个集中的管理程序，本文就是通过这个管理程序来进行Eclipse for PHP Developers的安装，只需要以下几步就可以完成。

第一步：解压缩安装包eclipse-inst-linux64.tar.gz，并且双击执行eclipse-inst文件，弹出Eclipse安装包类型选择界面，如图3-9所示。

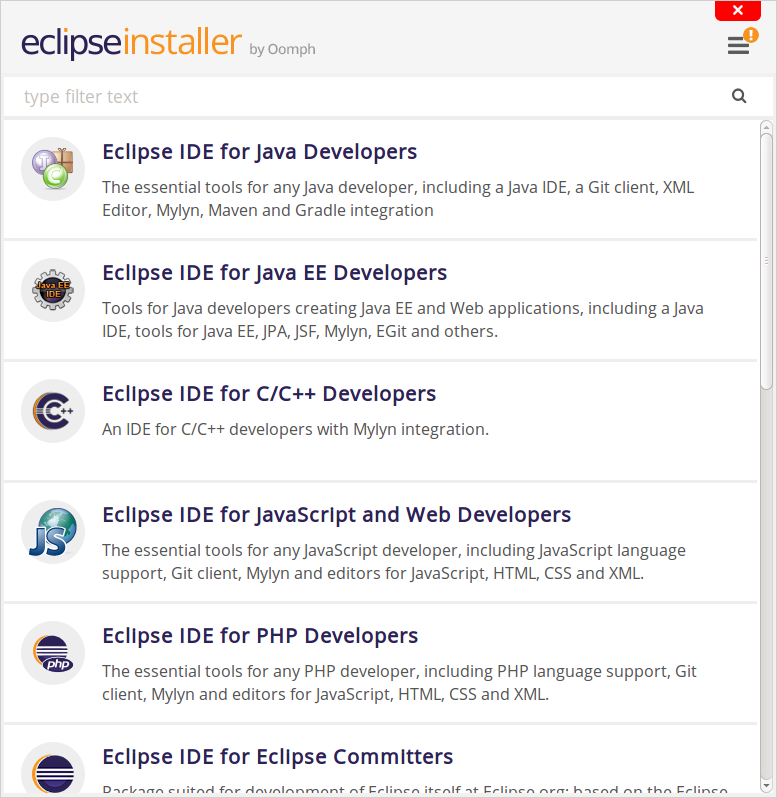


图3-9 Eclipse安装包类型选择界面

第二步：选择Eclipse for PHP Developers，设置安装路径，点击INSTALL进行安装，如图3-10所示。

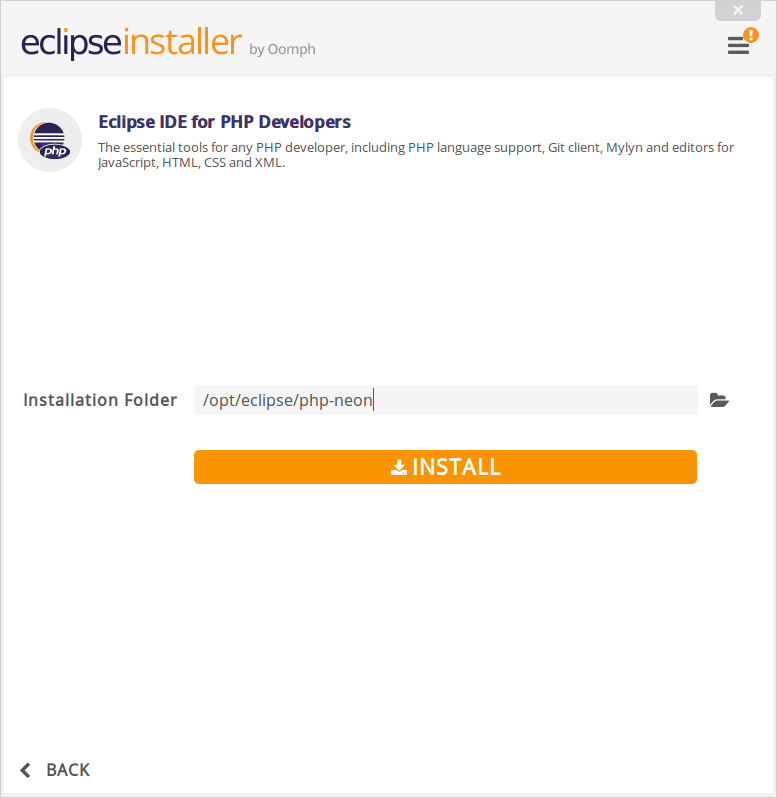


图3-10 安装路径选择界面

第三步：等待安装，当安装完成后，点击LAUNCH运行Eclipse for PHP Developers，如图3-11所示。

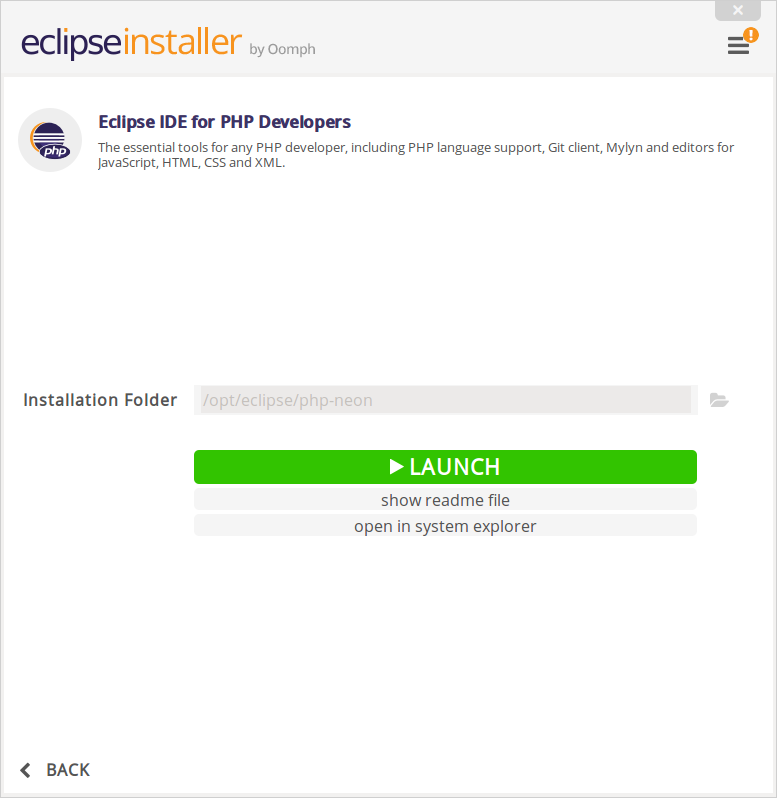


图3-11 安装完成择界面

第四步：设置Eclipse for PHP Developers的工作目录，这里需要注意的是要把Eclipse的工作目录设置在WebServer的www目录下，方便执行执行PHP程序，同时如果该工作目录以后不会再改动，那么可以把“Use this as the default and do not ask again”打勾，以后就不会弹出该界面，如图3-12所示。

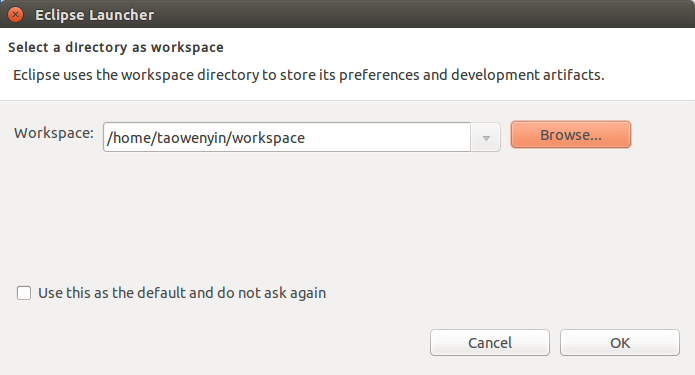


图3-12 工作目录配置

第五步：当设置完工作目录后，点击OK键就可以进入系统，此时可以通过点击菜单栏中的“File”->“New”->“PHP Project”创建PHP项目，如图3-13所示。

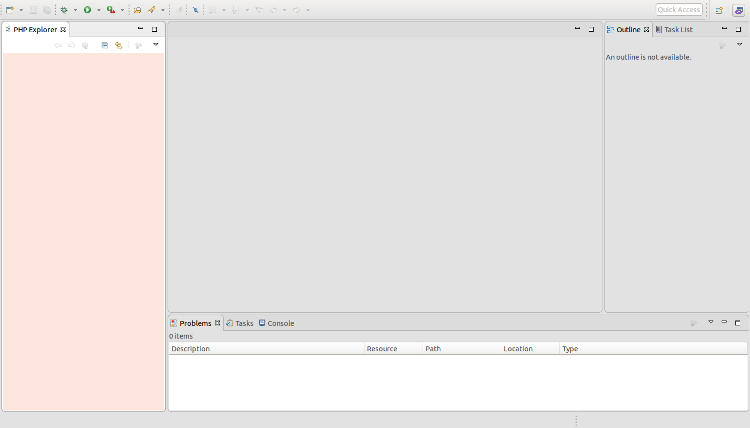


图3-13 Eclipse运行界面

## 3.4 Chrome DevTools实用的前端调试工具

以前听前辈们讲过这么一句话，世界上只有4种浏览器，分别是Trident、Webkit、Gecko、Presto，其实这些指的都不是浏览器，而是浏览器的内核，随着最新的Win10发布，在浏览器内核家族里又加入了Edge，但是除了Webkit之外其他内核都不是开源的，因此都只有个别浏览器在使用，而Webkit的代表浏览器就是Google的Chrome，同时该内核也是由Google开发的，随着Webkit内核的日渐成熟，国内的浏览器市场也是百花齐放，如360、搜狗、猎豹、QQ等知名厂商都采用了Webkit作为其浏览器的内核，因此可以说Webkit占据了国内浏览器的半壁江山，因此使用Webkit调试网页是可以说能代表最广大人民的利益，也因为这个原因本书的Web调试工具采用Webkit内置的调试工具Chrome DevTools。要使用Webkit内核的浏览器在Windows上有很多原则，如果360安全浏览器、猎豹等，但是在Linux上就只有Chrome这一个选择，因此在讲解Chrome DevTools之前首先要执行下面的指令来进行把Chrome下载，然后双击安装即可。

1 wget https://dl.google.com/linux/direct/google-chrome-stable\_current\_amd64.deb

安装好Chrome后打开Chrome DevTools有两种方式，一种是打开顶部菜单的工具，然后选择开发者工具，或者使用快捷键键Ctrl+Shift+I，打开后可以看到如图3-14的界面，一共有九大功能模块，其中最为常用的有六个，分别功能如下：

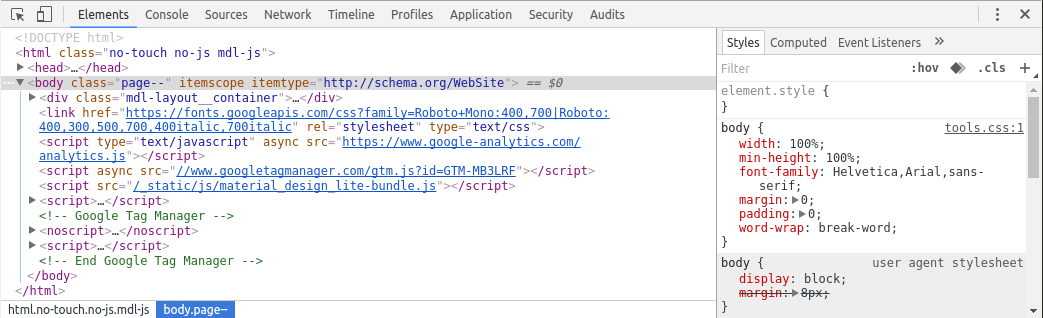


图3-14 Chrome DevTools运行界面

1、Elements：可以查看当前网页对应的HTML代码，通过在在该页面中编辑和修改页面元素的属性达到调试页面的效果，同时页面也会跟随代码的改变而改变，做到所见即所得的效果。

2、Console: 控制台工具，可以显示各种警告与错误信息，同时开发人员也可以通过该窗口进行信息打印，从而获取程序运行中的中间值。

3、Sources: 该页面主要用来对页面JavaScript的代码进行调试，该页面中可以实现对代码的暂停、单步调试、进入函数、跳出函数，以及查看临时和全局变量等。

4、Network：通过该页面可以查看当前页面从服务器端请求了哪些资源，这些资源的大小，以及所花费的时间，同时还能看到哪些资源是加载失败的，利用这些信息可以知道页面优化的方向。此外，该页面还可以查看HTTP的请求头和返回内容等。

5、Timeline：计算页面在从请求到展示整个过程中，每个行为所消耗的时间，例如计算CSS样式所需要的时间和执行JavaScript所花费的时间等。

6、Profiles：该页面是从整体上分析页面所消耗的时间，以及所占用的内存情况。

## 3.5 小结

通过本章的学习，了解Eclipse for PHP Developers开发工具的基本使用方法，同时也了解在进行Web应用开发是可以通过Chrome DevTools来进行页面的调试，此外掌握了如何在Ubuntu下进行XDebug的安装，以及基本配置方法，最后还掌握了如何在Ubuntu下安装VSCode，以及VSCode中插件的安装方法和两款用于调试和编辑PHP代码的插件，从而提高整个开发的效率。在学习完本章内容之后，读者可以自行安装XDebug、VSCode以及对应的插件，以供后续内容的使用。