Netkiller Developer 手札

目录

范例清单

Netkiller Developer 手札

November 15, 2008

ISBN#

Mr. Neo Chan, 陈景峯(BG7NYT)

中国广东省深圳市望海路半岛城邦三期
518067
+86 13113668890

<netkiller@msn.com>

电子书最近一次更新于 2022-01-08 14:05:33

版权声明

http://www.netkiller.cn
http://netkiller.github.io
http://netkiller.sourceforge.net
微信公众号: netkiller
微信：13113668890 请注明“读者”
QQ：13721218 请注明“读者”
QQ群：128659835 请注明“读者”
知乎专栏 | 多维度架构

2017-02-13

我的系列文档

编程语言

Netkiller Architect 手札	Netkiller Developer 手札	Netkiller Java 手札	Netkiller Spring 手札	Netkiller PHP 手札	Netkiller Python 手札
Netkiller Testing 手札	Netkiller Cryptography 手札	Netkiller Perl 手札	Netkiller Docbook 手札	Netkiller Project 手札	Netkiller Database 手札

致读者

Netkiller 系列手札 已经被 Github 收录，并备份保存在北极地下250米深的代码库中，备份会保留1000年。

Preserving open source software for future generations

The world is powered by open source software. It is a hidden cornerstone of modern civilization, and the shared heritage of all humanity.

The GitHub Arctic Code Vault is a data repository preserved in the Arctic World Archive (AWA), a very-long-term archival facility 250 meters deep in the permafrost of an Arctic mountain.

We are collaborating with the Bodleian Library in Oxford, the Bibliotheca Alexandrina in Egypt, and Stanford Libraries in California to store copies of 17,000 of GitHub's most popular and most-depended-upon projects—open source's “greatest hits”—in their archives, in museum-quality cases, to preserve them for future generations.

https://archiveprogram.github.com/arctic-vault/

自述

《Netkiller 系列 手札》是一套免费系列电子书，netkiller 是 nickname 从1999 开使用至今，“手札” 是札记，手册的含义。

2003年之前我还是以文章形式在BBS上发表各类技术文章，后来发现文章不够系统，便尝试写长篇技术文章加上章节目录等等。随着内容增加，不断修订，开始发布第一版，第二版......

IT知识变化非常快，而且具有时效性，这样发布非常混乱，经常有读者发现第一版例子已经过时，但他不知道我已经发布第二版。

我便有一种想法，始终维护一个文档，不断更新，使他保持较新的版本不过时。

第一部电子书是《PostgreSQL 实用实例参考》开始我使用 Microsoft Office Word 慢慢随着文档尺寸增加 Word 开始表现出力不从心。

我看到PostgreSQL 中文手册使用SGML编写文档，便开始学习 Docbook SGML。使用Docbook写的第一部电子书是《Netkiller Postfix Integrated Solution》这是Netkiller 系列手札的原型。

至于“手札”一词的来历，是因为我爱好摄影，经常去一个台湾摄影网站，名字就叫“摄影家手札”。

由于硬盘损坏数据丢失 《Netkiller Postfix Integrated Solution》 的 SGML文件已经不存在；Docbook SGML存在很多缺陷 UTF-8支持不好，转而使用Docbook XML.

目前技术书籍的价格一路飙升，动则￥80，￥100，少则￥50，￥60. 技术书籍有时效性，随着技术的革新或淘汰，大批书记成为废纸垃

圾。并且这些书技术内容雷同，相互抄袭，质量越来越差，甚至里面给出的例子错误百出，只能购买影印版，或者翻译的版本。

在这种背景下我便萌生了自己写书的想法，资料主要来源是我的笔记与例子。我并不想出版，只为分享，所有我制作了基于CC License 发行的系列电子书。

本书注重例子，少理论（捞干货），只要你对着例子一步一步操作，就会成功，会让你有成就感并能坚持学下去，因为很多人遇到障碍就会放弃，其实我就是这种人，只要让他看到希望，就能坚持下去。

1. 写给读者

为什么写这篇文章

有很多想法,工作中也用不到所以未能实现，所以想写出来,和大家分享.有一点写一点,写得也不好,只要能看懂就行,就当学习笔记了.

开始零零碎碎写过一些文档，也向维基百科供过稿，但维基经常被ZF封锁，后来发现sf.net可以提供主机存放文档，便做了迁移。并开始了我的写作生涯。

这篇文档是作者20年来对工作的总结,是作者一点一滴的积累起来的，有些笔记已经丢失，所以并不完整。

因为工作太忙整理比较缓慢。目前的工作涉及面比较窄所以新文档比较少。

我现在花在技术上的时间越来越少，兴趣转向摄影，无线电。也想写写摄影方面的心得体会。

写作动力:

曾经在网上看到外国开源界对中国的评价，中国人对开源索取无度，但贡献却微乎其微.这句话一直记在我心中，发誓要为中国开源事业做我仅有的一点微薄贡献

另外写文档也是知识积累，还可以增加在圈内的影响力.

人跟动物的不同,就是人类可以把自己学习的经验教给下一代人.下一代在上一代的基础上再创新,不断积累才有今天.

所以我把自己的经验写出来,可以让经验传承

没有内容的章节:

目前我自己一人维护所有文档，写作时间有限，当我发现一个好主题就会加入到文档中，待我有时间再完善章节，所以你会发现很多章节是空无内容的.

文档目前几乎是流水帐试的写作，维护量很大，先将就着看吧.

我想到哪写到哪,你会发现文章没一个中心,今天这里写点,明天跳过本章写其它的.
文中例子绝对多,对喜欢复制然后粘贴朋友很有用,不用动手写,也省时间.
理论的东西,网上大把,我这里就不写了,需要可以去网上查.
我爱写错别字,还有一些是打错的,如果发现请指正.
文中大部分试验是在Debian/Ubuntu/Redhat AS上完成.

写给读者

至读者:

我不知道什么时候，我不再更新文档或者退出IT行业去从事其他工作，我必须给这些文档找一个归宿，让他能持续更新下去。

我想捐赠给某些基金会继续运转，或者建立一个团队维护它。

我用了20年时间坚持不停地写作，持续更新，才有今天你看到的《Netkiller 手扎》系列文档，在中国能坚持20年，同时没有任何收益的技术类文档，是非常不容易的。

有很多时候想放弃，看到外国读者的支持与国内社区的影响，我坚持了下来。

中国开源事业需要各位参与，不要成为局外人，不要让外国人说:

中国对开源索取无度，贡献却微乎其微。

我们参与内核的开发还比较遥远，但是进个人能力，写一些文档还是可能的。

系列文档

下面是我多年积累下来的经验总结，整理成文档供大家参考:

Netkiller Architect 手札

Netkiller Developer 手札

Netkiller PHP 手札

Netkiller Python 手札

Netkiller Testing 手札

Netkiller Cryptography 手札

Netkiller Linux 手札

Netkiller FreeBSD 手札

Netkiller Shell 手札

Netkiller Security 手札

Netkiller Web 手札

Netkiller Monitoring 手札

Netkiller Storage 手札

Netkiller Mail 手札

Netkiller Docbook 手札

Netkiller Version 手札

Netkiller Database 手札

Netkiller PostgreSQL 手札

Netkiller MySQL 手札

Netkiller NoSQL 手札

Netkiller LDAP 手札

Netkiller Network 手札

Netkiller Cisco IOS 手札

Netkiller H3C 手札

Netkiller Multimedia 手札

Netkiller Management 手札

Netkiller Spring 手札

Netkiller Perl 手札

Netkiller Amateur Radio 手札

2. 作者简介

陈景峯 (ㄔㄣ ㄐㄧㄥ ㄈㄥ)

Nickname： netkiller | English name: Neo chen | Nippon name: ちんけいほう (音訳) | Korean name: 천징봉 | Thailand name: ภูมิภาพภูเขา | Vietnam: Trần Cảnh Phong

Callsign: BG7NYT | QTH: ZONE CQ24 ITU44 ShenZhen, China

程序猿，攻城狮，挨踢民工, Full Stack Developer, UNIX like Evangelist, 业余无线电爱好者（呼号：BG7NYT）,户外运动，山地骑行以及摄影爱好者。

《Netkiller 系列 手札》 的作者

成长阶段

1981年1月19日(庚申年腊月十四)出生于黑龙江省青冈县建设乡双富大队第一小队

1989年9岁随父母迁居至黑龙江省伊春市，悲剧的天朝教育，不知道那门子归定，转学必须降一级，我本应该上一年级，但体制让我上学前班，那年多都10岁了

1995年小学毕业，体制规定借读要交3000两银子(我曾想过不升初中)，亲戚单位分楼告别平房，楼里没有地方放东西，把2麻袋书送给我，无意中发现一本电脑书BASIC语言，我竟然看懂了，对于电脑知识追求一发而不可收，后面顶零花钱，压岁钱主要用来买电脑书《MSDOS 6.22》《新编Unix实用大全》《跟我学Foxbase》。。。。。。

1996年第一次接触UNIX操作系统，BSD UNIX, Microsoft Xinux(盖茨亲自写的微软Unix，知道的人不多)

1997年自学Turbo C语言，苦于没有电脑，后来学校建了微机室才第一次使用QBASIC(DOS 6.22 自带命令)，那个年代只能通过软盘拷贝转播，Trubo C编译器始终没有搞到，

1997年第一次上Internet网速只有9600Bps, 当时全国兴起各种信息港域名格式是www.xxxx.info.net, 访问的第一个网站是NASA下载了很多火星探路者拍回的照片，还有“淞沪”sohu的前身

1998~2000年在哈尔滨学习计算机，充足的上机时间，但老师让我们练打字（明伦五笔/WT）打字不超过80个/每分钟还要强化训练，不过这个给我的键盘功夫打了好底。

1999年学校的电脑终于安装了光驱，在一张工具盘上终于找到了Turbo C, Borland C++与Quick Basic编译器，当时对VGA图形编程非常感兴趣，通过INT33中断控制鼠标，使用绘图函数模仿windows界面。还有操作 UCDOS 中文字库，绘制矢量与点阵字体。

2000年沉迷于Windows NT与Back Office各种技术，神马主域控制器，DHCP，WINS，IIS，域名服务器，Exchange邮件服务器，MS Proxy, NetMeeting...以及ASP+MS SQL开发；用56K猫下载了一张LINUX。ISO镜像，安装后我兴奋的24小时没有睡觉。

职业生涯

2001 年来深圳进城打工,成为一名外来务工者. 在一个4人公司做PHP开发，当时PHP的版本是2.0, 开始使用Linux Redhat 6.2.当时很多门户网站都是用FreeBSD,但很难搞到安装盘，在网易社区认识了一个网友,从广州给我寄了一张光盘，FreeBSD 3.2

2002 年我发现不能埋头苦干,还要学会"做人".后辗转广州工作了半年，考了一个Cisco CCNA认证。回到深圳重新开始，在车公庙找到一家工作做Java开发

2003 年这年最惨,公司拖欠工资16000元,打过两次官司2005才付清.

2004 年开始加入分布式计算团队,目前成绩，工作仍然是Java开发并且开始使用PostgreSQL数据库。

2004-10月开始玩户外和摄影

2005-6月成为中国无线电运动协会会员,呼号BG7NYT,进了一部Yaesu FT-60R手台。公司的需要转回PHP与MySQL，相隔几年发现PHP进步很大。在前台展现方面无人能敌，于是便前台使用PHP，后台采用Java开发。

2006 年单身生活了这么多年,终于找到归宿. 工作更多是研究PHP各种框架原理

2007 物价上涨,金融危机，休息了4个月（其实是找不到工作），关外很难上439.460中继，搞了一台Yaesu FT-7800.

2008 终于找到英文学习方法，《Netkiller Developer 手札》，《Netkiller Document 手札》

2008-8-8 08:08:08 结婚,后全家迁居湖南省常德市

2009《Netkiller Database 手札》,2009-6-13学车，年底拿到C1驾照

2010 对电子打击乐产生兴趣，计划学习爵士鼓。由于我对Linux热爱，我轻松的接管了公司的运维部，然后开发运维两把抓。我印象最深刻的是公司一次上架10个机柜，我们用买服务器纸箱的钱改善伙食。我将40多台服务器安装BOINC做压力测试，获得了中国第二的名次。

2011 平凡的一年，户外运动停止，电台很少开，中继很少上，摄影主要是拍女儿与家人，年末买了一辆山地车

2012 对油笔画产生了兴趣，活动基本是骑行银湖山绿道，

2013 开始学习民谣吉他，同时对电吉他也极有兴趣；最终都放弃了。这一年深圳开始推数字中继2013-7-6日入手Motorola

MOTOTRBO XIR P8668，Netkiller 系列手机从Sourceforge向Github迁移；年底对MYSQL UDF，Engine与PHP扩展开发产生很浓的兴趣，拾起遗忘10+年的C，写了几个mysql扩展（图片处理，fifo管道与ZeroMQ），10月份入Toyota Rezi 2.5V并写了一篇《攻城狮的苦逼选车经历》

2014-9-8 在淘宝上买了一架电钢琴 Casio Privia PX-5S pro 开始陪女儿学习钢琴，由于这家钢琴是合成器电钢，里面有打击乐，我有对键盘鼓产生了兴趣。

2014-10-2号罗浮山两日游，对中国道教文化与音乐产生了兴趣，10月5号用了半天时间学会了简谱。10月8号入Canon 5D Mark III + Canon Speedlite 600EX-RT香港过关被查。

2014-12-20号对乐谱制作产生兴趣（https://github.com/SheetMusic/Piano），给女儿做了几首钢琴伴奏曲，MuseScore制谱然后生成MIDI与WAV文件。

2015-09-01 晚饭后拿起爵士鼓基础教程尝试在Casio Privia PX-5S pro演练，经过反复琢磨加上之前学钢琴的乐理知识，终于在02号晚上，打出了简单的基本节奏，迈出了第一步。

2016 对弓箭（复合弓）产生兴趣，无奈天朝法律法规不让玩。每周游泳轻松1500米无压力，年底入 xbox one s 和 Yaesu FT-2DR, 同时开始关注功放音响这块

2017 7月9号入 Yamaha RX-V581 功放一台，连接Xbox打游戏爽翻了，入Kindle电子书，计划学习蝶泳，果断放弃运维和开发知识体系转攻区块链。

2018 从溪山美地搬到半岛城邦，丢弃了多年攒下的家底。11 月开始玩 MMDVM，使用 Yaesu FT-7800 发射，连接MMDVM中继板，树莓派，覆盖深圳湾，散步骑车通联两不误。

2019 卖了常德的房子，住了5次院，哮喘反复发作，决定停止电子书更新，兴趣转到知乎，B站

2020 准备找工作

职业生涯路上继续打怪升级

3. 如何获得文档

下载 Netkiller 手札 (epub,kindle,chm,pdf)

EPUB https://github.com/netkiller/netkiller.github.io/tree/master/download/epub

MOBI https://github.com/netkiller/netkiller.github.io/tree/master/download/mobi

PDF https://github.com/netkiller/netkiller.github.io/tree/master/download/pdf

CHM https://github.com/netkiller/netkiller.github.io/tree/master/download/chm

通过 GIT 镜像整个网站

https://github.com/netkiller/netkiller.github.com.git

$ git clone https://github.com/netkiller/netkiller.github.com.git

镜像下载

整站下载

```
wget -m http://www.netkiller.cn/index.html
```

指定下载

```
wget -m wget -m http://www.netkiller.cn/linux/index.html
```

Yum 下载文档

获得光盘介质，RPM包，DEB包，如有特别需要，请联系我

YUM 在线安装电子书

http://netkiller.sourceforge.net/pub/repo/

```
# cat >> /etc/yum.repos.d/netkiller.repo <<EOF
[netkiller]
```

```
name=Netkiller Free Books
baseurl=http://netkiller.sourceforge.net/pub/repo/
enabled=1
gpgcheck=0
gpgkey=
EOF
```

查找包

```
# yum search netkiller

netkiller-centos.x86_64 : Netkiller centos Cookbook
netkiller-cryptography.x86_64 : Netkiller cryptography Cookbook
netkiller-docbook.x86_64 : Netkiller docbook Cookbook
netkiller-linux.x86_64 : Netkiller linux Cookbook
netkiller-mysql.x86_64 : Netkiller mysql Cookbook
netkiller-php.x86_64 : Netkiller php Cookbook
netkiller-postgresql.x86_64 : Netkiller postgresql Cookbook
netkiller-python.x86_64 : Netkiller python Cookbook
netkiller-version.x86_64 : Netkiller version Cookbook
```

安装包

```
yum install netkiller-docbook
```

4. 打赏（Donations）

If you like this documents, please make a donation to support the authors' efforts. Thank you!

您可以通过微信，支付宝，贝宝给作者打赏。

银行(Bank)

招商银行(China Merchants Bank)

开户名：陈景峰

账号：9555500000007459

微信（Wechat）

支付宝（Alipay）

PayPal Donations

https://www.paypal.me/netkiller

5. 联系方式

主站 http://www.netkiller.cn/

备用 http://netkiller.github.io/

繁体网站 http://netkiller.sourceforge.net/

联系作者

Mobile: +86 13113668890

Email: netkiller@msn.com

QQ群: 128659835 请注明“读者”

QQ: 13721218

ICQ: 101888222

注：请不要问我安装问题！

博客 Blogger

知乎专栏 https://zhuanlan.zhihu.com/netkiller

LinkedIn: http://cn.linkedin.com/in/netkiller

OSChina: http://my.oschina.net/neochen/

Facebook: https://www.facebook.com/bg7nyt

Flickr: http://www.flickr.com/photos/bg7nyt/

Disqus: http://disqus.com/netkiller/

solidot: http://solidot.org/~netkiller/

SegmentFault: https://segmentfault.com/u/netkiller

Reddit: https://www.reddit.com/user/netkiller/

Digg: http://www.digg.com/netkiller

Twitter: http://twitter.com/bg7nyt

weibo: http://weibo.com/bg7nyt

Xbox club

我的 xbox 上的ID是 netkiller xbox，我创建了一个俱乐部 netkiller 欢迎加入。

Radio

CQ CQ CQ DE BG7NYT:

如果这篇文章对你有所帮助,请寄给我一张QSL卡片, qrz.cn or qrz.com or hamcall.net

Personal Amateur Radiostations of P.R.China

ZONE CQ24 ITU44 ShenZhen, China

Best Regards, VY 73! OP. BG7NYT

守听频率 DMR 438.460 -8 Color 12 Slot 2 Group 46001

守听频率 C4FM 439.360 -5 DN/VW

MMDVM Hotspot:

Callsign: BG7NYT QTH: Shenzhen, China

YSF: YSF80337 - CN China 1 - W24166/TG46001

DMR: BM_China_46001 - DMR Radio ID: 4600441

第 1 章 Developer & Programming language

1. System programming language

A System programming language is usually used to mean "a language for system programming": that is, a language designed for writing system software as distinct from application software.

http://en.wikipedia.org/wiki/System_programming_language

2. 编程语言热度排行榜

2.1. TIOBE Programming 编程语言排行榜

TIOBE Programming Community Index - 编程语言排行榜 http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html

2.2. Programming Languages Reference Sheets - Hyperpolyglot

http://hyperpolyglot.org/

2.3. DB-Engines - DB-Engines Ranking

http://db-engines.com/en/ranking

2.4. Web Framework Benchmarks

http://www.techempower.com/benchmarks/

2.5. The 10 hottest JavaScript framework projects

The 10 hottest JavaScript framework projects

2.6. 各种语言性能测试

http://benchmarksgame.alioth.debian.org/

2.7. IDE 横向比较

http://en.wikipedia.org/wiki/Comparison_of_integrated_development_environments

2.8. Top 10 Programming Languages

http://spectrum.ieee.org/computing/software/top-10-programming-languages

IEEE Spectrum's 2014 Ranking

2.9. PYPL PopularitY of Programming Language

http://pypl.github.io/PYPL.html

2.10. RedMonk 编程语言排行榜

http://redmonk.com/sogrady/

2.11. Developer Survey Results 2017

http://stackoverflow.com/research/developer-survey-2016

https://stackoverflow.com/insights/survey/2017#technology-most-loved-dreaded-and-wanted-languages

3. 软件工程 (Software Engineering)

4. 项目运作

5. 收集需求

6. 标准与规范

6.1. Project directory

一种很蠢目录规划:

```
project
project/library
project/log
project/tmp
project/...
```

目录规划原则，临时文件分离，日志分离，配置文件分离;这样有利于在负载均衡环境中克隆节点。

- 项目目录/workspace/project
- 临时目录 /workspace/tmp/
- 日志 目录/workspace/log/
- 配置文件/workspace/conf

6.2. 版本控制及如何运作

版本库布局

版本库布局

1. trunk
2. branches
3. tags
4. releases

多项目版本库布局

```
                                           /-> branches
                       .---> project 2 ---> trunk
                      /                    \-> tags
                     /
                    /                                   /->
branches
repositories -------> project 1 ---> trunk
                        \
\-> tags
                         \
                          \                        /-> branches
                           `---> project 3 ---> trunk
                                               \-> tags
```

策略

trunk 主干，作为主干你要保证他的代码是可运行的。

branches 分支，代码来自主干，可以有很多分支，分支用于多个团队同步开发，最终要将代码合并到主干。例如：branches/member, branches/blog, branches/shop 分别来自不同的团队

tags 一般用于做快照，一旦建立永不更改

releases 发行本版，代码来自主干， 与tags功能一样，命名更直观。例如每个月为一个周期，发布一次代码 releases/v1.0, releases/1.5

```
                          .-------------------------------------
----------------------------------> cart r100 ---> ...
                         /
^
                        .----------------------> blog r3 ---> r4
... ... r(10) ---> r(n) ---> r(...) --->/---> ...
                       /                                   ^
\                           /
               .-> branches r1  ---> member r2 ---/---------> r4
... r(10) --\----> r(14) ---> r(...)--/---> ...
```

```
              /                       ^          /
\           \         \             /
repositories ----> trunk    r1  -----/----------/-------------->
r(11) ---> r(12) ---> r(15) ---> r(...) ---> ...
              \
\         \          \
               `-> tags     r1  --------------------------------
----------> v1.0.0 r(13)   \          \
                       \
\
                        `---------------------------------------
-----------------------> v1.0.1 (r16)   \
                         `--------------------------------------
------------------------------------> v1.0.2 r(n)
```

```
.> unstable 2.1.1

/ \
                                              unstable 1.1
unstable 2.1     unstable 3.1
                                                          ^ \
^ \         / \
               .-> branches ---> unstable 1.0 ---/   \
unstable 2.0 ---/   \   unstable 3.0    unstable---> ...
              /                      ^                \
^          \      ^    \          ^
repositories ----> trunk    --------/-----------------> stable -
--/-----> stable ---/---> stable ---/-> stable ---> ...
              \                                         \
\              \
               `-> tags     ----------------------------> stable
1.0          \              \
                       \
\              \
                        `---------------------------------------
----------------> stable 2.0   \
                         `--------------------------------------
-------------------------------> stable 3.0
```

怎样访问版本库

美工、页面人员采用WebDav访问Subversion。Photoshop,Dreamwaver 软件对WebDav有很好地支持，他们不需要学习如何使用Subversion。

开发者通过Svn客户端访问代码库，既可以使用开发IDE集成工具也可以使用单独工具。像TortoiseSVN等等。

nightly version

每天晚上做一个快照

rc1,rc2,rc4

6.3. 代码审查

Code Review

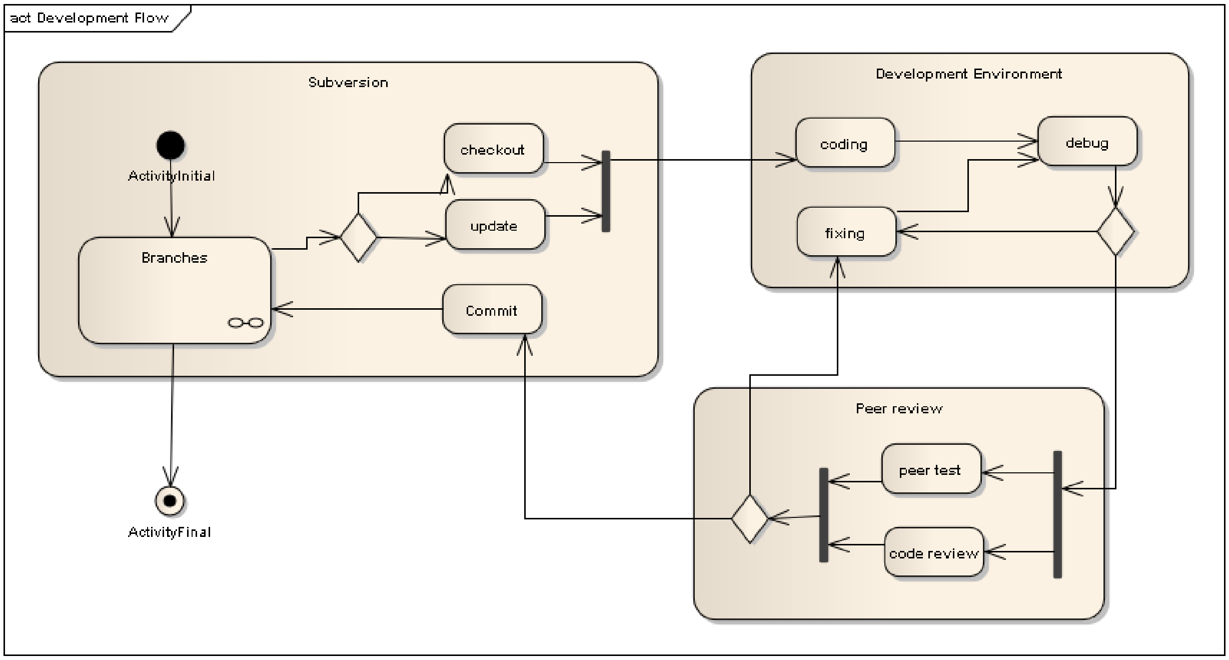
一对一代码审查可以提高程序质量.

```


developer neo ---> coding ---> commit --->.
                           /               \
developer zen ---> review '                 \
                                           --->  svn repos
developer neo ---> review .                 /
                           \               /
developer zen ---> coding ---> commit --->`


```

Coding workflow



6.4. Redis Key

Redis Key 使用“：”分割例如

```
set SMS:CAPTCHA 1234
```

6.5. 错误编码

格式 ABCD

A: 编码

1 用户错误

2 网络错误

3 系统错误

4 应用服务器错误

5 应用程序错误

6 缓存错误

7 数据库错误

8 搜索引擎

B 编码

0 成功

1 失败

剩余 C D 用户自行编码

6.6. HTML 标准

校验

```
https://html5.validator.nu/?
doc=https%3A%2F%2Fwww.netkiller.cn%2Findex.html
```

XHTML/HTML

CSS

Script

6.7. 编码风格

```
https://code.google.com/p/google-styleguide/
http://lxr.linux.no/linux/Documentation/CodingStyle
http://perldoc.perl.org/perlstyle.html
http://www.gnu.org/prep/standards/
```

java 编程规范

一个输入多个出口

```
                        Object row = input.readLine();
                        if (row != null) {
                                if (this.process != null) {
                                        row = 
this.process.run(row);
                                }
                                if (row != null) {
                                        boolean outputStatus = 
this.output.write(row);
                                        if (this.position != 
null && outputStatus) {


this.position.set(row);

                                        }
                                }
                        }
```

Spring Data JPA

```
        @Autowired
        private TableRepostitory tableRepostitory;

        @Autowired
        private JdbcTemplate jdbcTemplate;

        @PersistenceContext
        private EntityManager entityManager;
```

php 文件

http://www.php-fig.org

格式与 编码

使用 UNIX 风格换行, 请在你的编辑器内调整

```
UNIX (LF或"\n")
MAC OS (CR 或"\r")
Windows CRLF \r\n
```

源码文件使用 UTF-8

有些IDE环境 UTF-8 BOM

循环嵌套

if, while, for, foreach, do ... loop, switch... 等的嵌套必须小于等于3层

如下面的例子，可读性极差。

```
if (xxx){
        if (xxx){
                if(xxx){
                        if(xxx){
                                if(xxx){

                                }
                        }
                }
                if(xxx){
                        if(xxx){
                        }
                }
        }
        if (xxx){
                if(xxx){
                        if(xxx){
                        }
                }
                if(xxx){
                        if(xxx){
                        }
                }
        }
}
```

加以改造

```
func aaa(p){
        if(p){
                if(xxx){
                }
        }
}
func bbb(b){
        if(b){
                if(xxx){
                        if(xxx){

                        }
                }
        }
}

if(xxx){
        aaa(xxx)
}
if(b){
        bbb(b)
}
```

取出行尾的空格以及多余的换行符

一个空格占用一个字节,换行符Window是两个字节\r\n, Unix与Mac占用一个字节

php 标签

禁止这样使用

```
<?
...
?>
```

正确的使用方法

```
<?php
...

or

<?php
...
?>
```

头部注释

```
<?php
/**
 * Project Name
 *
 * @author     $Author: netkiller $
 * @copyright  Copyright (c) 2012 Company
 * @version    $Id: chapter.coding.xml 584 2013-05-15 05:13:17Z netkiller $
 */

<?php
/**
 * Project Name
 *
 * @author     $Author: netkiller $
 * @license    GNU General Public License 2.0
 * @version    $Id: chapter.coding.xml 584 2013-05-15 05:13:17Z
netkiller $
 */
```

String

双引号要处理字符串转义，性能上不如单引号，如果你不需要转义字符串，或者字符串中不含原转译字符，建议你使用单引号

```
print("string")
```

每次输出会检索特殊字符串如： \r, \n, \t, \0xFF 等等

```
print('string')
```

Database

使用pdo_mysql替代mysql

错误的写法，通过字符串链接拼接sql语句极容易出现注入漏洞

```
$sql = "select * from table where id=".$id;
$sql = "select * from table where id='".$id."'";
$sql = "INSERT INTO fruit(name, colour) VALUES ('".$name."',
'".$colour."')";
```

正确的写法

```
$sql = "select * from table where id=?";
$sql = "INSERT INTO fruit(name, colour) VALUES (?, ?)";
```

```
$sql = <<<____SQL
     CREATE TABLE IF NOT EXISTS `ticket_hist` (
       `tid` int(11) NOT NULL,
       `trqform` varchar(40) NOT NULL,
       `trsform` varchar(40) NOT NULL,
       `tgen` datetime NOT NULL,
       `tterm` datetime,
       `tstatus` tinyint(1) NOT NULL
     ) ENGINE=ARCHIVE COMMENT='ticket archive';
____SQL;
```

结果集使用注意事项

返回数据库查询结果有几种形式

数组形式

```
Array
(
    [0] => banana
    [1] => yellow
)

Array
(
    [NAME] => banana
    [COLOUR] => yellow
)
```

对象形式

```
Object
(
        Obj->NAME
        Obj->COLOUR
)
```

正确的使用方式

```
print($row[name])
print($row->name)
```

错误的使用使方式

```
print($row[0])
```

避免使用 "*"查询，一会影响性能，二增加带宽开销

```
$sql = "select * from tab where status=0 limit 1";
```

如果程序使用$row[1]读取结果，有可能当数据库结构改变，增加字段，字段顺序发生变化，输出数据都会出错

索引

下面的例子，不会使用索引

```
$sql = "select id, name, created from tab where id != 100";
```

```
EXPLAIN select * from members where id != '1010';
索引失效
EXPLAIN select count(*) from members where id != '1010';
索引有效
```

缓存

下面的例子，数据不会缓存查询结果

```
$sql = "select id, name, created from tab where created=now()";
```

6.8. 安全

Interface

SQL注入

第 2 章 Editor

- vi

  vi file

- emacs

  emacs

- nano/pico

  nano file / pico file

- joe

  joe file

  ```
  sudo apt-get install joe
  ```

1. VIM

Vi IMproved - enhanced vi editor (transitional package)

```
sudo apt-get install vim-perl vim-python vim-latexsuite
```

New text file line delimiter

```
:set ff
```

```
:set ff=dos
:set ff=unix
:set ff=mac
```

vim中的出现^M

```
sudo apt-get install tofrodos
dos2unix
```

输入

```
:set notextmode
```

tab 符

```
set expandtab tabstop=4 shiftwidth=4 softtabstop=4
foldmethod=marker
```

粘贴是禁用格式化功能

:set paste

Regular Expressions

search & replace

```
%s/neo/leo/g
%s#/logs#/www#g
```

会去掉**vim**文件中的空白行

```
:%s/^\s*\n/
```

大小写转换

将光标移动到想要大小写转换的地方然后键入

guw (大写转换成小写)

gUw (小写转换成大写)

g~w (是改变光标到词尾)

guu gUU g~~ (是改变一行)

split

```
:split <filename>        按拆分模式打开文件，默认为上下拆分
:vsplit <filename>       左右拆分
```

Ctrl-w w

切换活动窗口

Ctrl-w Ctrl-w

同上

Ctrl-w h/j/k/l

切换活动窗口为 左/上/下/右 边的窗口

:q

关闭光标所在的分屏

vimrc

```
neo@netkiller:~$ cat /home/neo/.vimrc
set expandtab tabstop=4 shiftwidth=4 softtabstop=4
foldmethod=marker
set paste
set ruler
set number
set nocp " don't open Vim in Vi-compatible mode
set bs=2 " setup backspace to delete previous char in insert
mode
set encoding=utf-8 fileencodings= " set UTF-8 for all files
set autoindent
set smartindent
set t_Co=256
syntax on
set syntax=tt2html
set hlsearch
colorscheme elflord
```

vimdiff - edit two or three versions of a file with Vim and show differences

```
# vimdiff  FILE_LEFT  FILE_RIGHT
# vim -d  FILE_LEFT  FILE_RIGHT
# vim FILE_LEFT
:vertical diffsplit FILE_RIGHT
```

光标移动

左右窗口联动控制

```
:set scrollbind
:set noscrollbind
```

在各个差异点之间快速移动。

```
]c        跳转到下一个差异点
[c        跳转到上一个差异点
```

如果在命令前加上数字的话，可以跳过一个或数个差异点，从而实现跳的更远。比如如果在位于第一个差异点的行输入"2]c"，将越过下一个差异点，跳转到第三个差异点。

文件合并

```


文件比较的最终目的之一就是合并，以消除差异。如果希望把一个差异点中当前文件的
内容复制到另一个文件里，可以使用命令
dp  (diff "put")

如果希望把另一个文件的内容复制到当前行中，可以使用命令
do (diff "get"，之所以不用dg，是因为dg已经被另一个命令占用了)

如果希望手工修改某一行，可以使用通常的vim操作。如果希望在两个文件之间来回跳
转，可以用下列命令序列：
Ctrl-w, w

在修改一个或两个文件之后，vimdiff会试图自动来重新比较文件，来实时反映比较
结果。但是也会有处理失败的情况，这个时候需要手工来刷新比较结果：
```

```
:diffupdate
```

如果希望撤销修改，可以和平常用vim编辑一样，直接

```
<ESC>, u
```

但是要注意一定要将光标移动到需要撤销修改的文件窗口中。

上下文

上下文的展开和查看
比较和合并文件的时候经常需要结合上下文来确定最终要采取的操作。Vimdiff 缺省是会把不同之处上下各 6 行的文本都显示出来以供参考。其他的相同的文本行被自动折叠。如果希望修改缺省的上下文行数，可以这样设置：

```
:set diffopt=context:3
```

可以用简单的折叠命令来临时展开被折叠的相同的文本行：
zo （folding open，之所以用z这个字母，是因为它看上去比较像折叠着的纸）

然后可以用下列命令来重新折叠：
zc （folding close）

保存于退出

同时操作两个文件
在比较和合并告一段落之后，可以用下列命令对两个文件同时进行操作。比如同时退出：
:qa （quit all）

如果希望保存全部文件：
:wa （write all）

或者是两者的合并命令，保存全部文件，然后退出：
:wqa （write, then quit all）

如果在退出的时候不希望保存任何操作的结果：
:qa! （force to quit all）

vim backup script

```
#!/bin/bash
########################################
# vim script for automatic backup by neo
########################################

datetime=`date +"%Y-%m-%d.%H:%M:%S"`
current_date=`date +"%Y-%m-%d"`
current_time=`date +"%H:%M:%S"`
filename=`basename $1`
original=$1

if [ -f ${original}.original ]
then
    echo "[${datetime}] [B] ${original} to
${original}.original" >> ~/backup/history.log
else
    cp ${original} ${original}.original
fi

if [ -d ~/backup/${current_date} ]
then
    echo "[${datetime}] [O] ${original}" >>
~/backup/history.log
else
    mkdir -p ~/backup/${current_date}
    echo [${datetime}] [C] mkdir ~/backup/${current_date} >>
~/backup/history.log
fi

backup=~/backup/${current_date}/${filename}.${current_time}
if [ -f ${original} ];
then
    cp ${original} ${backup}
    echo "[${datetime}] [B] ${original} to ${backup}" >>
~/backup/history.log
fi

vim ${original}
```

```
datetime=`date +"%Y-%m-%d.%H:%M:%S"`
current_date=`date +"%Y-%m-%d"`
current_time=`date +"%H:%M:%S"`
newfile=~/backup/${current_date}/${filename}.${current_time}

if [ -f ${original} ];
then
    if [ -f ${backup} ];
    then
        original_sha=`sha1sum ${backup} |awk -F ' ' '{print
$1}'`
        newfile_sha=`sha1sum ${original} |awk -F ' ' '{print
$1}'`

        if [ $original_sha = $newfile_sha ];
        then
            echo "[${datetime}] --- " >> ~/backup/history.log
            exit
        fi

    fi
    cp ${original} ${newfile}
    echo "[${datetime}] [M] ${original}" >>
~/backup/history.log
    echo "[${datetime}] [B] ${original} to ${newfile}" >>
~/backup/history.log

fi
echo "[${datetime}] --- " >> ~/backup/history.log
exit
```

2. Eclipse

```
Window-Perferenes-General-Keys

Remove Trailing Whitespace
```

AnyEdit

http://andrei.gmxhome.de/anyedit/

Remove trailing whitespace:

Window -> Preferences -> PHP -> Editor -> Save Actions

[v] Remove trailing whitespace

() All lines (o) Ignore empty lines

AnyEdit tools plugin for Eclipse

Eclim

http://eclim.org/index.html

安装Eclim

```
java -jar eclim_1.7.0.jar
```

Vrapper

Eclipse 的 VIM 插件

http://vrapper.sourceforge.net/home/

3. Komodo Edit

http://www.activestate.com/komodo_edit/

4. Syntext Serna Free

http://www.syntext.com/downloads/serna-free/

5. SciTE

http://www.scintilla.org/

```
$ apt-cache search SciTE
scite - Lightweight GTK-based Programming Editor
sciteproj - project manager for the SciTE editor
```

第 3 章 Comparison, merge and patch utilities

1. comparison, merge

sdiff

```
neo@netkiller:/tmp$ sdiff file1 file2
#!/bin/bash
#!/bin/bash
echo "Helloworld"                                             |
echo "Helloword"
echo "Line 1"
echo "Line 1"
```

vimdiff

```
$ vimdiff file1 file2
```

2. patch - apply a diff file to an original

diff

-r 是一个递归选项，设置了这个选项，diff会将两个不同版本源代码目录中的所有对应文件全部都进行一次比较，包括子目录文件。

-N 选项确保补丁文件将正确地处理已经创建或删除文件的情况。

-u 选项以统一格式创建补丁文件，这种格式比缺省格式更紧凑些。

patch

-p0 选项要从当前目录查找目的文件（夹）

-p1 选项要忽略掉第一层目录，从当前目录开始查找。

-E 选项说明如果发现了空文件，那么就删除它

-R 选项说明在补丁文件中的“新”文件和“旧”文件现在要调换过来了（实际上就是给新版本打补丁，让它变成老版本）

例 3.1. single file

make a patch file

```
$ diff -uN file.old file.new > file.patch
```

take a patch file

```
$ patch -p0 < file.patch
```

例 3.2. a lot of file or directory

```
$ diff —uNr dir.old dir.new >my.patch
```

```
$ patch -p1 < file.patch
```

3. WinMerge

http://winmerge.org/

开源，一般化

4. Beyond Compare Pro

最好用的文件比较合并工具

部分 I. Web page

第 4 章 gulpjs

安装

```
npm install gulp-cli
npm install gulp -D
```

创建 gulpfile.js 文件

```
var gulp = require('gulp');
var pug = require('gulp-pug');
var less = require('gulp-less');
var minifyCSS = require('gulp-csso');

gulp.task('html', function(){
  return gulp.src('client/templates/*.pug')
    .pipe(pug())
    .pipe(gulp.dest('build/html'))
});

gulp.task('css', function(){
  return gulp.src('client/templates/*.less')
    .pipe(less())
    .pipe(minifyCSS())
    .pipe(gulp.dest('build/css'))
});

gulp.task('default', [ 'html', 'css' ]);
```

排除目录

```
gulp.src(['js/**/*.js', '!js/**/*.min.js'])
```

1. Tasks automation

1.1. gulp-changed

```
// npm install --save-dev gulp gulp-changed gulp-jscs gulp-uglify

var gulp = require('gulp');
var changed = require('gulp-changed');
var jscs = require('gulp-jscs');
var uglify = require('gulp-uglify');

var SRC = 'src/**/*.js';
var DEST = 'dist';

gulp.task('default', function() {
    return gulp.src(SRC)
        .pipe(changed(DEST)) // changed 任务需要提前知道目标目录位置才能找出哪些文件是被修改过的,只有被更改过的文件才会通过这里
        .pipe(jscs())
        .pipe(uglify())
        .pipe(gulp.dest(DEST));
});
```

1.2. 显示处理进度

显示处理中的文件

```
gulp.task('minify-css', function () {
        gulp.src([src + '/**/css/**/*.css', "!"+src +
'/**/css/**/*.min.css'])
        .on('data', function(file) {
                console.log('%s Read %d bytes of data',
file.path, file.contents.length);
        })
        .pipe(concat("finally.css"))
        .pipe(rename({ suffix: '.min' }))
        .pipe(minifycss())
        .pipe(gulp.dest( dist ));

});
```

1.3. notify

```
npm install --save-dev gulp-notify
```

```
var notify = require('gulp-notify');
```

```
.pipe(notify({ message: 'Styles task complete' }));
```

1.4. del

```
var gulp = require('gulp');
var del = require('del');
```

```
gulp.task('clean:mobile', function (cb) {
  del([
    'dist/**/css/*.min.css',
    'dist/mobile/**/*',
    '!dist/mobile/deploy.json'
  ], cb);
});

gulp.task('default', ['clean:mobile']);

// Clean
gulp.task('clean', function() {
        return del(['dist/styles', 'dist/scripts',
'dist/images']);
});
```

1.5. start

```


// Default task
gulp.task('default', ['clean'], function() {
        gulp.start('styles', 'scripts', 'images');
});
```

2. watch

```
// including plugins
var gulp = require('gulp');
var watch = require('gulp-watch');

gulp.task('watch', function() {
    watch(__dirname + "/css/**/*.css", function() {
        gulp.run('minify-css');
    });
    watch(__dirname + "/js/**/*.js", function() {
        gulp.run('minify-js');
    });
});
```

```
gulp watch
```

3. HTML Minification

```
npm install --save-dev gulp-minify-html
```

gulpfile.js:

```
// including plugins
var gulp = require('gulp');
var minifyHtml = require("gulp-minify-html");

// task
gulp.task('minify-html', function () {
    gulp.src('./html/*.html') // path to your files
    .pipe(minifyHtml())
    .pipe(gulp.dest('path/to/destination'));
});
```

```
gulp minify-html
```

禁止处理 SSI 服务器端包含

```
gulp.task('minify-html', function () {
    gulp.src( src +'/cfd/*.html')
    .pipe(minifyHtml({ssi:true, quotes:false}))
    .pipe(gulp.dest( dist + '/cfd/'));
});
```

4. CSS Minification

4.1. gulp-minify-css

```
npm install --save-dev gulp-minify-css
```

```
// including plugins
var gulp = require('gulp');
var minifyCss = require("gulp-minify-css");

// task
gulp.task('minify-css', function () {
    gulp.src('./css/one.css') // path to your file
    .pipe(minifyCss())
    .pipe(gulp.dest('path/to/destination'));
});

// task
gulp.task('minify-multi-css', function () {
    gulp.src(__dirname+'/css/*.css') // path to your file
    .pipe(minifyCss())
    .pipe(gulp.dest('path/to/destination'));
});
```

```
gulp minify-css
gulp minify-multi-css
```

4.2. gulp-clean-css

```
var gulp = require('gulp');
var minifycss = require('gulp-clean-css');
var concat = require('gulp-concat');
var uglify = require('gulp-uglify');
var rename = require('gulp-rename');

gulp.task('minify-css', function () {

        gulp.src('./css/*.css')
        .pipe(concat("finally.css"))
        .pipe(rename({ suffix: '.min' }))
        .pipe(minifycss())
        .pipe(gulp.dest('build/css'));

});

gulp.task('minify-js', function(){

        gulp.src(__dirname + "/js/*.js")
        .pipe(concat("finally.js"))
        .pipe(rename({ suffix: '.min' }))
        .pipe(uglify())
        .pipe(gulp.dest('build/js'))

});
```

compatibility

```
var gulp = require('gulp');
var cleanCSS = require('gulp-clean-css');

gulp.task('minify-css', function() {
  return gulp.src('styles/*.css')
    .pipe(cleanCSS({compatibility: 'ie8'}))
    .pipe(gulp.dest('dist'));
});
```

callback

```
var gulp = require('gulp');
var cleanCSS = require('gulp-clean-css');

gulp.task('minify-css', function() {
    return gulp.src('styles/*.css')
        .pipe(cleanCSS({debug: true}, function(details) {
            console.log(details.name + ': ' +
details.stats.originalSize);
            console.log(details.name + ': ' +
details.stats.minifiedSize);
        }))
        .pipe(gulp.dest('dist'));
});
```

4.3. gulp-make-css-url-version

给css文件里引用url加版本号（md5sum），像这样:

```
background: url(../images/pc-banner-bg.jpg?
v=4facbd0914639f296faec4dba4d358f0) no-repeat;}
```

```
npm install --save-dev gulp-make-css-url-version
```

```
var gulp = require('gulp'),
minifier = require('gulp-minify-css');
cssver = require('gulp-make-css-url-version');

gulp.task('testCssmin', function () {
    gulp.src('src/css/*.css')
        .pipe(cssver()) //给css文件里引用文件加版本号（文件MD5）
        .pipe(minifier())
        .pipe(gulp.dest('dist/css'));
});
```

4.4. CSS 冗余分析

检查出重复定义的CSS

Installation

```
npm install gulp-csscss --save-dev
```

Example

```
var gulp = require('gulp');
var csscss = require('gulp-csscss');

gulp.task('default', function() {
  gulp.src('src/style.css')
    .pipe(csscss())
});
```

5. JS Minification

```
npm install --save-dev gulp-uglify
```

gulpfile.js:

```
// including plugins
var gulp = require('gulp');
var uglify = require("gulp-uglify");

// task
gulp.task('minify-js', function () {
    gulp.src('./JavaScript/*.js') // path to your files
    .pipe(uglify())
    .pipe(gulp.dest('path/to/destination'));
});
```

Run:

```
gulp minify-js
```

5.1. JS 校验

```
var gulp = require('gulp');
var header = require('gulp-header');  //给文本文件头部追加内容
var footer = require('gulp-footer');
var concat = require('gulp-concat');
var jshint = require('gulp-jshint'); //js代码校验
var cached = require('gulp-cached');
var remember = require('gulp-remember'); //gulp-remember is
a gulp plugin that remembers files that have passed through
it. gulp-remember adds all the files it has ever seen back into
the stream.

gulp.task('scripts', function() {
  return gulp.src('src/**/*.js')
      .pipe(cached('scripts'))        // 只传递更改过的文件
      .pipe(jshint())                 // 对这些更改过的文件做一些特
殊的处理...
      .pipe(header('(function () {')) // 比如 jshinting ^^^
      .pipe(footer('})();'))          // 增加一些类似模块封装的东西
      .pipe(remember('scripts'))      // 把所有的文件放回 stream
      .pipe(concat('main.js'))         // 合并文件的操作
      .pipe(gulp.dest('public/'));
});
```

6. CSS Sprite

简介

gulp-spriter：帮助前端工程师将css代码中的切片图片合并成雪碧图，支持retina图片。

功能

使用二叉树排列算法，对图片排序优化
自动收集css中带切片的图片（仅对background-image:url("slice/xx.png")有效）
自动在原来的css中添加background-position属性
支持生成适用于高清设备的雪碧图，并在css文件追加媒体查询css代码

依赖

gulp-spriter使用spritesmith作为图片生成的基础算法

安装

```
npm install gulp-spriter
```

配置

导入gulp-spriter依赖：

```
var spriter = require("gulp-spriter");
```

gulpfile配置文件中增加task，如下：

```
gulp.task("css",["clean"],function(){
  return gulp.src("./src/css/xxx.css")
         .pipe(spriter({
            sprite:"test.png",
            slice:"./src/slice",
            outpath:"./build/tests"
          }))
         .pipe(gulp.dest('./build/css'))
})
```

参数

sprite:[string] 必须，设置输出的雪碧图名称
slice：[string] 必须，切片文件存放位置，基于根目录
outpath：[string] 必须，输出的雪碧图位置

7. Compress Images

```
optimizationLevel: 5, //类型：Number  默认：3  取值范围：0-7（优化等级）
progressive: true, //类型：Boolean 默认：false 无损压缩jpg图片
interlaced: true, //类型：Boolean 默认：false 隔行扫描gif进行渲染
multipass: true //类型：Boolean 默认：false 多次优化svg直到完全优化
```

```
var imagemin = require('gulp-imagemin');

gulp.task('images', function() {
  return gulp.src('src/images/**/*')
    .pipe(imagemin({ optimizationLevel: 3, progressive: true, interlaced: true }))
    .pipe(gulp.dest('dist/assets/img'))
    .pipe(notify({ message: 'Images task complete' }));
});
```

8. WEBP格式图片

```
npm install --global gulp-imageisux
npm install --save-dev gulp-imageisux

var imageisux = require('gulp-imageisux');

gulp.task('imageisux', function() {
        return gulp.src(['img/*'])
                           .pipe(imageisux('/dirpath/',true));
});
```

9. Sass Compilation

Using gulp-sass

```
npm install --save-dev gulp-sass
```

gulpfile.js:

```
// including plugins
var gulp = require('gulp');
var sass = require("gulp-sass");

// task
gulp.task('compile-sass', function () {
    gulp.src('./Sass/one.sass') // path to your file
    .pipe(sass())
    .pipe(gulp.dest('path/to/destination'));
});
```

Run:

```
gulp compile-sass
```

10. Less Compilation

Using gulp-less

```
npm install --save-dev gulp-less
```

gulpfile.js:

```
// including plugins
var gulp = require('gulp');
var less = require("gulp-less");

// task
gulp.task('compile-less', function () {
    gulp.src('./Less/one.less') // path to your file
    .pipe(less())
    .pipe(gulp.dest('path/to/destination'));
});
```

Run:

```
gulp compile-less
```

11. 重命名文件名

```
Using gulp-rename

npm install --save-dev gulp-rename
```

gulpfile.js:

```
// including plugins
var gulp = require('gulp');
var rename = require('gulp-rename');
var coffee = require("gulp-coffee");

// task
gulp.task('rename', function () {
    gulp.src('./CoffeeScript/one.coffee') // path to your file
    .pipe(coffee())  // compile coffeeScript
    .pipe(rename('renamed.js')) // rename into "renamed.js"
(original name "one.js")
    .pipe(gulp.dest('path/to/destination'));
});
```

Run:

```
gulp rename
```

12. 合并文件

Concatenate files using gulp-concat

```
npm install --save-dev gulp-concat
```

gulpfile.js:

```
// including plugins
var gulp = require('gulp');
var concat = require("gulp-concat");

// task
gulp.task('concat', function () {
    gulp.src('./javascript/*.js') // path to your files
    .pipe(concat('concat.js'))  // concat and name it
"concat.js"
    .pipe(gulp.dest('path/to/destination'));
});
```

Run:

```
gulp concat
```

13. 文件头

```
Using gulp-header and Node’s file system

npm install --save-dev gulp-header
```

Copyright 头文件

```
# vim Copyright

/*
Author: netkiller <netkiller@msn.com>
Website: https://www.netkiller.cn
Version: <%= version %>
*/
```

Version 文件

```
# vim Version
1.0.0
```

gulpfile.js:

```
// including plugins
var gulp = require('gulp')
, fs = require('fs')
, concat = require("gulp-concat")
, header = require("gulp-header");

// functions

// Get version using NodeJs file system
var getVersion = function () {
    return fs.readFileSync('Version');
};

// Get copyright using NodeJs file system
var getCopyright = function () {
    return fs.readFileSync('Copyright');
};

// task
gulp.task('concat-copyright-version', function () {
    gulp.src('./javascript/*.js')
    .pipe(concat('finaly.js')) // concat and name it "concat-
copyright-version.js"
    .pipe(header(getCopyrightVersion(), {version:
getVersion()}))
    .pipe(gulp.dest('path/to/destination'));
});
```

Run:

```
gulp concat-copyright-version
```

14. yargs 命令行参数传递

```
npm install --save-dev yargs
```

```
var argv = require('yargs').argv;

gulp.task('my-task', function() {
    return gulp.src(argv.a == 1 ? options.SCSS_SOURCE :
options.OTHER_SOURCE)
        .pipe(sass({style:'nested'}))
        .pipe(autoprefixer('last 10 version'))
        .pipe(concat('style.css'))
        .pipe(gulp.dest(options.SCSS_DEST));
});
```

```
var argv = require('yargs').argv,
    gulpif = require('gulp-if'),
    rename = require('gulp-rename'),
    uglify = require('gulp-uglify');

gulp.task('my-task-stage', function() {
  gulp.src('src/**/*.js')
    .pipe(concat('out.js'))
    .pipe(gulpif(argv.production, uglify()))
    .pipe(gulpif(argv.production, rename({suffix: '.min'})))
    .pipe(gulp.dest('dist/'));
});
```

```
gulp my-task -a 1
gulp my-task-stage --production
```

14.1. gulp-util

```
var util = require('gulp-util');

gulp.task('styles', function() {
  return gulp.src(['src/styles/' + (util.env.theme ?
util.env.theme : 'main') + '.scss'])
    .pipe(compass({
        config_file: './config.rb',
        sass   : 'src/styles',
        css    : 'dist/styles',
        style  : 'expanded'

    }))
    .pipe(autoprefixer('last 2 version', 'safari 5', 'ie 8',
'ie 9', 'ff 17', 'opera 12.1', 'ios 6', 'android 4'))
    .pipe(livereload(server))
    .pipe(gulp.dest('dist/styles'))
    .pipe(notify({ message: 'Styles task complete' }));
});
```

```
gulp watch --theme literature
```

14.2. minimist

```
var gulp   = require('gulp');

// npm install gulp yargs gulp-if gulp-uglify
var args   = require('yargs').argv;
var gulpif = require('gulp-if');
var uglify = require('gulp-uglify');
var isProduction = args.env === 'production';

gulp.task('scripts', function() {
  return gulp.src('**/*.js')
    .pipe(gulpif(isProduction, uglify())) // only minify if
production
    .pipe(gulp.dest('dist'));
});

gulp scripts --env production
```

```
Pass arguments from the command line

// npm install --save-dev gulp gulp-if gulp-uglify minimist

var gulp = require('gulp');
var gulpif = require('gulp-if');
var uglify = require('gulp-uglify');

var minimist = require('minimist');

var knownOptions = {
  string: 'env',
  default: { env: process.env.NODE_ENV || 'production' }
};

var options = minimist(process.argv.slice(2), knownOptions);

gulp.task('scripts', function() {
  return gulp.src('**/*.js')
    .pipe(gulpif(options.env === 'production', uglify())) //
only minify in production
    .pipe(gulp.dest('dist'));
});
Then run gulp with:

$ gulp scripts --env development
```

15. gulp-sourcemaps

```
var gulp = require('gulp');
var plugin1 = require('gulp-plugin1');
var plugin2 = require('gulp-plugin2');
var sourcemaps = require('gulp-sourcemaps');

gulp.task('javascript', function() {
  gulp.src('src/**/*.js')
    .pipe(sourcemaps.init())
      .pipe(plugin1())
      .pipe(plugin2())
    .pipe(sourcemaps.write())
    .pipe(gulp.dest('dist'));
});
```

16. gulp-zip

```
var gulp             = require( 'gulp' ),
    zip              = require( 'gulp-zip' );

gulp.task( 'zip', function() {

  return gulp.src( [
    '!{.gitignore,README.md}',
    '!node_modules', '!node_modules/**',
    '!dist', '!dist/**',
    './**',
  ] )
    .pipe( zip('archive.zip') )
    .pipe( gulp.dest( 'dist' ) );

});
```

17. 清理JS中的console.log()调试语句

```
var gulp = require('gulp');
var stripDebug = require('gulp-strip-debug');

gulp.task('default', function () {
    return gulp.src('src/app.js')
        .pipe(stripDebug())
        .pipe(gulp.dest('dist'));
});
```

18. copy-dir

install

```
npm install copy-dir
```

usage

```
Sync:

var copydir = require('copy-dir');

copydir.sync('/my/from/path', '/my/target/path');
Async:

var copydir = require('copy-dir');

copydir('/my/from/path', '/my/target/path', function(err){
  if(err){
    console.log(err);
  } else {
    console.log('ok');
  }
});
add a filter
When you want to copy a directory, but some file or sub
directory is not you want, you can do like this:

Sync:

var path = require('path');
var copydir = require('copy-dir');

copydir.sync('/my/from/path', '/my/target/path', function(stat,
filepath, filename){
  if(stat === 'file' && path.extname(filepath) === '.html') {
    return false;
  }
  if (stat === 'directory' && filename === '.svn') {
    return false;
  }
  return true;
}, function(err){
  console.log('ok');
});
Async:

var path = require('path');
var copydir = require('copy-dir');

copydir('/a/b/c', '/a/b/e', function(stat, filepath, filename){
  //...
}, function(err) {
  //...
});
```

19. gulp-copy

Usage

```
// gulpfile.js

var gulpCopy = require('gulp-copy');
var sourceFiles = [ 'source1/*', 'source2/*.txt' ];
var destination = 'dest/';

return gulp
    .src(sourceFiles)
    .pipe(gulpCopy(outputPath, options))
    .dest(destination);
```

20.

21. Example

21.1. HTML,JS,CSS

```
var gulp = require('gulp');
var minifyHtml = require("gulp-minify-html");
var minifycss = require('gulp-clean-css');
//var minifycss = require("gulp-minify-css");
var concat = require('gulp-concat');
var uglify = require('gulp-uglify');
var rename = require('gulp-rename');
var path = require("path");
var glob = require("glob");
var watch = require('gulp-watch');

gulp.task('minify-html', function () {
    gulp.src('./*.html')
    .pipe(minifyHtml())
    .pipe(gulp.dest('build/'));
});

gulp.task('minify-css', function () {

        gulp.src('./css/*.css')
        .pipe(concat("finally.css"))
        .pipe(rename({ suffix: '.min' }))
        .pipe(minifycss())
        .pipe(gulp.dest('build/css'));

});

gulp.task('minify-js', function(){

        gulp.src(__dirname + "/js/*.js")
        .pipe(concat("finally.js"))
        .pipe(rename({ suffix: '.min' }))
        .pipe(uglify())
        .pipe(gulp.dest('build/js'))

});
```

```
gulp.task('default',function() {
    gulp.start('minify-css','minify-js');
});

gulp.task('watch', function() {
    watch(__dirname + "/css/**/*.css", function() {
        gulp.run('minify-css');
    });
    watch(__dirname + "/js/**/*.js", function() {
        gulp.run('minify-js');
    });
});
```

21.2. 命令行传递参数

```
var gulp = require('gulp');
var argv = require('yargs').argv;
//var minifyHtml = require("gulp-minify-html");
var minifycss = require('gulp-clean-css');
//var minifycss = require("gulp-minify-css");
var spriter = require("gulp-spriter");
var concat = require('gulp-concat');
var uglify = require('gulp-uglify');
var rename = require('gulp-rename');
var path = require("path");
var glob = require("glob");
var watch = require('gulp-watch');

//var src = __dirname + "/" + argv.stage + "/build/" +
argv.src;
//var dest = __dirname + "/" + argv.stage + "/build/" +
argv.src;
var src = __dirname + "/" + argv.src;
var dest = __dirname + "/dist/" + argv.src;
console.log(src);
```

```
gulp.task('minify-html', function () {
    gulp.src( src +'/*.html') // path to your files
    .pipe(minifyHtml())
    .pipe(gulp.dest( dest + '/'));
});

gulp.task('minify-css', function () {

        gulp.src(src + '/css/**/*.css')
        .pipe(concat("finally.css"))
        .pipe(rename({ suffix: '.min' }))
        .pipe(minifycss())
        .pipe(gulp.dest(dest + '/css'));

});

gulp.task('minify-js', function(){

        gulp.src(src + "/js/**/*.js")
        .pipe(concat("finally.js"))
        .pipe(rename({ suffix: '.min' }))
        .pipe(uglify())
        .pipe(gulp.dest( dest + '/js'))

});

gulp.task("spriter",["clean"],function(){
  return gulp.src( dest + "/css/finally.min.css")
         .pipe(spriter({
            sprite:"finally.png",
            slice: src + "/images",
            outpath: dest + "/images"
          }))
         .pipe(gulp.dest( dest + '/images'))
})


gulp.task('default',function() {
    gulp.start('minify-css','minify-js');
});

gulp.task('watch', function() {
    watch(src + "/css/**/*.css", function() {
        gulp.run('minify-css');
    });
    watch(src + "/js/**/*.js", function() {
        gulp.run('minify-js');
    });
});
```

第 5 章 webpack

第6章 minifier

```
#!/bin/bash

cd /usr/local/src/
wget
https://github.com/yui/yuicompressor/releases/download/v2.4.8/y
uicompressor-2.4.8.jar
mv yuicompressor-2.4.8.jar /usr/local/libexec/

cat >> /usr/local/bin/yuicompressor <<'EOF'
java -jar /usr/local/libexec/yuicompressor-2.4.8.jar $@
EOF

chmod +x /usr/local/bin/yuicompressor
```

```
$ yuicompressor

YUICompressor Version: 2.4.8

Usage: java -jar yuicompressor-2.4.8.jar [options] [input file]

Global Options
  -V, --version             Print version information
  -h, --help                Displays this information
  --type <js|css>           Specifies the type of the input
file
  --charset <charset>       Read the input file using <charset>
  --line-break <column>     Insert a line break after the
specified column number
  -v, --verbose             Display informational messages and
warnings
  -o <file>                 Place the output into <file>.
Defaults to stdout.
                            Multiple files can be processed
using the following syntax:
                            java -jar yuicompressor.jar -o
```

```
'.css$:-min.css' *.css
                            java -jar yuicompressor.jar -o
'.js$:-min.js' *.js

JavaScript Options
  --nomunge                 Minify only, do not obfuscate
  --preserve-semi           Preserve all semicolons
  --disable-optimizations   Disable all micro optimizations

If no input file is specified, it defaults to stdin. In this
case, the 'type'
option is required. Otherwise, the 'type' option is required
only if the input
file extension is neither 'js' nor 'css'.
```

第 7 章 CSS Frameworks

```
<style>
        html{filter:gray;}

html{filter:progid:DXImageTransform.Microsoft.BasicImage(graysc
ale=1);}
</style>
```

1. 浏览器判断

```
<!--[if IE 8]>
  <link rel="stylesheet" type="text/css" href="ie8.css">
<![endif]-->
<!--[if IE 7]>
  <link rel="stylesheet" type="text/css" href="ie7.css">
<![endif]-->
<!--[if IE 6]>
  <link rel="stylesheet" type="text/css" href="ie6.css">
<![endif]-->
```

```
<!--[if lt IE 7 ]><html class="ie6" lang="zh-cn"><![endif]-->
<!--[if IE 7 ]><html class="ie7" lang="zh-cn"><![endif]-->
<!--[if IE 8 ]><html class="ie8" lang="zh-cn"><![endif]-->
<!--[if IE 9 ]><html class="ie9" lang="zh-cn"><![endif]-->
```

2. Sass: Syntactically Awesome Style Sheets

http://sass-lang.com/

```
gem install sass
```

2.1.

3. Less

http://www.lesscss.net/

第 8 章 stylesheet

4. css 冗余/废弃样式检查

https://code.google.com/archive/p/css-redundancy-checker/

```
wget https://storage.googleapis.com/google-code-archive-
source/v2/code.google.com/css-redundancy-checker/source-
archive.zip
unzip source-archive.zip
gem install hpricot
ruby css-redundancy-checker.rb [cssfile] [directory of html
files OR .txt file listing urls to use]
```

```
# vim url.txt
http://www.netkiller.cn/zh-cn/
http://www.netkiller.cn/index.html
http://www.netkiller.cn/zh-tw/index.html

ruby css-redundancy-checker.rb your.css url.txt
```

第 9 章 HTML

HTML XHTML HTML5

1. iPhone WebApp

移动设备开发

1.1. 拨打电话

禁用电话号码识别

```
<meta name="format-detection" content="telephone=no"  />
```

声明电话链接

```
<a href="tel:13113668890">拨打电话</a>
```

1.2. iphone 图标设置

```
<!DOCTYPE html>
<html manifest="iphone.manifest">
```

```
<head>
    <meta name="viewport" content="user-scalable=no,
width=device-width, initial-scale=1.0, maximum-scale=1.0"/>
    <meta name="apple-mobile-web-app-capable" content="yes" />
    <meta name="apple-mobile-web-app-status-bar-style"
content="black" />
    <link rel="apple-touch-icon" href="apple-touch-icon.png"/>
    <link rel="apple-touch-startup-image" href="apple-touch-
startup-image.png" />
    <link rel="stylesheet" href="iphone.css" type="text/css"
media="screen, mobile" title="main" charset="utf-8">
    <title>offline neo</title>
</head>
<body>
    ...
    ...
    ...
  <script type="text/javascript" src="iphone.js"></script>
</body>
</html>
```

2. frame

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Frames</title>
</head>
<frameset rows="0,100%" frameborder="0">
   <frame name="top" src="" />
   <frame name="main" src="http://www.netkiller.cn/" />
   <noframes>
   <body>
      Your browser does not support frames.
   </body>
   </noframes>
</frameset>
</html>
```

第 10 章 HTML5

1. header

```
<header class="post-header">
    <h1>title</h1>
    <p class="meta"></p>
</header>
```

2. article

```
<article class="post-content">
        content
</article>
```

3. table

```
<table>
    <caption>Council budgets (in £) 2018</caption>
    <thead>
        <tr>
            <th>Items</th>
            <th scope="col">Yorkshire</th>
            <th scope="col">Lancashire</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <th scope="row">Donuts</th>
            <td>3,000</td>
            <td>5,000</td>
        </tr>
        <tr>
            <th scope="row">Stationary</th>
            <td>18,000</td>
            <td>17,000</td>
        </tr>
    </tbody>
    <tfoot>
        <tr>
            <th scope="row">Totals</th>
            <td>21,000</td>
            <td>22,000</td>
        </tr>
    </tfoot>
</table>
```

第 11 章 Javascript

1. window

1.1. window.location

href

```
var source=window.location.href;
if(source.indexOf('www.example.com')>0){
        ...
}
```

hostname

```
if(window.location.hostname == "www.example.com"){
        ...
}
```

```
<span>网址： <script>document.write(window.location.hostname);
</script></span>
```

2. navigator

2.1. userAgent

```
document.write(navigator.userAgent);
document.write(navigator.userAgent.indexOf("MicroMessenger"));
```

```
<script>
var userAgent = window.navigator.userAgent.toLowerCase();
var tags = ["iphone", "android", "phone", "mobile", "wap",
"netfront", "java", "opera mobi", "opera mini", "ucweb",
"windows ce", "symbian", "series", "webos", "sony",
"blackberry", "dopod", "nokia", "samsung", "palmsource", "xda",
"pieplus", "meizu", "midp", "cldc", "motorola", "foma",
"docomo", "up.browser", "up.link", "blazer", "helio", "hosin",
"huawei", "novarra", "coolpad", "webos", "techfaith",
"palmsource", "alcatel", "amoi", "ktouch", "nexian",
"ericsson", "philips", "sagem", "wellcom", "bunjalloo", "maui",
"smartphone", "iemobile", "spice", "bird", "zte-", "longcos",
"pantech", "gionee", "portalmmm", "jig browser", "hiptop",
"benq", "haier", "^lct", "320x320", "240x320", "176x220", "w3c
", "acs-", "alav", "alca", "amoi", "audi", "avan", "benq",
"bird", "blac", "blaz", "brew", "cell", "cldc", "cmd-", "dang",
"doco", "eric", "hipt", "inno", "ipaq", "java", "jigs", "kddi",
"keji", "leno", "lg-c", "lg-d", "lg-g", "lge-", "maui", "maxo",
"midp", "mits", "mmef", "mobi", "mot-",
                                "moto", "mwbp", "nec-", "newt",
"noki", "oper", "palm", "pana", "pant", "phil", "play", "port",
"prox", "qwap", "sage", "sams", "sany", "sch-", "sec-", "send",
"seri", "sgh-", "shar", "sie-", "siem", "smal", "smar", "sony",
"sph-", "symb", "t-mo", "teli", "tim-", "tsm-", "upg1", "upsi",
"vk-v", "voda", "wap-", "wapa", "wapi", "wapp", "wapr", "webc",
"winw", "winw", "xda", "xda-", "Googlebot-Mobile"];

console.log(userAgent);

for (var i = 0; i < tags.length; i++) {
        var tag = tags[i];
        //document.write(tag);
        if(userAgent.indexOf(tag) !== -1) {
                //console.log(tag);
                var hostname = document.location.hostname;
                var domain =
hostname.substring(hostname.lastIndexOf(".",
hostname.lastIndexOf(".") - 1) + 1);
                //document.write(domain);
                document.location = "//m."+domain;
        }
}

</script>
```

3. document

3.1. referrer

referrer

```
javascript:alert(document.referrer);
```

3.2. domain

document.domain;

去掉主机，例如www

```
document.domain.split(".").slice(-2).join(".");
```

4. String 字符串处理

4.1. JSON.parse

```
var json = '{"result":true,"count":1}',
    obj = JSON.parse(json);

alert(obj.count);
```

4.2. replace 替换

正则替换手机号码

```
    var str = "13113668890";
    var res = str.replace(/([0-9]{1,3})([0-9]{1,4})([0-9]
{1,4})/, "$1****$3");
```

5. Date and Time

```
var dateObject=new Date();
document.writeln(dateObject.toDateString());
Mon Mar 28 2016

document.writeln(dateObject.toLocaleDateString());
2016年3月28日

document.writeln(dateObject.toISOString())
2016-03-28T08:57:30.244Z

document.writeln(dateObject.toISOString().slice(0,10));
2016-03-28

document.writeln(dateObject.toISOString().slice(11,19));
09:11:12

document.writeln(dateObject.toTimeString().slice(0,9));
17:16:11

document.writeln(new Date("2016-3-30").getTime());
1459267200000
```

```
var today = new Date();
var h = today.getHours();
var m = today.getMinutes();
var s = today.getSeconds();
```

例 11.1. 倒数计时例子

```
function checkTime(i)
{
   if (i < 10) {
       i = "0" + i;
    }
   return i;
}
```

```
var ts = (new Date("2016-3-30")) - (new Date());//计算剩余的毫秒数
var dd = parseInt(ts / 1000 / 60 / 60 / 24, 10);//计算剩余的天数
var hh = parseInt(ts / 1000 / 60 / 60 % 24, 10);//计算剩余的小时数
var mm = parseInt(ts / 1000 / 60 % 60, 10);//计算剩余的分钟数
var ss = parseInt(ts / 1000 % 60, 10);//计算剩余的秒数
day = checkTime(dd);
hour = checkTime(hh);
minute = checkTime(mm);
second = checkTime(ss);

document.writeln(day + "天" + hour + "时" + minute + "分" + second
+ "秒");
```

6. from 表单相关事件

6.1. onblur

转换为小写字母

```
<input type="text" id="email" name="form.email"
value="${form.email}" onblur="this.value =
this.value.toLowerCase();" maxlength="30" class=""
tabindex="94"/>
```

7. 禁止复制与鼠标右键

```
<script language="JavaScript" type="text/javascript">
document.oncontextmenu=new
Function("event.returnValue=false;");
document.onselectstart=new
Function("event.returnValue=false;");
</script>
```

8. DOMDocument

8.1. createTextNode

```
<SCRIPT>
function fnChangeNode(){
  var oTextNode = document.createTextNode("文本节点已创建");
  var oReplaceNode = oSpan.childNodes(0);
  oReplaceNode.replaceNode(oTextNode);
}
</SCRIPT>

<span ID="oSpan" onclick="fnChangeNode()">
点击此处
</span>
```

9. Microsoft.XMLHTTP

9.1. Get

```
<script type="text/javascript" language="javascript">
    var http_request = false;
    function makeRequest(url) {

        http_request = false;

        if (window.XMLHttpRequest) { // Mozilla, Safari,...
            http_request = new XMLHttpRequest();
            if (http_request.overrideMimeType) {
                http_request.overrideMimeType('text/xml');
            }
        } else if (window.ActiveXObject) { // IE
            try {
                http_request = new
ActiveXObject("Msxml2.XMLHTTP");
            } catch (e) {
                try {
                    http_request = new
ActiveXObject("Microsoft.XMLHTTP");
                } catch (e) {}
            }
        }

        if (!http_request) {
            alert('Giving up :( Cannot create an XMLHTTP
instance');
            return false;
        }
        http_request.onreadystatechange = alertContents;
        http_request.open('GET', url, true);
        http_request.send(null);
    }

    function alertContents() {

        if (http_request.readyState == 4) {
            if (http_request.status == 200) {
                alert(http_request.responseText);
            } else {
                alert('There was a problem with the request.');
            }
        }

    }
</script>
<span
    style="cursor: pointer; text-decoration: underline"
    onclick="makeRequest('http://127.0.0.1/tmp/xml/test.php')">
        Make a request
</span>
```

9.2. POST

```
<script type="text/javascript" language="javascript">
    var http_request = false;
    function makeRequest(url) {

        http_request = false;

        if (window.XMLHttpRequest) { // Mozilla, Safari,...
            http_request = new XMLHttpRequest();
            if (http_request.overrideMimeType) {
                http_request.overrideMimeType('text/xml');
            }
        } else if (window.ActiveXObject) { // IE
            try {
                http_request = new
ActiveXObject("Msxml2.XMLHTTP");
            } catch (e) {
                try {
                    http_request = new
ActiveXObject("Microsoft.XMLHTTP");
                } catch (e) {}
            }
        }
        if (!http_request) {
            alert('Giving up :( Cannot create an XMLHTTP
instance');
            return false;
        }
        http_request.onreadystatechange = alertContents;

        attr = 'name=neo&nickname=netkiller';
        http_request.open('POST', url, true);
        http_request.setRequestHeader ("Content-
Length",attr.length);
                http_request.setRequestHeader ("CONTENT-
TYPE","application/x-www-form-urlencoded");
        http_request.send(attr);

    }

    function alertContents() {

        if (http_request.readyState == 4) {
            if (http_request.status == 200) {
                alert(http_request.responseText);
            } else {
                alert('There was a problem with the request.');
            }
        }

    }
</script>
<input type="text" name="textbox">
<br>
<span
    style="cursor: pointer; text-decoration: underline"
    onclick="makeRequest('http://127.0.0.1/tmp/xml/test.php')">
        Make a request
</span>
```

10. jQuery

过程 11.1.

-

10.1. Selectors(选择器)

```
    if(window.location.hostname.indexOf("example.com") !== -1
){
        $("#nav3").hide();
        $("#platform-nav li:nth-child(3)").hide();
        $("#platform-nav li:nth-child(4)").hide();
        $(".footer .fcon .coll").hide();
        $(".footer .fcon .col3").hide();
        $(".footer .fcon .col4").hide();
    }
```

10.2. jQuery 属性操作

is

```
<a id="startMenu" href="#" class="more">

<section id="menu" class="right_menu disnone">
  <dl>
    <dt><i></i></dt>
    <Dd>您好! 创富金融欢迎您</Dd>
  </dl>
  <ul>
    <li><a href="#">首页</a></li>
    <li><a href="#">简介</a></li>
    <li><a href="#">...</a></li>
    <div style="clear:both;"></div>
  </ul>
</section>

<script>
        $(document).ready(function() {

                $("#startMenu").click(function() {
                        if($("#menu").is(":visible")){
                                $("#menu").hide();
                        }else{
                                $("#menu").show();
                        }

                });
        });
</script>
```

css

```



$("button").click(function(){
  $("p:first").addClass("intro");
});




$( "p" ).removeClass( "myClass yourClass" )
$( "p" ).removeClass( "myClass noClass" ).addClass(
"yourClass" );


<p>hello</p>

<p id="hello">hello</p>
<script type="text/javascript">

$("p").addClass("Helloworld");
$("#hello").addClass("Helloworld");
```

```
</script>
```

10.3. 时间触发

setTimeout 定时执行一次

```
$(document).ready(function(){
        setTimeout(function(){
                $("#error").hide();
        },3000);
});
```

setInterval 间隔执行

```
$(document).ready(function(){
        setInterval(function(){
                alert("test");
        },3000);
});
```

10.4. text

```
<p>hello</p>

<p id="hello">hello</p>
<script type="text/javascript">

$("p").text("Helloworld");
$("#hello").text("Helloworld");
```

```
</script>
```

10.5. inArray

返回值是数组的key

```
        var host = window.location.hostname;
        var domains = 
["netkiller.github.io","www.netkiller.cn"];

        if(jQuery.inArray( host, domains ) != -1) {
        ...
        ...
        }
```

10.6. Ajax

Load

GET

```
jQuery.ajax({
        type:"GET",
        url: "/path/to/url",
        data: "code="+code,
        success:function(data){
                if(data.status){
                        alert(data.text)
                }
        },
        error: function(){
        }
});
```

Post

jsonp

```
<script
src="http://ajax.googleapis.com/ajax/libs/jquery/1.3.2/jquery.m
in.js"></script>
<script>
$.getJSON('http://www.foobar.com/json.php?callback=?',
function(data){
alert(data.foo);
});
</script>

<?php
echo $_GET['callback'], '(', json_encode(array('foo' =>
'bar')), ')';
?>
```

```
// Using YQL and JSONP
$.ajax({
    url: "http://query.yahooapis.com/v1/public/yql",

    // The name of the callback parameter, as specified by
the YQL service
    jsonp: "callback",
    // Tell jQuery we're expecting JSONP
    dataType: "jsonp",

    // Tell YQL what we want and that we want JSON
    data: {
        q: "select title,abstract,url from search.news where 
query=\"cat\"",
        format: "json"
    },

    // Work with the response
    success: function( response ) {
        console.log( response ); // server response
    }
});
```

No 'Access-Control-Allow-Origin' header is present on the requested resource.

原因是 ajax 跨域请求造成

将 dataType: 'JSON' 替换为 dataType: 'JSONP'

同步 AJAX

Jquery ajax 请求默认是异步方式，通过 async: false, 参数修改为同步模式。

```
        function exchange(money){
                        var amount = 0;
                        jQuery.ajax({
                                type: "GET",
                                url: "ajax.php?money=" + money,
                                dataType: "json",
                                async: false,
                                data: "",
                                success: function (data) {
                                        if (data.amount) {
                                                amount =
data.amount
                                        }
                                },
                                error: function () {
                                }
                        });
                return amount;
                }

```

10.7. Form 表单处理

select

```


        var select = $('#bank');
        //$('option', select).remove();
        $.each(banklist, function(key,code) {
                var option = new Option(key, code)
            select.append( option );
        });

        $('#bank').append($("<option>
</option>").attr("value","CMB").text("招商银行"));


```

input

设置 value

```
$("#amount").val("100");
```

10.8. Jquery 事件

click 事件

```
$(document).ready(function() {

        $("#Button1").click(function() {
                $("#Tab1").hide();
                $("#Tab2").show();
        });
        $("#Button2").click(function() {
                $("#Tab1").show();
                $("#Tab2").hide();
        });
});
```

解除事件绑定

```
$( "#Button1").unbind( "click" );
$( "#Button2").unbind( "click" );
```

事件中绑定事件

```
$("#Button").click(function() {
        $( "#Button1").unbind( "click" );
        $( "#Button2").unbind( "click" );

        $("#Button1").click(function() {
                $("#Tab1").hide();
                $("#Tab2").show();
        });
        $("#Button2").click(function() {
                $("#Tab1").show();
                $("#Tab2").hide();
        });
});
```

10.9. Garlic.js - 表单数据持久化

http://garlicjs.org/

Garlic.js 可以让你自动的持久化表单中的数据到本地，直到表单被提交。这样用户就不用担心因为误操作导致表单输入的数据丢失。

使用方法很简单：

```
<script type="text/javascript">
  $( 'form' ).garlic();
</script>
```

11. Bootstrap

http://twbs.github.io/bootstrap/

12. ActiveWidgets - WebUI

13. Highslide

http://highslide.com/

Highslide JS is an image, media and gallery viewer written in JavaScript.

14. JavaScript 代码混淆

14.1. JavaScript Packer

http://joliclic.free.fr/php/javascript-packer/index.php

15. phantomjs - headless WebKit with JavaScript API

16. Javascript MVC Frameworks

http://codebrief.com/2012/01/the-top-10-javascript-mvc-frameworks-reviewed/

16.1. Backbone

http://backbonejs.org/

16.2. example

http://todomvc.com/

第 12 章 SSI

1. SSI 环境变量

显示所有环境变量

```
<!--#printenv -->
```

```
HTTP_USER_AGENT=curl/7.29.0
HTTP_ACCEPT=*/*
LAST_MODIFIED=Tuesday, 03-Nov-2015 09:57:28 HKT
DOCUMENT_URI=/
REMOTE_PORT=37482
SERVER_NAME=224.25.22.70
SERVER_SOFTWARE=Apache Tomcat/7.0.65 OpenJDK 64-Bit Server VM/20.0-b12
Linux
SCRIPT_FILENAME=/srv/apache-tomcat/webapps/ROOT/index.html
DATE_LOCAL=Tuesday, 03-Nov-2015 09:57:31 HKT
SERVER_ADDR=224.25.22.70
SERVER_PROTOCOL=HTTP/1.1
REQUEST_METHOD=GET
DOCUMENT_NAME=
SERVER_PORT=8080
SCRIPT_NAME=/index.html
REMOTE_ADDR=202.130.11.34
DATE_GMT=Tuesday, 03-Nov-2015 01:57:31 GMT
REMOTE_HOST=202.130.101.34
HTTP_HOST=224.25.22.70:8080
QUERY_STRING=
GATEWAY_INTERFACE=CGI/1.1
org.apache.catalina.ssi.SSIServlet=true
REQUEST_URI=/
```

1.1. QUERY_STRING GET参数传递

例如我们需要实现一个功能，test.html?后面的参数需要传递到页面中。

```
http://www.netkiller.cn/lp/test.html?
```

```
utm_source=ss&utm_medium=baidusem&utm_campaign=lpgrant
```

```
<a href="<!--#echo var="WWW_URL"-->/customer/CreateAccount.do?<!--#echo
var="QUERY_STRING"-->">新建用户</a>
```

1.2. SERVER_NAME 与 HTTP_HOST

```
server {
    listen      80;
    listen              443 ssl http2;
    server_name api.netkiller.com api.neo.com api.chen.com;
}
```

SERVER_NAME 如果一个主机配置多个域名，那么SERVER_NAME是域名列表中的第一个域名api.netkiller.com

HTTP_HOST 是当前进入网站的域名

2. set

设置环境变量

```
<!--#set var="foo" value="Bar" -->
<!--#echo var="foo"-->
```

环境变量

```
<!--#set var="WWW_URL"      value="//${SERVER_NAME}"-->
<!--#set var="IMG_URL"      value="//${SERVER_NAME}"-->
```

3. echo

显示环境变量

```
<!--#echo var="SERVER_NAME"--> <br />
<!--#echo var="DOCUMENT_URI"--> <br />
<!--#echo var="HTTP_HOST"--> <br />
<!--#echo var="SERVER_PORT"--> <br />
```

默认值

```
<!--# echo var="name" default="neo" -->
```

禁止编码数据，例如下面LIVE800_URL显示出来后&会被编码.

```
<!--#set var="LIVE800_URL" value="//${SERVER_NAME}/index.jsp?
id=111&pid=122&cid=222"-->
<!--#echo var='LIVE800_URL' default='' encoding='none'-->
```

4. 包含网页

```
<!--#include virtual="file-name" -->
```

包含一个配置文件

```
<!--# if
expr="${SERVER_NAME}=/^(www|images|info|myid|ad).example.com.*/
" -->
        <!--#include file="/include/cn/config.html"-->
<!--# else -->
        <!--#include virtual="/include/cn/config.html"-->
<!--# endif -->
```

5. if 条件判断

```
<!--# if expr="$name" -->
        <!--# echo var="name" -->
<!--# else -->
        netkiller
<!--# endif -->
```

```
<!--#config timefmt="%A" -->
<!--#if expr="$DATE_LOCAL = /Monday/" -->
<p>Meeting at 10:00 on Mondays</p>
<!--#elif expr="$DATE_LOCAL = /Friday/" -->
<p>Turn in your time card</p>
<!--#else -->
<p>Yoga class at noon.</p>
<!--#endif -->
```

```
<!--#if
expr="${SERVER_NAME}=/^(www|images|info|myid|ad).mydomain.com.*
/" -->

    <!--#set var="WWW_URL"
value="http://www.mydomain.com"-->
    <!--#set var="NEWS_URL" value="http://news.mydomain.com"-->
    <!--#set var="IMG_URL"
value="http://img.mydomain.com"-->
    <!--#set var="JS_URL"
value="http://img.mydomain.com/js"-->
    <!--#set var="CSS_URL"
value="http://img.mydomain.com/css"-->

<!--#else -->
```

```
...
...

<!--#endif -->


<!--#if expr="${DOCUMENT_URI}=/\/cn\/.*/"-->
        <!--#set var="LANG" value="cn"-->
<!--#elif expr="${DOCUMENT_URI}=/\/tw\/.*/"-->
        <!--#set var="LANG" value="tw"-->
<!--#elif expr="${DOCUMENT_URI}=/\/en\/.*/"-->
        <!--#set var="LANG" value="en"-->
<!--#endif-->


<!--# if expr="${SERVER_NAME}=/.*.example.com/" -->
    <!--#set var="WWW_URL"      value="//www.example1.com"-->
    <!--#set var="CSS_URL"      value="//css.example1.com"-->
    <!--#set var="IMG_URL"      value="//img.example1.com"-->
<!--# else -->
    <!--#set var="WWW_URL"      value="//www.example.com"-->
    <!--#set var="IMG_URL"      value="//img.example.com"-->
<!--# endif -->
```

判断HTTP与HTTPS

```


<!--#set var="HTML_HOST" value="http://www.example.com"-->
<!--#set var="INFO_HOST" value="http://info.example.com"-->
<!--#set var="NEWS_HOST" value="http://news.example.com"-->

<!--#if expr="${SERVER_PORT}=/443/"-->

<!--#set var="MYID_HOST" value="https://myid.example.com"-->
<!--#set var="IMG_HOST"
value="https://myid.example.com/images"-->
<!--#set var="JS_HOST" value="https://myid.example.com/images"-
->
<!--#set var="CSS_HOST"
value="https://myid.example.com/images"-->

<!--#else -->
```

```
<!--#set var="MYID_HOST" value="http://myid.example.com"-->
<!--#set var="IMG_HOST" value="http://images.example.com"-->
<!--#set var="JS_HOST" value="http://images.example.com"-->
<!--#set var="CSS_HOST" value="http://images.example.com"-->

<!--#endif -->

<!--#set var="IMAGE_POST_HOST" value="http://card-
up.example.com:4141"-->
<!--#set var="IMAGE_UPLOAD_HOST" value="http://card-
look.example.com:4242"-->


<!--#if expr="${DOCUMENT_URI}=/\/cn\/.*/"-->
        <!--#set var="LANG" value="cn"-->
<!--#elif expr="${DOCUMENT_URI}=/\/tw\/.*/"-->
        <!--#set var="LANG" value="tw"-->
<!--#elif expr="${DOCUMENT_URI}=/\/en\/.*/"-->
        <!--#set var="LANG" value="en"-->
<!--#endif-->
```

判断是否经过反向代理

```
<!--#if expr="${X_FORWARDED_FOR}"-->

<!--#set var="IMG_HOST" value="/images"-->
<!--#set var="JS_HOST" value="/images"-->
<!--#set var="CSS_HOST" value="/images"-->

<!--#else -->

<!--#set var="IMG_HOST" value="http://images.example.com"-->
<!--#set var="JS_HOST" value="http://images.example.com"-->
<!--#set var="CSS_HOST" value="http://images.example.com"-->

<!--#endif -->
```

&& 操作

```
<!--#if expr="(${HTTP_USER_AGENT} = /Mozilla\/4/) &&
(${HTTP_USER_AGENT} != /MSIE/)" -->
 Netscape styles
<!--#elif expr="(${HTTP_USER_AGENT} = /Mozilla\/4/) &&
(${HTTP_USER_AGENT} = /MSIE/)" -->
MSIE styles
<!--#else -->
You must be using Opera or other?
<!--#endif -->
```

6. FAQ 常见问题

6.1. SERVER_NAME 与 HTTP_HOST 有什么不同?

SERVER_NAME 与 HTTP_HOST 有什么不同，下面是nginx配置:

```
server {
    listen       80 ;
    server_name www.example.com example.com www.netkiller.cn;

    charset utf-8;
    access_log  /var/log/nginx/www.example.com.access.log;
    error_log  /var/log/nginx/www.example.com.error.log;

    if ($query_string = "") {
       set $args "";
    }

    location / {
        root /www/example.com/www.example.com;
        index index.html;
        }
}
```

当你使用上面的域名访问服务器时 SERVER_NAME 取到的永远是 server_name 配置的第一个域名，即：www.example.com

而 HTTP_HOST 是你浏览器URL上面的域名

第 13 章 Theme & UI

1. bootstrap

http://twitter.github.com/bootstrap/index.html

第 14 章 3rd party

1. Share Buttons

Share Buttons, Share Plugin, Share Analytics, Media Solutions

http://sharethis.com

2. discussions

http://disqus.com/

3. Highlight

3.1. SyntaxHighlighter

http://alexgorbatchev.com/SyntaxHighlighter/

3.2. highlight.js

https://highlightjs.org

4. 所见即所得现在编辑工具

4.1. FCKeditor

4.2. NicEdit

4.3. TinyMCE

4.4. WYSIWYG

4.5. Quill

http://quilljs.com/

第 15 章 Div+CSS页面设计

最近几年，随着业界越来越关注XHTML+CSS的标准化设计，一个新兴职业已经诞生，这就是“网站重构师”，这个新兴职业人才紧缺，他们主要的职责是将HTML+Table+Javascript的架构向XHTML+CSS+Ajax迁移。

1. 页面元素命名

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8" />
        <title>恒信贵金属报表系统</title>
    </head>
    <body>
        <div id="header">
                <div id="logo"></div>
                <div id="banner"></div>
                <div id="navigation"></div>
        </div>
        <div id="wrapper">
                <div id="sidebar"></div>
                <div id="content"></div>
        </div>
        <div id="footer">
                <div id="footer-nav"></div>
                <div id="copyright"></div>
                <div id="legal"></div>
        </div>
    </body>
<html>
```

2. XHTML+DIV+CSS

为何使用表格排版是不明智的选择？为什么要选择DIV+CSS？

首选我来说说表格排版，表格排版也是有好处的，一是排版速度快，二是兼容性比CSS好。做为一般的小网站还是比较适合的，如果在大型网站使用表格就不太合适。 表格必须定义很多属性如width="100%" border="0" cellpadding="0" cellspacing="0"，并且有时候tr标签显得多余。

例 15.1. 表格排版范例

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=utf-8" />


<title>Table Example</title>
</head>

<body>
<table width="100%" border="0" cellspacing="0"
cellpadding="0">
  <tr>
    <td>Logo</td>
    <td>Banner</td>
  </tr>
</table>
<table width="100%" border="0" cellspacing="0"
cellpadding="0">
  <tr>
    <td>Home</td>
    <td>News</td>
    <td>Contact</td>
    <td>About</td>
  </tr>
</table>
<table width="100%" border="0" cellspacing="0"
cellpadding="0">
  <tr>
    <td><table width="100%" border="0" cellspacing="0"
cellpadding="0">
      <tr>
        <td>Top 10 </td>
      </tr>
      <tr>
        <td>xxxxxxxx</td>
      </tr>
      <tr>
        <td>xxxxxxxx</td>
      </tr>
      <tr>
        <td>xxxxxxxx</td>
      </tr>
      <tr>
        <td>xxxxxxxx</td>
      </tr>
    </table>
      <table width="100%" border="0" cellspacing="0"
cellpadding="0">
        <tr>
          <td>Link </td>
        </tr>
        <tr>
          <td>xxxxxxxx</td>
        </tr>
        <tr>
          <td>xxxxxxxx</td>
        </tr>
        <tr>
          <td>xxxxxxxx</td>
        </tr>
        <tr>
          <td>xxxxxxxx</td>
        </tr>
      </table></td>
    <td><table width="100%" border="0" cellspacing="0"
cellpadding="0">
      <tr>
```

```
        <td align="center">Article Title </td>
      </tr>
      <tr>
        <td><p>Contect</p>

<p>XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXX</p>

<p>XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXX</p>
          <p>XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX</p>
          <p>XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX</p></td>
      </tr>
      <tr>
        <td>Feedback</td>
      </tr>
    </table></td>
  </tr>
</table>
<table width="100%" border="0" cellspacing="0"
cellpadding="0">
  <tr>
    <td align="center">Copyright XXXX </td>
  </tr>
</table>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
</body>
</html>
```

你可以对比上面看看div+css是如何规划版面，并且css很多定义是可以重用的。

例 15.2. XHTML+DIV+CSS排版范例

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
```

```
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=utf-8" />
<title>hello world</title>
<style type="text/css">
<!--
body{
        width: 795px;
}
div1{
        border-color: #119EBA;
        border-width: 1px;
        border-style: solid;
        margin: 5px;
}
#header{
}
#logo, #banner{
        float:left;
        height: 75px;
}
#nav{
        clear:both;
}
#nav ul {
        list-style-type:none;
        margin: 0px;
        padding:0px;
}
#nav ul li{
        float:left;
        width: 100px;
}
#main{ clear:both;}
#main #left {
        float:left;
        width: 30%;
}
#main #right {
        float:right;
        width: 70%;
}
.box{}
```

```
.box h2 {
        margin: 0;
        padding: 0px;
}
.box a { display:block;}
#footer{ clear:both}
#footer #copyright {
        text-align:center;
}

.article {
        border-color: black;
        border-width: 1px;
        border-style: solid;
        margin: 5px;
        padding: 10px;
}
.article .article_title{
        font-size: 24px;
        font-weight:bold;
        text-align:center;
}
.article .article_content{ font-size:10px;}
-->
</style>
</head>

<body>

<div id="header">
        <div id="logo"> Logo </div>
        <div id="banner"> Banner </div>
</div>

<div id="nav">
        <ul>
                <li><a href="#"> Home </a></li>
                <li><a href="#"> News </a></li>
                <li><a href="#"> Person </a></li>
                <li><a href="#"> Group </a></li>
                <li><a href="#"> Network </a></li>
        </ul>
</div>

<div id="main">
```

```
        <div id="left">
                <div class="box">
                        <h2>title</h2>
                        <a href="#"> link </a>
                        <a href="#"> link </a>
                        <a href="#"> link </a>
                </div>

                <div class="box">
                        <h2>title</h2>
                        <a href="#"> link </a>
                        <a href="#"> link </a>
                        <a href="#"> link </a>
                </div>
        </div>
        <div id="right">
                <div class="article">
                        <div class="article_title">
                                Article Title
                        </div>
                        <div class="article_content">

<p>XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
</p>

<p>XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
</p>

<p>XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
</p>

<p>XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
</p>
                        </div>
                </div>
        </div>
</div>

<div id="footer">
        <div id="copyright"> Copyright Neo Chan</div>
</div>

</body>
</html>
```

上面例子我们可以看到div与table相比所使用的标签更少，无形中给网站减了肥。

CSS 的class,id 名称定义规范:

1. 一定要简单，可读例如header，footer

2. 对于在页面中不重复，自始至终只出现一次可定义为id，例如id="header"，id="footer"

3. 对于在页面中经常重复出现的，可定义为class，例如id="article_block"，id="news_block"

注意

不要使用HTML属性，尽量使用css。 herf,src,class,id等属性除外。

下面是一个例子

```
<font color="red" size="12" face="Arial, Helvetica, sans-serif" >Hello workd</font>
```

你应该使用CSS实现,如果能使用CSS实现尽量不要多用一条HTML和属性。

```
<style type="text/css">
.hello{
        color:red;
        font-size:12px;
        font-family:Arial, Helvetica, sans-serif;
}
</style>
<div id="hello">Hello workd</font>
```

3. 页面结构设计

页面结构从上到下依次是

- header 主要包括导航，登录，Logo, Banner
- body 网站主要内容，并且还可以分为左右两栏，左中右三栏。
- footer 导航，版权

header,footer将显示在所有页面，一般很少改动。

3.1. Home page (首页)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=utf-8" />
<title>Title</title>
</head>

<body>

<div id="header">
        <div id="logo"></div>
    <div id="banner"></div>
    <div id="nav"></div>
</div>

<div id="page1">
        <div class="left_nav"></div>
    <div class="right"></div>
</div>
```

```
<div id="page2">

</div>

<div id="page3">

</div>

<div id="footer">
  <div id="footer_nav"></div>
  <div id="copyright"></div>
</div>

</body>
</html>
```

page1 打开首页看到的第一屏页面，page2，page3需要按翻页键

不要将 page1，page2，page3 放到一个DIV中

3.2. 导航烂

```
<table>
        <tr>
                <td>Home</td>
                <td>News</td>
                <td>About</td>
                <td>Contact</td>
        </tr>
</table>
```

```
<div id="nav">
        <ul>
                <li><a href="#"> Home </a></li>
                <li><a href="#"> News </a></li>
                <li><a href="#"> Person </a></li>
                <li><a href="#"> Group </a></li>
                <li><a href="#"> Network </a></li>
        </ul>
</div>
```

3.3. Left Bar

```
        <div id="left">
                <div class="box">
                        <h2>title</h2>
                        <a href="#"> link </a>
                        <a href="#"> link </a>
                        <a href="#"> link </a>
                </div>

                <div class="box">
                        <h2>title</h2>
                        <a href="#"> link </a>
                        <a href="#"> link </a>
                        <a href="#"> link </a>
                </div>
        </div>
```

3.4. 区块设计 Block

网站经常用一些方块规划版面。

- 一种是矩形方框
- 另一种是有标题，标题下方是矩形方框
- 现在流行的是标题栏有多个选项卡，标题下方是矩形方框，当选择不同标题时，矩形方框中的内容随之改变。

传统方法如下:

例 15.3. 例子

table block example

```
<table>
        <tr>
                <td>
                        内容
                </td>
        </tr>
</table>
```

div+css block example

```
<div class="simple_box">
                        内容
</table>
```

例 15.4. 例子

table title block example

```
<table>
        <tr>
                <td>Top 10</td>
        </tr>
        <tr>
                <td>
                        <table>
```

```
                                <tr>
                                        <td>No.1</td>
                                </tr>
                                <tr>
                                        <td>No.2</td>
                                </tr>
                                <tr>
                                        <td>No.3</td>
                                </tr>
                                <tr>
                                        <td>No.n</td>
                                </tr>
                        </table>
                </td>
        </tr>
</table>
```

div+css title block example

```
<div class="title_block">
        <h2>
                Title
        <h2>
        <div>
                Content
        </div>
</div>
```

使用dl标记实现

```
<dl class="title_block">
        <dt>Title<dt>
        <dd>
                Content
        </dd>
</dl>
```

4. 表格

这里的表格不是指用于排版，而是表格数据。

```
<style type="text/css">
.hello{
        width: 100%;
}
.hello tr{

}
.hello td{

}
</style>
<table class="mytable">
        <tr>
                <td></td>
        </tr>
</table>
```

5. 图片优化

5.1. onMouseOver/onMouseOut

我们在网站冲浪常常看会看到很多图片按钮，当鼠标入上去或鼠标移开图片会随之改变，这个的按钮至少需要三张小图片才能实现这样的功能。

我先说说这样做的缺点

- 三张图片，你的浏览器会启动三个线程链接你的图片服务器，不划算。
- 一旦其中一幅图片下载过程中中断，用户当把鼠标放到按钮上时，可能会出现一个红叉叉。
- 图片太多不好维护，易产生垃圾，占用磁盘空间，linux ext3 一个空文件占用2048

最优方法是使用一张图片，将三幅图片平行或垂直排开，放到一幅图片中，然后使用CSS控制显示你需要的部分。

5.2. 使用一幅图片处理BLOCK四角

corner.gif

stylesheet

```


<style type="text/css">
<!--

.clear { clear: both; height: 0; font-size: 0; line-height:
0; zoom: 1 }
```

```
.containerPlain {
        background-color: #fff;
        border-right: 1px solid #cacaca;
        border-left: 1px solid #cacaca;
        padding: 0 3px;
}

.left_top_corner {
        display: block;
        width: 4px;
        height: 4px;
        background-position: top left;
        float: left;
        font-size: 0;
}

.right_top_corner {
        display: block;
        width: 4px;
        height: 4px;
        background-position: top right;
        float: right;
        font-size: 0;
}

.left_bottom_corner {
        display: block;
        width: 4px;
        height: 4px;
        background-position: bottom left;
        float: left;
        font-size: 0;
}


.right_bottom_corner {
        display: block;
        width: 4px;
        height: 4px;
        background-position: bottom right;
        float: right;
        font-size: 0;
}
.left_bottom_corner, .right_bottom_corner ,
.left_top_corner, .right_top_corner{
        background-image: url(corners/corner.gif);
}

.middle_top_line {
        display: block;
        float: left;
        height: 3px;
        line-height: 0;
        border-top: 1px solid #cacaca;
}

.middle_bottom_line {
        display: block;
        float: left;
        height: 3px;
        border-bottom: 1px solid #cacaca;
        font-size: 0;
}

.middle_top_line, .middle_bottom_line {
                width: 167px;
}

-->
</style>
```

HTML

```
<div style="width:175px;">
        <span class="left_top_corner"></span> <span
class="middle_top_line"></span> <span
class="right_top_corner"></span>
        <div class="containerPlain">
        You Content
        </div>
        <span class="left_bottom_corner"></span> <span
class="middle_bottom_line"></span> <span
class="right_bottom_corner"></span>
</div>
```

下面是一个更复杂的例子

- corner.gif
- block_title_left.gif
- block_title_right.gif

stylesheet

```
<style type="text/css">
<!--

.clear { clear: both; height: 0; font-size: 0; line-height:
0; zoom: 1 }

.containerPlain {
        background-color: #fff;
        border-right: 1px solid #cacaca;
        border-left: 1px solid #cacaca;
        padding: 0 3px;
        clear: both;
}

.left_top_corner {
        display: block;
        width: 4px;
        height: 4px;
        background-position: top left;
        float: left;
        font-size: 0;
}

.right_top_corner {
        display: block;
        width: 4px;
        height: 4px;
        background-position: top right;
        float: right;
        font-size: 0;
}

.left_bottom_corner {
        display: block;
        width: 4px;
        height: 4px;
        background-position: bottom left;
        float: left;
        font-size: 0;
}


.right_bottom_corner {
        display: block;
        width: 4px;
        height: 4px;
        background-position: bottom right;
        float: right;
        font-size: 0;
}
.left_bottom_corner, .right_bottom_corner ,
.left_top_corner, .right_top_corner{
        background-image: url(corners/corner.gif);
}

.middle_top_line {
        display: block;
        float: left;
        height: 3px;
        line-height: 0;
        border-top: 1px solid #cacaca;
}

.middle_bottom_line {
        display: block;
        float: left;
        height: 3px;
        border-bottom: 1px solid #cacaca;
        font-size: 0;
}
```

```
.middle_top_line, .middle_bottom_line {
                width: 167px;
}


.block_title {
        line-height: 26px;
        height: 26px;
        background-image: url(corners/block_title_left.png);
        background-repeat: no-repeat;
        padding-left: 10px;
        font-size: 13px;
        font-weight: bold;
        background-color: #dddbdc;
}

.block_title_right {
        display: block;
        background-image: url(corners/block_title_right.png);
        background-repeat: no-repeat;
        background-postition: right;
        float: right;
        width: 4px;
        height: 26px;
}
-->
</style>
```

HTML

```
<div style="width:175px;">
  <span class="left_top_corner"></span> <span
class="middle_top_line"></span> <span
class="right_top_corner"></span>
  <div class="containerPlain">
    <div class="block_title">
                <span class="block_title_right"></span> Title
    </div>
    <div style="padding: 10px 7px 7px 7px">
```

```
                Content
        </div>
  </div>
  <span class="left_bottom_corner"></span> <span
class="middle_bottom_line"></span> <span
class="right_bottom_corner"></span>
</div>
```

5.3. 图片用背景图代替 img 标记

```
图片用背景图代替<img src="">
```

5.4. 合并图片

下面是摘取LinkedIn网页,作为例子.

合并多张小图片为一张图片,然后通过偏移量取其中一部分显示,这样做的目的是,加快浏览器下载速度,降低与服务器建立连接的开销.

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" dir="ltr"
lang="en-US">
<head profile="http://gmpg.org/xfn/11">
        <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
        <title>LinkedIn Blog</title>
        <style type="text/css">
/*
Theme Name:  LinkedIn Blog
Theme URI:   http://blog.linkedin.com/
Description: LinkedIn's main blog theme
```

```
Author:       Prajakta Godbole
Author URI:   http://linkedin.com/
Version:      2.0
*/

/*
Reset styles
Copyright (c) 2011, Yahoo! Inc. All rights reserved.
Code licensed under the BSD License:
http://developer.yahoo.com/yui/license.html
version: 2.9.0
*/
html{color:#000;background:#FFF}body,div,dl,dt,dd,ul,ol,li,h1
,h2,h3,h4,h5,h6,pre,code,form,fieldset,legend,input,button,te
xtarea,select,p,blockquote,th,td{margin:0;padding:0}table{bor
der-collapse:collapse;border-
spacing:0}fieldset,img{border:0}address,button,caption,cite,c
ode,dfn,em,input,optgroup,option,select,strong,textarea,th,va
r{font:inherit}del,ins{text-decoration:none}li{list-
style:none}caption,th{text-align:left}h1,h2,h3,h4,h5,h6{font-
size:100%;font-
weight:normal}q:before,q:after{content:''}abbr,acronym{border
:0;font-variant:normal}sup{vertical-
align:baseline}sub{vertical-align:baseline}legend{color:#000}

/* Colors and fonts */
html { background-color: #F5F5F5; }
body { font-family: Arial, Helvetica, "Nimbus Sans L", sans-
serif; padding-top: 20px;}
a { color: #006fb3; text-decoration: none; }
a:hover { color: #006fb3; text-decoration: underline;}

/* Sidebar */
#sidebar { width: 312px; float: left; margin-left: 20px;}
#sidebar .widgets { border: 1px solid #ddd; background-color:
#FFF; margin-bottom: 50px;
        -webkit-border-radius: 5px; -moz-border-radius: 5px;
border-radius: 5px;}
#sidebar .widgets h2 { color: #4d4e54; font-size: 14px;
clear: both; margin-bottom: 13px;}
#sidebar .widgets ul li { font-size: 11.5px; }
#sidebar .widgets ul li a { color: #4d4e54; }
#sidebar .widgets .widget-bg { position: absolute; top:
-13px; right: 15px; width: 35px; height: 40px; }
```

```
/* Follow us list */
#sidebar .follow-us-widget { overflow: hidden; padding-
bottom: 35px; border-bottom: 1px solid #ccc; }
#sidebar .follow-us-widget .widget-wrapper { padding: 15px;}
#sidebar ul#follow-list li { float: left; position: relative;
margin-right: 17px; zoom: 1; display: inline;}
#sidebar ul#follow-list li:last-child { margin-right: 0;}
#sidebar ul#follow-list li .follow-div { margin:0; padding:0;
width: 33px; height: 38px; }
#sidebar ul#follow-list li a { display: block; width: 33px;
height: 38px; text-indent: -9999px; background:
url('http://blog.linkedin.com/wp-
content/themes/linkedin/images/sprite3.png');}
#sidebar ul#follow-list li a#follow-lnkd { background-
position: 0 0; }
#sidebar ul#follow-list li a#follow-twtr { background-
position: -33px 0; }
#sidebar ul#follow-list li a#follow-fb { background-position:
-66px 0; width: 32px; }
#sidebar ul#follow-list li a#follow-flickr { background-
position: -130px 0; width: 32px;}
#sidebar ul#follow-list li a#follow-youtube { background-
position: -98px 0; width: 32px;}
#sidebar ul#follow-list li a#follow-rss { background-
position: -162px 0; width: 32px; }
#sidebar .widgets ul#follow-list li.last { margin-right: 0;}

/* Flickr */
#sidebar .flickr-widget { position: relative; border-bottom:
1px solid #ccc; }
#sidebar .flickr-widget .widget-wrapper { padding: 15px;}
#sidebar .flickr-widget h2 { margin-bottom: 20px; }
#sidebar .flickr-widget .widget-bg { background:
url('images/sprite3.png') -267px 0 no-repeat;}
#sidebar #flickr-img-grp { margin-bottom: 10px;
overflow:hidden; }
#sidebar #flickr-img-grp .flickr-img { float: left; margin: 0
15px 15px 0; }
        </style>
</head>
<body>
        <div id="sidebar">

                <div class="widgets">
                        <div class="follow-us-widget">
```

```
                                <div class="widget-wrapper">
                                        <h2>Follow Us
Links</h2>

                                        <ul id="follow-list">
                                                <li><a
id="follow-lnkd"

href="https://www.linkedin.com/company/linkedin"
target="_blank">LinkedIn</a>
                                                </li>
                                                <li><a
id="follow-twtr" href="http://twitter.com/LinkedIn"

target="_blank">Twitter</a></li>
                                                <li><a
id="follow-fb"

href="https://www.facebook.com/LinkedIn"
target="_blank">Facebook</a>
                                                </li>
                                                <li><a
id="follow-youtube"

href="http://www.youtube.com/user/LinkedIn"
target="_blank">YouTube</a>
                                                </li>
                                                <li><a
id="follow-flickr"

href="http://www.flickr.com/groups/linkedin/pool/"

target="_blank">Flickr</a></li>
                                                <li
class="last"><a id="follow-rss"

href="http://feeds.feedburner.com/LinkedInBlog"
target="_blank">RSS</a>
                                                </li>
                                        </ul>

                                </div>
                        </div>

                </div>
```

```
        </div>

</body>
</html>
```

6. HTML嵌入图片

```
<img src="data:image/png;base64," />
```

7. 页面内容安全

7.1. 禁止鼠标右键

修改body标签，加入onContextMenu="return false" onSelectStart="return false"。

```
<body bgColor="#FFFFFF" onContextMenu="return false"
onSelectStart="return false">
```

7.2. 禁止复制剪切 及粘贴

禁止拷贝文字：

```
<input name="textfield" type="text" value="不能复制里面的字"
oncopy="return false;" oncut="return false;" onpaste="return
false">
```

8. html,css 有效性检查 Validation

有效性检查包括

1. Markup Validation
2. CSS Validation.
3. URL Validation.

9. 自适应宽度超出截取并显示省略字符

```
<style type="text/css">
.ellipsis {
    white-space: nowrap;                /*保留文字间的空白*/
    width: 200px;
    overflow: hidden;                   /*超出部分隐藏，*/
    text-overflow: ellipsis;    /*超出部分显示成... */
    border:1px solid #333;
}
</style>

<div class="ellipsis">
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx </div>
```

第 16 章 Angular

https://angularjs.org/

1. Function

1.1.

```
    $scope.getClass = function(a){
|                            <li class="baidu">
            return a;
|                                <a href=""></a>
    }
```

1.2. ng-bind

```
<div>
  Hello, {{user.name}}
</div>

<div>
  Hello, <span ng-bind="user.name"></span>
</div>
```

部分 II. Ruby

第 17 章 Ruby

1. Ubuntu ruby 1.9.1

```
sudo apt-get install ruby1.9.1
```

1.1. rubygems

```
wget http://production.cf.rubygems.org/rubygems/rubygems-
1.6.2.tgz
tar zxvf rubygems-1.6.2.tgz
cd rubygems-1.6.2/
sudo  ruby setup.rb
# gem1.8
```

ubuntu

```
$ sudo apt-get install ruby1.9.1
$ sudo apt-get install rubygems1.8
```

2. sanp 安裝

```
ln -s /var/lib/snapd/snap /snap
snap install ruby --classic
/snap/bin/ruby -v
```

```
PATH=/snap/bin:$PATH
/snap/bin/gem install rails
```

第18章Variable

1. String

2. Date

使用strftime获得具体的年份，月份或日期等等，常用：

```
CODE    OUTPUT  DESCRIPTION
t.strftime("%H")        => "22" # Gives Hour of the time in 24 hour
clock format
t.strftime("%I")        => "10" # Gives Hour of the time in 12 hour
clock format
t.strftime("%M")        => "49" # Gives Minutes of the time
t.strftime("%S")        => "27" # Gives Seconds of the time
t.strftime("%Y")        => "2013"         # Gives Year of the time
t.strftime("%m")        => "09" # Gives month of the time
t.strftime("%d")        => "12" # Gives day of month of the time
t.strftime("%w")        => "4"  # Gives day of week of the time
t.strftime("%a")        => "Thu"        # Gives name of week day in
short form of the
t.strftime("%A")        => "Thursday"   # Gives week day in full form of
the time
t.strftime("%b")        => "Sep"        # Gives month in short form of
the time
t.strftime("%B")        => "September"  # Gives month in full form of
the time
t.strftime("%y")        => "13" # Gives year without century of the time
t.strftime("%Y")        => "2013"       # Gives year without century  of
the time
t.strftime("%Z")        => "IST"        # Gives Time Zone of the time
t.strftime("%p")        => "PM" # Gives AM / PM of the time
```

```
# 2012-03-06 15:28:08
puts Time.now.strftime("%Y-%m-%d %H:%M:%S")

# 03/06/12 03:39 PM
puts Time.now.strftime("%x %I:%M %p")
```

3. Array

第 19 章 Class

1. rmagick

```
sudo gem install rmagick
```

第 20 章 Ruby on Rails

1. Getting Started

```
$ sudo apt-get install ruby1.9.1
$ sudo ln -s /usr/bin/ruby1.9.1 /usr/bin/ruby

$ sudo apt-get install libopenssl-ruby1.9.1
```

```
$ sudo apt-get install rubygems1.9.1
$ sudo ln -s /usr/bin/gem1.9.1 /usr/bin/gem
```

Installing Rails

```
$ sudo apt-get install rails
```

or

```
$ sudo gem install rails

Successfully installed rake-0.8.7
Successfully installed activesupport-2.3.5
Successfully installed activerecord-2.3.5
Successfully installed rack-1.0.1
Successfully installed actionpack-2.3.5
Successfully installed actionmailer-2.3.5
Successfully installed activeresource-2.3.5
Successfully installed rails-2.3.5
8 gems installed
Installing ri documentation for rake-0.8.7...
Installing ri documentation for activesupport-2.3.5...
Installing ri documentation for activerecord-2.3.5...
Installing ri documentation for rack-1.0.1...
Installing ri documentation for actionpack-2.3.5...
Installing ri documentation for actionmailer-2.3.5...
Installing ri documentation for activeresource-2.3.5...
Installing ri documentation for rails-2.3.5...
Updating class cache with 0 classes...
Installing RDoc documentation for rake-0.8.7...
Installing RDoc documentation for activesupport-2.3.5...
Installing RDoc documentation for activerecord-2.3.5...
Installing RDoc documentation for rack-1.0.1...
Installing RDoc documentation for actionpack-2.3.5...
Installing RDoc documentation for actionmailer-2.3.5...
Installing RDoc documentation for activeresource-2.3.5...
Installing RDoc documentation for rails-2.3.5...
```

2. Creating a New Rails Project

export PATH=$PATH:/home/neo/.gem/ruby/1.8/bin

2.1. Creating the Blog Application

```
$ rails blog
      create
      create  app/controllers
      create  app/helpers
      create  app/models
      create  app/views/layouts
      create  config/environments
      create  config/initializers
      create  config/locales
      create  db
      create  doc
      create  lib
      create  lib/tasks
      create  log
      create  public/images
      create  public/javascripts
      create  public/stylesheets
      create  script/performance
      create  test/fixtures
      create  test/functional
      create  test/integration
      create  test/performance
      create  test/unit
      create  vendor
      create  vendor/plugins
      create  tmp/sessions
      create  tmp/sockets
      create  tmp/cache
      create  tmp/pids
      create  Rakefile
      create  README
      create  app/controllers/application_controller.rb
      create  app/helpers/application_helper.rb
      create  config/database.yml
      create  config/routes.rb
      create  config/locales/en.yml
      create  db/seeds.rb
      create  config/initializers/backtrace_silencers.rb
      create  config/initializers/inflections.rb
      create  config/initializers/mime_types.rb
      create  config/initializers/new_rails_defaults.rb
      create  config/initializers/session_store.rb
      create  config/environment.rb
      create  config/boot.rb
      create  config/environments/production.rb
      create  config/environments/development.rb
      create  config/environments/test.rb
      create  script/about
      create  script/console
      create  script/dbconsole
      create  script/destroy
      create  script/generate
      create  script/runner
      create  script/server
      create  script/plugin
      create  script/performance/benchmarker
      create  script/performance/profiler
      create  test/test_helper.rb
      create  test/performance/browsing_test.rb
      create  public/404.html
      create  public/422.html
      create  public/500.html
      create  public/index.html
      create  public/favicon.ico
      create  public/robots.txt
      create  public/images/rails.png
      create  public/javascripts/prototype.js
      create  public/javascripts/effects.js
      create  public/javascripts/dragdrop.js
      create  public/javascripts/controls.js
      create  public/javascripts/application.js
      create  doc/README_FOR_APP
      create  log/server.log
      create  log/production.log
      create  log/development.log
      create  log/test.log
```

2.2. Configuring a Database

default database

```
$ gem install sqlite3-ruby
```

```
$ rails blog -d mysql
or
$ rails blog -d postgresql
```

Creating the Database

```
$ cd blog
$ rake db:create
```

2.3. Hello world

home controller

```
$ script/generate controller home index
      exists  app/controllers/
      exists  app/helpers/
      create  app/views/home
      exists  test/functional/
      create  test/unit/helpers/
      create  app/controllers/home_controller.rb
      create  test/functional/home_controller_test.rb
      create  app/helpers/home_helper.rb
      create  test/unit/helpers/home_helper_test.rb
      create  app/views/home/index.html.erb
```

edit view

```
$ vim app/views/home/index.html.erb
```

2.4. Starting up the Web Server

```
$ script/server
=> Booting WEBrick
=> Rails 2.3.5 application starting on http://0.0.0.0:3000
=> Call with -d to detach
=> Ctrl-C to shutdown server
[2010-05-22 16:24:04] INFO  WEBrick 1.3.1
[2010-05-22 16:24:04] INFO  ruby 1.9.1 (2010-01-10) [x86_64-
linux]
[2010-05-22 16:24:04] INFO  WEBrick::HTTPServer#start:
pid=30711 port=3000
```

2.5. Setting the Application Home Page

The first step to doing this is to delete the default page from your application

```
$ rm -rf public/index.html
or
$ mv public/index.html public/index.html.off
```

```
$ vim config/routes.rb

  map.connect ':controller/:action/:id'
  map.connect ':controller/:action/:id.:format'
  map.root :controller => "home"
```

2.6. FAQ

http://rbjl.net/20-rubybuntu-2-troubleshooting-common-ruby-ubuntu-problems

3. capistrano

第 21 章 FAQ

1. no such file to load — mkmf

```
sudo apt-get install ruby-dev
```

部分 III. Node.js

1. FAQ

1.1. request to https://api.netkiller.cn/v1/getinfo failed, reason: self signed certificate

问题原因是node不是别自签名证书

```
[www@localhost ~]$ export NODE_TLS_REJECT_UNAUTHORIZED='0'
[www@localhost ~]$ pm2 restart project --update-env
```

设置环境变量 NODE_TLS_REJECT_UNAUTHORIZED=0 就可以解决

```
process.env.NODE_TLS_REJECT_UNAUTHORIZED='0'
process.env['NODE_TLS_REJECT_UNAUTHORIZED'] = '0';
```

第 22 章 Node.js 安装

http://nodejs.org/

1. Ubuntu

```
$ sudo apt-get install nodejs
$ sudo apt-get install npm
```

2. Ubuntu Node.js 9.x

```
curl -sL https://deb.nodesource.com/setup_9.x | sudo -E bash -
sudo apt-get install -y nodejs
sudo apt-get install -y build-essential
```

3. CentOS

CentOS 8 Stream

```
dnf install -y nodejs
```

CentOS 7

```
[root@netkiller www]# yum install -y nodejs
[root@netkiller www]# node --version
v6.9.1
```

安装 cnpm

```
[root@gitlab ~]# npm config set registry
https://registry.npm.taobao.org
[root@gitlab ~]# npm config get registry
https://registry.npm.taobao.org/
[root@gitlab ~]# npm install -g cnpm
```

4. Mac

Node

```
neo@MacBook-Pro ~ % brew install node
neo@MacBook-Pro ~ % node -v
v12.1.0
```

Node8

```
neo@MacBook-Pro ~ % brew install node@8

neo@MacBook-Pro ~ % export
PATH="/usr/local/opt/node@8/bin:$PATH"

neo@MacBook-Pro ~ % node -v
v8.16.0
```

5. npm -- node package manager

npm registry

```
$ npm install mysql
```

mirror

```
npm config set registry https://registry.npm.taobao.org
```

```
npm install --registry=https://registry.npm.taobao.org
```

link

```
# npm link gulp
/root/node_modules/gulp -> /srv/node-v7.10.0-linux-
x64/lib/node_modules/gulp
```

6. Yarn

npm 安裝 yarn

```
npm install -g yarn
```

Ubuntu

```
curl -sS https://dl.yarnpkg.com/debian/pubkey.gpg | sudo apt-
key add -
echo "deb https://dl.yarnpkg.com/debian/ stable main" | sudo
tee /etc/apt/sources.list.d/yarn.list

sudo apt update && sudo apt install yarn
```

CentOS

```
curl --silent --location https://dl.yarnpkg.com/rpm/yarn.repo |
sudo tee /etc/yum.repos.d/yarn.repo
curl --silent --location https://rpm.nodesource.com/setup_8.x |
sudo bash -

sudo yum install yarn
```

CentOS 8 Stream

```
curl -sL https://dl.yarnpkg.com/rpm/yarn.repo -o
/etc/yum.repos.d/yarn.repo
dnf install -y yarn
```

安装

7. pm2

Production process manager for Node.js apps with a built-in load balancer http://pm2.io

```
                npm install -g pm2
```

启动进程

```
[ethereum@netkiller web3.mainnet]$ pm2 start restful.v2.js
[PM2] Applying action restartProcessId on app [restful.v2](ids:
0)
[PM2] [restful.v2](0) ✓
[PM2] Process successfully started
┌─────────────┬────┬──────┬───────┬────────┬─────────┬────────┬─
───┬───────────┬──────────┬──────────┐
│ App name    │ id │ mode │ pid   │ status │ restart │ uptime │
cpu │ mem         │ user     │ watching │
├─────────────┼────┼──────┼───────┼────────┼─────────┼────────┼─
───┼───────────┼──────────┼──────────┤
│ restful.v2  │ 0  │ fork │ 11480 │ online │ 17      │ 0s     │
0%  │ 10.3 MB     │ ethereum │ disabled │
└─────────────┴────┴──────┴───────┴────────┴─────────┴────────┴─
───┴───────────┴──────────┴──────────┘
 Use `pm2 show <id|name>` to get more details about an app
```

执行启动程序名称

```
[ethereum@netkiller web3.mainnet]$ pm2 start restful.v2.js --
name restful
[PM2] Applying action restartProcessId on app [restful.v2](ids:
0)
[PM2] [restful.v2](0) ✓
```

```
[PM2] Process successfully started
┌──────────────┬────┬──────┬───────┬────────┬─────────┬────────┬─
─────┬───────────┬──────────┬──────────┐
│ App name     │ id │ mode │ pid   │ status │ restart │ uptime │
cpu  │ mem        │ user     │ watching │
├──────────────┼────┼──────┼───────┼────────┼─────────┼────────┼─
─────┼───────────┼──────────┼──────────┤
│ restful      │ 0  │ fork │ 11480 │ online │ 17      │ 0s     │
0%   │ 10.3 MB    │ ethereum │ disabled │
└──────────────┴────┴──────┴───────┴────────┴─────────┴────────┴─
─────┴───────────┴──────────┴──────────┘
 Use `pm2 show <id|name>` to get more details about an app
```

停止程序运行

```
[ethereum@netkiller web3.mainnet]$ pm2 stop pm2-http-interface
[PM2] Applying action stopProcessId on app [pm2-http-interface]
(ids: 1)
[PM2] [pm2-http-interface](1) ✓
┌─────────────────────┬────┬──────┬───────┬─────────┬─────────┬─
───────┬──────┬────────────┬──────────┬──────────┐
│ App name            │ id │ mode │ pid   │ status  │ restart │
uptime │ cpu  │ mem        │ user     │ watching │
├─────────────────────┼────┼──────┼───────┼─────────┼─────────┼─
───────┼──────┼────────────┼──────────┼──────────┤
│ pm2-http-interface  │ 1  │ fork │ 0     │ stopped │ 0       │ 0
│ 0%   │ 0 B        │ ethereum │ disabled │
│ restful.v2          │ 0  │ fork │ 11480 │ online  │ 17      │
11m    │ 0%   │ 26.6 MB    │ ethereum │ disabled │
└─────────────────────┴────┴──────┴───────┴─────────┴─────────┴─
───────┴──────┴────────────┴──────────┴──────────┘
 Use `pm2 show <id|name>` to get more details about an app
```

查看进程列表

```
[ethereum@netkiller web3.mainnet]$ pm2 list
┌────────────┬────┬──────┬───────┬────────┬─────────┬────────┬─
───┬────────────┬──────────┬──────────┐
│ App name   │ id │ mode │ pid   │ status │ restart │ uptime │
cpu │ mem        │ user      │ watching │
├────────────┼────┼──────┼───────┼────────┼─────────┼────────┼─
───┼────────────┼──────────┼──────────┤
│ restful.v2 │ 0  │ fork │ 11480 │ online │ 17      │ 2m     │
0%  │ 28.1 MB    │ ethereum │ disabled │
└────────────┴────┴──────┴───────┴────────┴─────────┴────────┴─
───┴────────────┴──────────┴──────────┘
 Use `pm2 show <id|name>` to get more details about an app
```

查看启动进程的详细信息

```
[ethereum@netkiller web3.mainnet]$ pm2 show restful.v2
 Describing process with id 0 - name restful.v2
┌───────────────────┬─────────────────────────────────────────────
──────┐
│ status            │ online
│
│ name              │ restful.v2
│
│ restarts          │ 17
│
│ uptime            │ 4m
│
│ script path       │ /home/ethereum/web3.mainnet/restful.v2.js
│
│ script args       │ N/A
│
│ error log path    │ /home/ethereum/.pm2/logs/restful.v2-error-
0.log │
│ out log path      │ /home/ethereum/.pm2/logs/restful.v2-out-
0.log   │
│ pid path          │ /home/ethereum/.pm2/pids/restful.v2-0.pid
│
│ interpreter       │ node
│
│ interpreter args  │ N/A
│
```

```
││
│ script id          │ 0
│
│ exec cwd           │ /home/ethereum/web3.mainnet
│
│ exec mode          │ fork_mode
│
│ node.js version    │ 9.11.1
│
│ watch & reload     │ ✘
│
│ unstable restarts  │ 0
│
│ created at         │ 2018-04-24T01:37:40.184Z
│
│└───────────────────┴─────────────────────────────────
└──────┘
 Revision control metadata
│┌──────────────────┬──────────────────────────────────
└───────────────────┐
│ revision control │ git
│
│ remote url       │ https://github.com/ibook/web3.example.git
│
│ repository root  │ /home/ethereum/web3.mainnet
│
│ last update      │ 2018-04-24T01:41:31.017Z
│
│ revision         │ f8aaa8769a5e039e0aba44aee5dd2571e5b71032
│
│ comment          │ Merge branch 'master' of
https://github.com/ibook/web3.example │
│ branch           │ master
│
│└─────────────────┴───────────────────────────────────
└───────────────────┘
 Code metrics value
┌──────────────────┬────────┐
│ Loop delay       │ 1.28ms │
│ Active requests  │ 0      │
│ Active handles   │ 4      │
└──────────────────┴────────┘
 Add your own code metrics: http://bit.ly/code-metrics
 Use `pm2 logs restful.v2 [--lines 1000]` to display logs
 Use `pm2 monit` to monitor CPU and Memory usage restful.v2
```

系统资源监控

```
[ethereum@netkiller web3.mainnet]$ pm2 monit

┌─ Process list ─────────────┐┌─ Global Logs
────────────────────────────────────────────────────┐
│[ 0] restful.v2     Mem:   ││
│                           ││
│                           ││
│                           ││
│                           ││
│                           ││
│                           ││
│                           ││
│                           ││
│                           ││
│                           ││
│                           ││
│                           ││
│                           ││
└───────────────────────────┘└──────────────────────────────────
───────────────────────────────┘
┌─ Custom metrics (http://b─┐┌─ Metadata
───────────────────────────────────────────────────────┐
│ Loop delay         1.05ms ││ App Name              restful.v2
```

```
││
│ Active requests        0  ││ Restarts              17
│
│ Active handles         4  ││ Uptime                6m
│
└────────────────────────────┘ └──────────────────────────────────
────────────────────────────────┘
 left/right: switch boards | up/down/mouse: scroll | Ctrl-C:
exit        To go further check out
```

监控接口

pm2可以提供一个接口获取监控信息

```
[ethereum@netkiller web3.mainnet]$ pm2 web
Launching web interface on 0.0.0.0:9615
[PM2][WARN] Applications pm2-http-interface not running,
starting...
[PM2] App [pm2-http-interface] launched (1 instances)
[PM2] Process launched
┌─────────────────────┬────┬──────┬───────┬────────┬─────────┬────
─────┬─────┬──────────┬──────────┬──────────┐
│ App name            │ id │ mode │ pid   │ status │ restart │
uptime │ cpu │ mem       │ user      │ watching │
├─────────────────────┼────┼──────┼───────┼────────┼─────────┼────
─────┼─────┼──────────┼──────────┼──────────┤
│ pm2-http-interface │ 1  │ fork │ 11809 │ online │ 0       │ 0s
  0%    │ 2.7 MB    │ ethereum │ disabled │
│ restful.v2          │ 0  │ fork │ 11480 │ online │ 17      │ 8m
  0%    │ 26.6 MB   │ ethereum │ disabled │
└─────────────────────┴────┴──────┴───────┴────────┴─────────┴────
─────┴─────┴──────────┴──────────┴──────────┘
 Use `pm2 show <id|name>` to get more details about an app
```

通过 http://localhost:9615 取得监控数据

```
[ethereum@netkiller web3.mainnet]$ curl http://localhost:9615
{"system_info":{"hostname":"netkiller","uptime":842027},"monit":
{"loadavg":
[2.017578125,2.232421875,2.29150390625],"total_mem":8202674176,"
free_mem":143159296,"cpu":[{"model":"Intel(R) Xeon(R) Platinum
8163 CPU @ 2.50GHz","speed":2494,"times":
{"user":4894565000,"nice":7400,"sys":438689200,"idle":1763658900
,"irq":0}},{"model":"Intel(R) Xeon(R) Platinum 8163 CPU @
2.50GHz","speed":2494,"times":
{"user":4212099800,"nice":10400,"sys":300205500,"idle":319757380
0,"irq":0}},{"model":"Intel(R) Xeon(R) Platinum 8163 CPU @
2.50GHz","speed":2494,"times":
{"user":4412640700,"nice":7000,"sys":334607800,"idle":2757973000
,"irq":0}},{"model":"Intel(R) Xeon(R) Platinum 8163 CPU @
2.50GHz","speed":2494,"times":
{"user":4134519800,"nice":8700,"sys":297732100,"idle":3231274500
,"irq":0}}],"interfaces":{"lo":
[{"address":"127.0.0.1","netmask":"255.0.0.0","family":"IPv4","m
ac":"00:00:00:00:00:00","internal":true,"cidr":"127.0.0.1/8"}],"
eth0":
[{"address":"172.31.180.30","netmask":"255.255.240.0","family":"
IPv4","mac":"00:16:3e:02:8b:75","internal":false,"cidr":"172.31.
180.30/20"}]}},"processes":
[{"pid":11480,"name":"restful.v2","pm2_env":
{"username":"ethereum","windowsHide":true,"instance_var":"NODE_A
PP_INSTANCE","exec_mode":"fork_mode","watch":false,"treekill":tr
ue,"autorestart":true,"automation":true,"pmx":true,"vizion":true
,"name":"restful.v2","node_args":
[],"pm_exec_path":"/home/ethereum/web3.mainnet/restful.v2.js","e
nv":
{"PM2_INTERACTOR_PROCESSING":"true","PM2_USAGE":"CLI","_":"/srv/
node/bin/pm2","HISTTIMEFORMAT":"%Y-%m-%d %H:%M:%S
","LESSOPEN":"||/usr/bin/lesspipe.sh
%s","LOGNAME":"ethereum","GOROOT":"/srv/go","HOME":"/home/ethere
um","SHLVL":"1","HISTCONTROL":"ignoredups","LANG":"zh_CN.UTF-
8","PWD":"/home/ethereum/web3.mainnet","PATH":"/usr/local/bin:/b
in:/usr/bin:/usr/local/sbin:/usr/sbin:/srv/go-
ethereum/bin:/srv/go/bin:/srv/go-
ipfs:/srv/node/bin::/home/ethereum/.local/bin:/home/ethereum/bin
","MAIL":"/var/spool/mail/ethereum","LS_COLORS":"rs=0:di=01;34:l
n=01;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=40;33;01:cd=40;33;01
:or=40;31;01:mi=01;05;37;41:su=37;41:sg=30;43:ca=30;41:tw=30;42:
ow=34;42:st=37;44:ex=01;32:*.tar=01;31:*.tgz=01;31:*.arc=01;31:*
.arj=01;31:*.taz=01;31:*.lha=01;31:*.lz4=01;31:*.lzh=01;31:*.lzm
a=01;31:*.tlz=01;31:*.txz=01;31:*.tzo=01;31:*.t7z=01;31:*.zip=01
;31:*.z=01;31:*.Z=01;31:*.dz=01;31:*.gz=01;31:*.lrz=01;31:*.lz=0
1;31:*.lzo=01;31:*.xz=01;31:*.bz2=01;31:*.bz=01;31:*.tbz=01;31:*
.tbz2=01;31:*.tz=01;31:*.deb=01;31:*.rpm=01;31:*.jar=01;31:*.war
=01;31:*.ear=01;31:*.sar=01;31:*.rar=01;31:*.alz=01;31:*.ace=01;
31:*.zoo=01;31:*.cpio=01;31:*.7z=01;31:*.rz=01;31:*.cab=01;31:*.
jpg=01;35:*.jpeg=01;35:*.gif=01;35:*.bmp=01;35:*.pbm=01;35:*.pgm
=01;35:*.ppm=01;35:*.tga=01;35:*.xbm=01;35:*.xpm=01;35:*.tif=01;
35:*.tiff=01;35:*.png=01;35:*.svg=01;35:*.svgz=01;35:*.mng=01;35
:*.pcx=01;35:*.mov=01;35:*.mpg=01;35:*.mpeg=01;35:*.m2v=01;35:*.
mkv=01;35:*.webm=01;35:*.ogm=01;35:*.mp4=01;35:*.m4v=01;35:*.mp4
v=01;35:*.vob=01;35:*.qt=01;35:*.nuv=01;35:*.wmv=01;35:*.asf=01;
35:*.rm=01;35:*.rmvb=01;35:*.flc=01;35:*.avi=01;35:*.fli=01;35:*
.flv=01;35:*.gl=01;35:*.dl=01;35:*.xcf=01;35:*.xwd=01;35:*.yuv=0
1;35:*.cgm=01;35:*.emf=01;35:*.axv=01;35:*.anx=01;35:*.ogv=01;35
:*.ogx=01;35:*.aac=01;36:*.au=01;36:*.flac=01;36:*.mid=01;36:*.m
idi=01;36:*.mka=01;36:*.mp3=01;36:*.mpc=01;36:*.ogg=01;36:*.ra=0
1;36:*.wav=01;36:*.axa=01;36:*.oga=01;36:*.spx=01;36:*.xspf=01;3
6:","USER":"ethereum","TIME_STYLE":"long-
iso","HISTSIZE":"1000","TERM":"xterm","SHELL":"/bin/bash","HOSTN
AME":"netkiller","XDG_SESSION_ID":"392","PM2_HOME":"/home/ethere
um/.pm2","restful.v2":
{}},"pm_cwd":"/home/ethereum/web3.mainnet","exec_interpreter":"n
ode","instances":1,"pm_out_log_path":"/home/ethereum/.pm2/logs/r
estful.v2-out-
0.log","pm_err_log_path":"/home/ethereum/.pm2/logs/restful.v2-
error-0.log","pm_pid_path":"/home/ethereum/.pm2/pids/restful.v2-
0.pid","km_link":false,"vizion_running":false,"NODE_APP_INSTANCE
":0,"PM2_INTERACTOR_PROCESSING":"true","PM2_USAGE":"CLI","_":"/s
rv/node/bin/pm2","HISTTIMEFORMAT":"%Y-%m-%d %H:%M:%S
","LESSOPEN":"||/usr/bin/lesspipe.sh
%s","LOGNAME":"ethereum","GOROOT":"/srv/go","HOME":"/home/ethere
um","SHLVL":"1","HISTCONTROL":"ignoredups","LANG":"zh_CN.UTF-
8","PWD":"/home/ethereum/web3.mainnet","PATH":"/usr/local/bin:/b
in:/usr/bin:/usr/local/sbin:/usr/sbin:/srv/go-
ethereum/bin:/srv/go/bin:/srv/go-
ipfs:/srv/node/bin::/home/ethereum/.local/bin:/home/ethereum/bin
","MAIL":"/var/spool/mail/ethereum","LS_COLORS":"rs=0:di=01;34:l
n=01;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=40;33;01:cd=40;33;01
:or=40;31;01:mi=01;05;37;41:su=37;41:sg=30;43:ca=30;41:tw=30;42:
ow=34;42:st=37;44:ex=01;32:*.tar=01;31:*.tgz=01;31:*.arc=01;31:*
.arj=01;31:*.taz=01;31:*.lha=01;31:*.lz4=01;31:*.lzh=01;31:*.lzm
a=01;31:*.tlz=01;31:*.txz=01;31:*.tzo=01;31:*.t7z=01;31:*.zip=01
;31:*.z=01;31:*.Z=01;31:*.dz=01;31:*.gz=01;31:*.lrz=01;31:*.lz=0
1;31:*.lzo=01;31:*.xz=01;31:*.bz2=01;31:*.bz=01;31:*.tbz=01;31:*
.tbz2=01;31:*.tz=01;31:*.deb=01;31:*.rpm=01;31:*.jar=01;31:*.war
=01;31:*.ear=01;31:*.sar=01;31:*.rar=01;31:*.alz=01;31:*.ace=01;
31:*.zoo=01;31:*.cpio=01;31:*.7z=01;31:*.rz=01;31:*.cab=01;31:*.
```

```
jpg=01;35:*.jpeg=01;35:*.gif=01;35:*.bmp=01;35:*.pbm=01;35:*.pgm
=01;35:*.ppm=01;35:*.tga=01;35:*.xbm=01;35:*.xpm=01;35:*.tif=01;
35:*.tiff=01;35:*.png=01;35:*.svg=01;35:*.svgz=01;35:*.mng=01;35
:*.pcx=01;35:*.mov=01;35:*.mpg=01;35:*.mpeg=01;35:*.m2v=01;35:*.
mkv=01;35:*.webm=01;35:*.ogm=01;35:*.mp4=01;35:*.m4v=01;35:*.mp4
v=01;35:*.vob=01;35:*.qt=01;35:*.nuv=01;35:*.wmv=01;35:*.asf=01;
35:*.rm=01;35:*.rmvb=01;35:*.flc=01;35:*.avi=01;35:*.fli=01;35:*
.flv=01;35:*.gl=01;35:*.dl=01;35:*.xcf=01;35:*.xwd=01;35:*.yuv=0
1;35:*.cgm=01;35:*.emf=01;35:*.axv=01;35:*.anx=01;35:*.ogv=01;35
:*.ogx=01;35:*.aac=01;36:*.au=01;36:*.flac=01;36:*.mid=01;36:*.m
idi=01;36:*.mka=01;36:*.mp3=01;36:*.mpc=01;36:*.ogg=01;36:*.ra=0
1;36:*.wav=01;36:*.axa=01;36:*.oga=01;36:*.spx=01;36:*.xspf=01;3
6:","USER":"ethereum","TIME_STYLE":"long-
iso","HISTSIZE":"1000","TERM":"xterm","SHELL":"/bin/bash","HOSTN
AME":"netkiller","XDG_SESSION_ID":"392","PM2_HOME":"/home/ethere
um/.pm2","status":"online","pm_uptime":1524533860184,"axm_action
s":[],"axm_monitor":{"Loop delay":{"alert":
{},"unit":"ms","type":"libuv/latency","historic":true,"agg_type"
:"avg","value":"0.91ms"},"Active requests":{"alert":
{},"type":"Active
requests","historic":true,"agg_type":"avg","value":0},"Active
handles":{"alert":{},"type":"Active
handles","historic":true,"agg_type":"avg","value":4}},"axm_optio
ns":
{"default_actions":true,"transactions":false,"http":false,"http_
latency":200,"http_code":500,"ignore_routes":
[],"profiling":true,"errors":true,"alert_enabled":true,"custom_p
robes":true,"network":false,"ports":false,"ignoreFilter":
{"method":["OPTIONS"],"url":[]},"excludedHooks":
[],"deep_metrics":false,"event_loop_dump":false,"v8":false,"modu
le_conf":
{},"module_name":"restful.v2","module_version":"2.10.2","pmx_ver
sion":"1.6.4","error":true},"axm_dynamic":
{},"created_at":1524533860184,"pm_id":0,"restart_time":17,"unsta
ble_restarts":0,"node_version":"9.11.1","versioning":
{"type":"git","url":"https://github.com/ibook/web3.example.git",
"revision":"f8aaa8769a5e039e0aba44aee5dd2571e5b71032","update_ti
me":"2018-04-24T01:46:31.016Z","comment":"Merge branch 'master'
of
https://github.com/ibook/web3.example","unstaged":true,"branch":
"master","remotes":
["origin"],"remote":"origin","branch_exists_on_remote":true,"ahe
ad":false,"next_rev":null,"prev_rev":"a42ab811b17763684ad25b24ad
cf10836f39f2c3","repo_path":"/home/ethereum/web3.mainnet"},"exit
_code":1},"pm_id":0,"monit":{"memory":27947008,"cpu":0}},
{"pid":11809,"name":"pm2-http-interface","pm2_env":
```

```
{"username":"ethereum","windowsHide":true,"instance_var":"NODE_A
PP_INSTANCE","exec_mode":"fork_mode","env":{"pm2-http-
interface":"
{}","PM2_HOME":"/home/ethereum/.pm2","XDG_SESSION_ID":"394","HOS
TNAME":"netkiller","SHELL":"/bin/bash","TERM":"xterm-
256color","HISTSIZE":"1000","TIME_STYLE":"long-
iso","USER":"ethereum","LS_COLORS":"rs=0:di=38;5;27:ln=38;5;51:m
h=44;38;5;15:pi=40;38;5;11:so=38;5;13:do=38;5;5:bd=48;5;232;38;5
;11:cd=48;5;232;38;5;3:or=48;5;232;38;5;9:mi=05;48;5;232;38;5;15
:su=48;5;196;38;5;15:sg=48;5;11;38;5;16:ca=48;5;196;38;5;226:tw=
48;5;10;38;5;16:ow=48;5;10;38;5;21:st=48;5;21;38;5;15:ex=38;5;34
:*.tar=38;5;9:*.tgz=38;5;9:*.arc=38;5;9:*.arj=38;5;9:*.taz=38;5;
9:*.lha=38;5;9:*.lz4=38;5;9:*.lzh=38;5;9:*.lzma=38;5;9:*.tlz=38;
5;9:*.txz=38;5;9:*.tzo=38;5;9:*.t7z=38;5;9:*.zip=38;5;9:*.z=38;5
;9:*.Z=38;5;9:*.dz=38;5;9:*.gz=38;5;9:*.lrz=38;5;9:*.lz=38;5;9:*
.lzo=38;5;9:*.xz=38;5;9:*.bz2=38;5;9:*.bz=38;5;9:*.tbz=38;5;9:*.
tbz2=38;5;9:*.tz=38;5;9:*.deb=38;5;9:*.rpm=38;5;9:*.jar=38;5;9:*
.war=38;5;9:*.ear=38;5;9:*.sar=38;5;9:*.rar=38;5;9:*.alz=38;5;9:
*.ace=38;5;9:*.zoo=38;5;9:*.cpio=38;5;9:*.7z=38;5;9:*.rz=38;5;9:
*.cab=38;5;9:*.jpg=38;5;13:*.jpeg=38;5;13:*.gif=38;5;13:*.bmp=38
;5;13:*.pbm=38;5;13:*.pgm=38;5;13:*.ppm=38;5;13:*.tga=38;5;13:*.
xbm=38;5;13:*.xpm=38;5;13:*.tif=38;5;13:*.tiff=38;5;13:*.png=38;
5;13:*.svg=38;5;13:*.svgz=38;5;13:*.mng=38;5;13:*.pcx=38;5;13:*.
mov=38;5;13:*.mpg=38;5;13:*.mpeg=38;5;13:*.m2v=38;5;13:*.mkv=38;
5;13:*.webm=38;5;13:*.ogm=38;5;13:*.mp4=38;5;13:*.m4v=38;5;13:*.
mp4v=38;5;13:*.vob=38;5;13:*.qt=38;5;13:*.nuv=38;5;13:*.wmv=38;5
;13:*.asf=38;5;13:*.rm=38;5;13:*.rmvb=38;5;13:*.flc=38;5;13:*.av
i=38;5;13:*.fli=38;5;13:*.flv=38;5;13:*.gl=38;5;13:*.dl=38;5;13:
*.xcf=38;5;13:*.xwd=38;5;13:*.yuv=38;5;13:*.cgm=38;5;13:*.emf=38
;5;13:*.axv=38;5;13:*.anx=38;5;13:*.ogv=38;5;13:*.ogx=38;5;13:*.
aac=38;5;45:*.au=38;5;45:*.flac=38;5;45:*.mid=38;5;45:*.midi=38;
5;45:*.mka=38;5;45:*.mp3=38;5;45:*.mpc=38;5;45:*.ogg=38;5;45:*.r
a=38;5;45:*.wav=38;5;45:*.axa=38;5;45:*.oga=38;5;45:*.spx=38;5;4
5:*.xspf=38;5;45:","MAIL":"/var/spool/mail/ethereum","PATH":"/us
r/local/bin:/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/srv/go-
ethereum/bin:/srv/go/bin:/srv/go-
ipfs:/srv/node/bin::/home/ethereum/.local/bin:/home/ethereum/bin
","PWD":"/home/ethereum/web3.mainnet","LANG":"zh_CN.UTF-
8","HISTCONTROL":"ignoredups","SHLVL":"1","HOME":"/home/ethereum
","GOROOT":"/srv/go","LOGNAME":"ethereum","LESSOPEN":"||/usr/bin
/lesspipe.sh %s","HISTTIMEFORMAT":"%Y-%m-%d %H:%M:%S
","_":"/srv/node/bin/pm2","PM2_USAGE":"CLI","PM2_JSON_PROCESSING
":"true"},"treekill":true,"autorestart":true,"automation":true,"
pmx":true,"vizion":true,"name":"pm2-http-interface","node_args":
[],"pm_exec_path":"/srv/node-
v9.11.1/lib/node_modules/pm2/lib/HttpInterface.js","pm_cwd":"/ho
```

```
me/ethereum/web3.mainnet","exec_interpreter":"node","instances":
1,"pm_out_log_path":"/home/ethereum/.pm2/logs/pm2-http-
interface-out-
1.log","pm_err_log_path":"/home/ethereum/.pm2/logs/pm2-http-
interface-error-
1.log","pm_pid_path":"/home/ethereum/.pm2/pids/pm2-http-
interface-
1.pid","km_link":false,"vizion_running":false,"NODE_APP_INSTANCE
":0,"pm2-http-interface":"
{}","PM2_HOME":"/home/ethereum/.pm2","XDG_SESSION_ID":"394","HOS
TNAME":"netkiller","SHELL":"/bin/bash","TERM":"xterm-
256color","HISTSIZE":"1000","TIME_STYLE":"long-
iso","USER":"ethereum","LS_COLORS":"rs=0:di=38;5;27:ln=38;5;51:m
h=44;38;5;15:pi=40;38;5;11:so=38;5;13:do=38;5;5:bd=48;5;232;38;5
;11:cd=48;5;232;38;5;3:or=48;5;232;38;5;9:mi=05;48;5;232;38;5;15
:su=48;5;196;38;5;15:sg=48;5;11;38;5;16:ca=48;5;196;38;5;226:tw=
48;5;10;38;5;16:ow=48;5;10;38;5;21:st=48;5;21;38;5;15:ex=38;5;34
:*.tar=38;5;9:*.tgz=38;5;9:*.arc=38;5;9:*.arj=38;5;9:*.taz=38;5;
9:*.lha=38;5;9:*.lz4=38;5;9:*.lzh=38;5;9:*.lzma=38;5;9:*.tlz=38;
5;9:*.txz=38;5;9:*.tzo=38;5;9:*.t7z=38;5;9:*.zip=38;5;9:*.z=38;5
;9:*.Z=38;5;9:*.dz=38;5;9:*.gz=38;5;9:*.lrz=38;5;9:*.lz=38;5;9:*
.lzo=38;5;9:*.xz=38;5;9:*.bz2=38;5;9:*.bz=38;5;9:*.tbz=38;5;9:*.
tbz2=38;5;9:*.tz=38;5;9:*.deb=38;5;9:*.rpm=38;5;9:*.jar=38;5;9:*
.war=38;5;9:*.ear=38;5;9:*.sar=38;5;9:*.rar=38;5;9:*.alz=38;5;9:
*.ace=38;5;9:*.zoo=38;5;9:*.cpio=38;5;9:*.7z=38;5;9:*.rz=38;5;9:
*.cab=38;5;9:*.jpg=38;5;13:*.jpeg=38;5;13:*.gif=38;5;13:*.bmp=38
;5;13:*.pbm=38;5;13:*.pgm=38;5;13:*.ppm=38;5;13:*.tga=38;5;13:*.
xbm=38;5;13:*.xpm=38;5;13:*.tif=38;5;13:*.tiff=38;5;13:*.png=38;
5;13:*.svg=38;5;13:*.svgz=38;5;13:*.mng=38;5;13:*.pcx=38;5;13:*.
mov=38;5;13:*.mpg=38;5;13:*.mpeg=38;5;13:*.m2v=38;5;13:*.mkv=38;
5;13:*.webm=38;5;13:*.ogm=38;5;13:*.mp4=38;5;13:*.m4v=38;5;13:*.
mp4v=38;5;13:*.vob=38;5;13:*.qt=38;5;13:*.nuv=38;5;13:*.wmv=38;5
;13:*.asf=38;5;13:*.rm=38;5;13:*.rmvb=38;5;13:*.flc=38;5;13:*.av
i=38;5;13:*.fli=38;5;13:*.flv=38;5;13:*.gl=38;5;13:*.dl=38;5;13:
*.xcf=38;5;13:*.xwd=38;5;13:*.yuv=38;5;13:*.cgm=38;5;13:*.emf=38
;5;13:*.axv=38;5;13:*.anx=38;5;13:*.ogv=38;5;13:*.ogx=38;5;13:*.
aac=38;5;45:*.au=38;5;45:*.flac=38;5;45:*.mid=38;5;45:*.midi=38;
5;45:*.mka=38;5;45:*.mp3=38;5;45:*.mpc=38;5;45:*.ogg=38;5;45:*.r
a=38;5;45:*.wav=38;5;45:*.axa=38;5;45:*.oga=38;5;45:*.spx=38;5;4
5:*.xspf=38;5;45:","MAIL":"/var/spool/mail/ethereum","PATH":"/us
r/local/bin:/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/srv/go-
ethereum/bin:/srv/go/bin:/srv/go-
ipfs:/srv/node/bin::/home/ethereum/.local/bin:/home/ethereum/bin
","PWD":"/home/ethereum/web3.mainnet","LANG":"zh_CN.UTF-
8","HISTCONTROL":"ignoredups","SHLVL":"1","HOME":"/home/ethereum
","GOROOT":"/srv/go","LOGNAME":"ethereum","LESSOPEN":"||/usr/bin
```

```
/lesspipe.sh %s","HISTTIMEFORMAT":"%Y-%m-%d %H:%M:%S
","_":"/srv/node/bin/pm2","PM2_USAGE":"CLI","PM2_JSON_PROCESSING
":"true","status":"online","pm_uptime":1524534352075,"axm_action
s":[],"axm_monitor":{"Loop delay":{"alert":
{},"unit":"ms","type":"libuv/latency","historic":true,"agg_type"
:"avg","value":"0.78ms"},"Active requests":{"alert":
{},"type":"Active
requests","historic":true,"agg_type":"avg","value":0},"Active
handles":{"alert":{},"type":"Active
handles","historic":true,"agg_type":"avg","value":4}},"axm_optio
ns":
{"default_actions":true,"transactions":false,"http":false,"http_
latency":200,"http_code":500,"ignore_routes":
[],"profiling":true,"errors":true,"alert_enabled":true,"custom_p
robes":true,"network":false,"ports":false,"ignoreFilter":
{"method":["OPTIONS"],"url":[]},"excludedHooks":
[],"deep_metrics":false,"event_loop_dump":false,"v8":false,"modu
le_conf":{},"module_name":"pm2-http-
interface","module_version":"2.10.2","pmx_version":"1.6.4","erro
r":true},"axm_dynamic":
{},"created_at":1524534352075,"pm_id":1,"restart_time":0,"unstab
le_restarts":0,"versioning":null,"node_version":"9.11.1"},"pm_id
":1,"monit":{"memory":24776704,"cpu":0}}]}
```

logs

```
$ pm2 logs project --lines 100
```

设置开机启动

```
[root@localhost ~]# pm2 startup
[PM2] Init System found: systemd
Platform systemd
Template
[Unit]
```

```
Description=PM2 process manager
Documentation=https://pm2.keymetrics.io/
After=network.target

[Service]
Type=forking
User=root
LimitNOFILE=infinity
LimitNPROC=infinity
LimitCORE=infinity
Environment=PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/b
in:/root/bin:/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/
bin
Environment=PM2_HOME=/root/.pm2
PIDFile=/root/.pm2/pm2.pid
Restart=on-failure

ExecStart=/usr/local/lib/node_modules/pm2/bin/pm2 resurrect
ExecReload=/usr/local/lib/node_modules/pm2/bin/pm2 reload all
ExecStop=/usr/local/lib/node_modules/pm2/bin/pm2 kill

[Install]
WantedBy=multi-user.target

Target path
/etc/systemd/system/pm2-root.service
Command list
[ 'systemctl enable pm2-root' ]
[PM2] Writing init configuration in /etc/systemd/system/pm2-
root.service
[PM2] Making script booting at startup...
[PM2] [-] Executing: systemctl enable pm2-root...
Created symlink /etc/systemd/system/multi-user.target.wants/pm2-
root.service → /etc/systemd/system/pm2-root.service.
[PM2] [v] Command successfully executed.
+---------------------------------------+
[PM2] Freeze a process list on reboot via:
$ pm2 save

[PM2] Remove init script via:
$ pm2 unstartup systemd
```

查看状态

```
[root@localhost ~]# systemctl status pm2-root
● pm2-root.service - PM2 process manager
   Loaded: loaded (/etc/systemd/system/pm2-root.service;
enabled; vendor preset: disabled)
   Active: inactive (dead)
     Docs: https://pm2.keymetrics.io/
```

有时我们不希望使用 root 启动程序，可以这样操作

```
[root@localhost ~]# su - www
Last login: Thu Jul 29 14:29:02 CST 2021 on pts/4
[www@localhost ~]$ pm2 startup
[PM2] Init System found: systemd
[PM2] To setup the Startup Script, copy/paste the following
command:
sudo env PATH=$PATH:/usr/bin
/usr/local/lib/node_modules/pm2/bin/pm2 startup systemd -u www -
-hp /home/www
[www@localhost ~]$ sudo env PATH=$PATH:/usr/bin
/usr/local/lib/node_modules/pm2/bin/pm2 startup systemd -u www -
-hp /home/www
[PM2] Init System found: systemd
Platform systemd
Template
[Unit]
Description=PM2 process manager
Documentation=https://pm2.keymetrics.io/
After=network.target

[Service]
Type=forking
User=www
LimitNOFILE=infinity
LimitNPROC=infinity
LimitCORE=infinity
Environment=PATH=/home/www/.local/bin:/home/www/bin:/usr/local/b
in:/usr/bin:/usr/local/sbin:/usr/sbin:/usr/bin:/bin:/usr/local/s
bin:/usr/local/bin:/usr/sbin:/usr/bin
Environment=PM2_HOME=/home/www/.pm2
```

```
PIDFile=/home/www/.pm2/pm2.pid
Restart=on-failure

ExecStart=/usr/local/lib/node_modules/pm2/bin/pm2 resurrect
ExecReload=/usr/local/lib/node_modules/pm2/bin/pm2 reload all
ExecStop=/usr/local/lib/node_modules/pm2/bin/pm2 kill

[Install]
WantedBy=multi-user.target

Target path
/etc/systemd/system/pm2-www.service
Command list
[ 'systemctl enable pm2-www' ]
[PM2] Writing init configuration in /etc/systemd/system/pm2-
www.service
[PM2] Making script booting at startup...
[PM2] [-] Executing: systemctl enable pm2-www...
Created symlink /etc/systemd/system/multi-user.target.wants/pm2-
www.service → /etc/systemd/system/pm2-www.service.
[PM2] [v] Command successfully executed.
+---------------------------------------+
[PM2] Freeze a process list on reboot via:
$ pm2 save

[PM2] Remove init script via:
$ pm2 unstartup systemd
```

```
[www@localhost ~]$ sudo systemctl enable pm2-www

[www@localhost ~]$ pm2 save
[PM2] Saving current process list...
[PM2] Successfully saved in /home/www/.pm2/dump.pm2
```

8. Type

String

```
var a = "I want apple";
var b = "an";
var position = 6;
var output = [a.slice(0, position), b,
a.slice(position)].join('');
console.log(output);

var n = "123456780";
var dot = ".";
var decimals = 4 * -1;
var output = [n.slice(0, decimals), dot,
n.slice(decimals)].join('');
console.log(Number(output));
```

Number

```
var number = Number("1000000");
console.log(number);
```

BigNumber

http://mikemcl.github.io/bignumber.js/

```
const BigNumber = require('bignumber.js');
```

```
// 93774367102000000000 - 378000000000000 =
93773989102000010000

x = new BigNumber(93774367102000000000);
y = new BigNumber(378000000000000);
z = x.minus(y);
z.toString();

n = 3.456
n = new BigNumber(x)
n.toFixed(5)
n.toFixed(10)
```

9. Loop

for json map

```
const contracts = {
    'EOS':'0x86fa049857e0209aa7d9e616f7eb3b3b78ecfdb0',
    'FBC':'0x0c6df9bb12b32aeec1e117936457ed83ad8a1c70',
    'USDT':''
};

for(let symbol in contracts) {
    console.log(symbol,contracts[symbol]);
}
```

forEach

```
arr.forEach(function (item) {
  someFn(item);
})

elements.forEach(function(element){

});
```

```
var a = ["a", "b", "c"];
a.forEach(function(entry) {
    console.log(entry);
});
```

10. 数据结构

set

```
var set = new Set();
set.add("a").add("b").add("d").add("c");

for (let v of set) {
    console.log(v);
}
```

map

```
var map = new Map();
map.set("a",1).set("b",2).set(999,3);

console.log("--------------------");
for(let [k,v] of map) {
    console.log(k,v);
}
```

第 23 章 Meteor

第 24 章 express

1. 安装

```
neo@MacBook-Pro ~ % mkdir node
neo@MacBook-Pro ~ % cd node
neo@MacBook-Pro ~/node % npm install express --save

neo@MacBook-Pro ~/node % npm list express
/Users/neo/node
└── express@4.16.3
```

```
npm install body-parser --save
npm install cookie-parser --save
npm install multer --save
```

```
echo 'node_modules' > .gitignore
```

Hello world 测试程序

```
neo@MacBook-Pro ~/node % cat demo.js
var express = require('express');
var app = express();

app.get('/', function (req, res) {
   res.send('Hello World');
})
```

```
var server = app.listen(8080, function () {

  var host = server.address().address
  var port = server.address().port

  console.log("应用实例，访问地址为 http://%s:%s", host, port)

})
```

```
neo@MacBook-Pro ~/node % node demo.js
应用实例，访问地址为 http://:::8080
```

浏览器访问 http://localhost:8080 将看到 Hello World 字符串

2. 配置

监听 0.0.0.0 地址

默认系统监听 127.0.0.1

```
var port = process.env.PORT || 8000;
app.listen(port, '0.0.0.0');
```

跨区请求

```
var allowCrossDomain = function(req, res, next) {
    res.header('Access-Control-Allow-Origin', '*');
    res.header('Access-Control-Allow-Methods', 
'GET,PUT,POST,DELETE');
    res.header('Access-Control-Allow-Headers', 'Content-
Type');

    next();
}

app.use(allowCrossDomain);
```

3. 请求和响应

Express 应用使用回调函数的参数： request 和 response 对象来处理请求和响应的数据。

```
app.get('/', function (req, res) {
   // --
})
```

request 和 response 对象的具体介绍：

```
Request 对象 - request 对象表示 HTTP 请求，包含了请求查询字符串，参数，内容，HTTP 头部等属性。常见属性有：

req.app：当callback为外部文件时，用req.app访问express的实例
req.baseUrl：获取路由当前安装的URL路径
req.body / req.cookies：获得「请求主体」/ Cookies
req.fresh / req.stale：判断请求是否还「新鲜」
req.hostname / req.ip：获取主机名和IP地址
req.originalUrl：获取原始请求URL
req.params：获取路由的parameters
req.path：获取请求路径
req.protocol：获取协议类型
req.query：获取URL的查询参数串
req.route：获取当前匹配的路由
req.subdomains：获取子域名
req.accepts()：检查可接受的请求的文档类型
req.acceptsCharsets / req.acceptsEncodings /
req.acceptsLanguages：返回指定字符集的第一个可接受字符编码
req.get()：获取指定的HTTP请求头
req.is()：判断请求头Content-Type的MIME类型


Response 对象 - response 对象表示 HTTP 响应，即在接收到请求时向客户端发送的 HTTP 响应数据。常见属性有：
```

```
res.app：同req.app一样
res.append()：追加指定HTTP头
res.set()在res.append()后将重置之前设置的头
res.cookie(name，value [，option])：设置Cookie
opition: domain / expires / httpOnly / maxAge / path / secure /
signed
res.clearCookie()：清除Cookie
res.download()：传送指定路径的文件
res.get()：返回指定的HTTP头
res.json()：传送JSON响应
res.jsonp()：传送JSONP响应
res.location()：只设置响应的Location HTTP头，不设置状态码或者close
response
res.redirect()：设置响应的Location HTTP头，并且设置状态码302
res.render(view,[locals],callback)：渲染一个view，同时向callback传
递渲染后的字符串，如果在渲染过程中有错误发生next(err)将会被自动调用。
callback将会被传入一个可能发生的错误以及渲染后的页面，这样就不会自动输出
了。
res.send()：传送HTTP响应
res.sendFile(path [，options] [，fn])：传送指定路径的文件 -会自动根据
文件extension设定Content-Type
res.set()：设置HTTP头，传入object可以一次设置多个头
res.status()：设置HTTP状态码
res.type()：设置Content-Type的MIME类型
```

cookie 处理

查看 Cookie

```
// express_cookie.js 文件
var express      = require('express')
var cookieParser = require('cookie-parser')
var util = require('util');

var app = express()
app.use(cookieParser())

app.get('/', function(req, res) {
    console.log("Cookies: " + util.inspect(req.cookies));
})
```

4. 路由

路由决定了由谁(指定函数)去响应客户端请求

路由包含了 GET 和 POST 路由

接下来我们扩展 Hello World，添加一些功能来处理更多类型的 HTTP 请求。

```
//  主页输出 "Hello World"
app.get('/', function (req, res) {
   console.log("主页 GET 请求");
   res.send('Hello GET');
})
```

```
//  POST 请求
app.post('/', function (req, res) {
   console.log("主页 POST 请求");
   res.send('Hello POST');
})
```

```
// 通配符匹配，对页面 ab开头，cd结尾的URL做出 GET 请求响应
app.get('/ab*cd', function(req, res) {
   console.log("/ab*cd GET 请求");
   res.send('正则匹配');
})
```

express.Router()

```
var express    = require('express');
var app        = express();
var router = express.Router();


router.get('/', function(req, res) {
    res.json({ "status": true, message: 'welcome to ethereum
api!' });
});

app.use('/api', router);

var port = process.env.PORT || 8080;
app.listen(port);
```

5. 静态文件

使用 express.static 来设置静态文件路径。

```
app.use(express.static('public'));
```

范例

```
var express = require('express');
var app = express();

app.use(express.static('public'));

app.get('/', function (req, res) {
   res.send('Hello World');
})

var server = app.listen(8080, function () {

  var host = server.address().address
  var port = server.address().port

  console.log("应用实例，访问地址为 http://%s:%s", host, port)

})
```

然后在项目目录创建 public/images 放入 logo.png 文件，通过地址 http://127.0.0.1:8080/images/logo.png 访问

6. Form 表单处理

login.html 文件

```
<html>
<body>
        <form action="/login" method="POST">
        First Name: <input type="text" name="first_name">
<br>
        Last Name: <input type="text" name="last_name">
        <input type="submit" value="Submit">
        </form>
</body>
</html>
```

node 文件

```
var express = require('express');
var app = express();

app.use(express.static('public'));

app.get('/', function (req, res) {
   res.send('Hello World');
})

app.get('/login.html', function (req, res) {
   res.sendFile( __dirname + "/" + "login.html" );
})

app.post('/login', function (req, res) {

   // 输出 JSON 格式
   var response = {
       "first_name":req.query.first_name,
       "last_name":req.query.last_name
   };
   console.log(response);
   res.end(JSON.stringify(response));
})

var server = app.listen(8080, function () {

  var host = server.address().address
  var port = server.address().port

  console.log("应用实例，访问地址为 http://%s:%s", host, port)

})
```

上传文件

index.html 文件

```


<html>
<head>
<title>文件上传表单</title>
</head>
<body>
        <h3>文件上传：</h3>
        请选择一个文件上传：<br />
        <form action="/file_upload" method="post"
enctype="multipart/form-data">
                <input type="file" name="image" size="50" />
                <br />
                <input type="submit" value="上传文件" />
        </form>
</body>
</html>
```

```
var express = require('express');
var app = express();
var fs = require("fs");

var bodyParser = require('body-parser');
var multer  = require('multer');

app.use(express.static('public'));
app.use(bodyParser.urlencoded({ extended: false }));
app.use(multer({ dest: '/tmp/'}).array('image'));

app.get('/index.html', function (req, res) {
   res.sendFile( __dirname + "/" + "index.html" );
})

app.post('/file_upload', function (req, res) {

   console.log(req.files[0]);  // 上传的文件信息

   var des_file = __dirname + "/" +
req.files[0].originalname;
   fs.readFile( req.files[0].path, function (err, data) {
        fs.writeFile(des_file, data, function (err) {
         if( err ){
              console.log( err );
         }else{
               response = {
                   message:'File uploaded successfully',
                   filename:req.files[0].originalname
              };
          }
          console.log( response );
          res.end( JSON.stringify( response ) );
       });
   });
})

var server = app.listen(8080, function () {

  var host = server.address().address
  var port = server.address().port

  console.log("应用实例，访问地址为 http://%s:%s", host, port)
})
```

7. 模板

安装模板

安装模板

```
npm install ejs
```

```
var express = require('express');
var app = express();

app.use(express.static('public'));
app.set("view engine","ejs");
app.set('views', __dirname + '/views');
```

配置模板

闭合标签

ejs 里，默认的闭合标记是 <% ... %> ， 我们也可以定义自己的标签。例如：

```
app.set("view options",{
   "open":"{{",
   "close":"}}"
});
```

传递变量

创建模板

```
neo@MacBook-Pro ~/web3.example % mkdir view
neo@MacBook-Pro ~/web3.example % vim view/account.ejs
```

传递变量

```
app.get('/account.html', function (req, res) {
        var accounts = [ '0xB94054c174995AE2A9E7fcf6c7924635FBa8ECF7',

'0xf56b81a2bcb964D2806071e9Be4289A5559BB0fA',

'0x997e5CA600E19447D0B82aFBf9c7F00De2B39B16',

'0x538b392D57d867A57eE8Eed05737cB08B4691302' ];
        res.render("account",{"accounts":accounts});
})
```

res.render("模板文件",{"key":value});

forEach

```
<h1>Users</h1>
<ul id="accounts">
    <% accounts.forEach(function(account, index){ %>
    <li><%= index %>, <%= account %></li>
    <% }) %>
</ul>
```

部分 IV. C/C++

第 25 章 Build tool

1. make - GNU make utility to maintain groups of programs

Makefile

```
$ sudo apt-get install make
```

使用make命令测试

1.1. autoconf - Generate configuration scripts

autoconf

```
$ sudo apt-get install autoconf
```

automake

```
$ sudo apt-get install automake
```

example

过程 25.1. autoconf and automake step by step

1. create directory

```
% mkdir devel
% cd devel
% mkdir hello
% cd hello
```

create a file

```
vim hello.c

   #include
   int main(int argc, char** argv)
   {
   printf(``Hello, GNU!\n'');
   return 0;
   }
```

2. autoscan

```
neo@debian:~/workspace/devel/hello$ autoscan
neo@debian:~/workspace/devel/hello$ ls
autoscan.log  configure.scan  hello.c
```

3. configure.in

```
cp configure.scan configure.in

neo@debian:~/workspace/devel/hello$ aclocal
neo@debian:~/workspace/devel/hello$ autoconf
neo@debian:~/workspace/devel/hello$ ls
autom4te.cache  autoscan.log  configure  configure.in
configure.scan  hello.c
```

4. Makefile.am

```
neo@debian:~/workspace/devel/hello$ vim Makefile.am
neo@debian:~/workspace/devel/hello$ cat Makefile.am
AUTOMAKE_OPTIONS= foreign
bin_PROGRAMS= hello
```

```
hello_SOURCES= hello.c
neo@debian:~/workspace/devel/hello$
```

```
$ automake --add-missing
configure.in: no proper invocation of AM_INIT_AUTOMAKE was
found.
configure.in: You should verify that configure.in invokes
AM_INIT_AUTOMAKE,
configure.in: that aclocal.m4 is present in the top-level
directory,
configure.in: and that aclocal.m4 was recently regenerated
(using aclocal).
automake: no `Makefile.am' found for any configure output
automake: Did you forget AC_CONFIG_FILES([Makefile]) in
configure.in?
```

2. CMake

http://www.cmake.org/

2.1. helloworld

安装CMake

```
$ sudo yum install gcc gcc-c++
$ sudo yum install make

$ sudo yum install cmake28
$ sudo ln -s /usr/bin/cmake28 /usr/bin/cmake
$ cmake --version
cmake version 2.8.9
```

创建 CMakeLists.txt 文件

```
$ cat CMakeLists.txt
PROJECT(example)
ADD_EXECUTABLE(example main.c)
```

创建 main.c 文件

```
$ cat main.c
#include <stdio.h>
int main() {
   printf("helloworld!\n");
   return 0;
}
```

编译程序

```
$ cmake .
-- Configuring done
-- Generating done
-- Build files have been written to: /home/neo/example

$ make
Scanning dependencies of target example
[100%] Building C object CMakeFiles/example.dir/main.c.o
Linking C executable example
[100%] Built target example

$ ./example
helloworld!
```

2.2. cmake_minimum_required

```
cmake_minimum_required(VERSION 2.8.7)
```

2.3. SET

```
SET(CMAKE_INSTALL_PREFIX /usr/local)
```

改变 CMAKE_INSTALL_PREFIX 变量

```
cmake -DCMAKE_INSTALL_PREFIX=/usr ..
```

2.4. ADD_SUBDIRECTORY

```
ADD_SUBDIRECTORY(src bin)
```

2.5. INCLUDE_DIRECTORIES

```
INCLUDE_DIRECTORIES(/usr/include/xen)
```

相当于 gcc -I/usr/include/xen

2.6. 编译文件

ADD_EXECUTABLE 编译可执行

```
SET(SRC_LIST main.cc
        src/file1.c
        src/file2.c
        )

ADD_EXECUTABLE(hello ${SRC_LIST})
```

ADD_LIBRARY 编译库文件

编译 *.a 文件

```
$ cat  CMakeLists.txt
cmake_minimum_required(VERSION 2.8)
PROJECT(zeromq)
ADD_LIBRARY(zeromq zeromq.c)
INCLUDE_DIRECTORIES(/usr/include/mysql)
TARGET_LINK_LIBRARIES(zeromq zmq)
```

编译共享库 *.so 文件

```
$ cat  CMakeLists.txt
cmake_minimum_required(VERSION 2.8)
PROJECT(zeromq)
ADD_LIBRARY(zeromq SHARED zeromq.c)
INCLUDE_DIRECTORIES(/usr/include/mysql)
```

```
TARGET_LINK_LIBRARIES(zeromq zmq)
```

2.7. EXECUTABLE_OUTPUT_PATH / LIBRARY_OUTPUT_PATH

```
SET(EXECUTABLE_OUTPUT_PATH ${PROJECT_BINARY_DIR}/bin)
SET(LIBRARY_OUTPUT_PATH ${PROJECT_BINARY_DIR}/lib)
```

2.8. TARGET_LINK_LIBRARIES

```
TARGET_LINK_LIBRARIES(hello log4cpp)
TARGET_LINK_LIBRARIES(hello zmq)
```

相当于 gcc -lzmq

2.9. INSTALL

```
INSTALL(PROGRAMS hello DESTINATION bin)

INSTALL(FILES COPYRIGHT README DESTINATION share/doc/hello)

INSTALL(DIRECTORY doc/ DESTINATION share/doc/hello)
```

3. scons - a software construction tool

http://www.scons.org/

创建一个hello.c测试文件

```
#include<stdio.h>

main()
{
    printf("Hello World!\n");
}
```

创建SConstruct文件（相当于Makefile）

```
$ cat SConstruct
Program('hello.c')
```

开始编译

```
$ scons
scons: Reading SConscript files ...
scons: done reading SConscript files.
scons: Building targets ...
gcc -o hello.o -c hello.c
gcc -o hello hello.o
scons: done building targets.
```

编译后产生的文件，尝试运行hello程序

```
$ ls
hello  hello.c  hello.o  SConstruct
```

```
$ ./hello
Hello World!
```

下面操作想当于 make clean

```
$ scons -c
scons: Reading SConscript files ...
scons: done reading SConscript files.
scons: Cleaning targets ...
Removed hello.o
Removed hello
scons: done cleaning targets.

$ ls
hello.c  SConstruct
```

4. Phing

http://www.phing.info/

```
$ pear channel-discover pear.phing.info
$ pear install phing/phing
```

第 26 章 C

1. compiler

1.1. gcc - The GNU C compiler

$ sudo apt-get install gcc

```
$ sudo apt-get install gcc
```

1.2. clang - Low-Level Virtual Machine (LLVM), C language family frontend

```
$ apt-cache search clang
llvm-3.0 - Low-Level Virtual Machine (LLVM)
clang - Low-Level Virtual Machine (LLVM), C language family
frontend
libclang-common-dev - clang library - Common development
package
libclang-dev - clang library - Development package
libclang1 - clang library
libsclang1 - SuperCollider language interpreter library
llvm-2.8 - Low-Level Virtual Machine (LLVM)
llvm-2.9 - Low-Level Virtual Machine (LLVM)
```

```
$ apt-get install clang
```

例 26.1. clang helloworld

```
$ cat hello.c
#include <stdio.h>
```

```
int main(int argc, char **argv) { printf("hello world\n"); }
```

```
$ vim hello.c
$ clang hello.c -o hello
$ ./hello
hello world
```

2. ldconfig

```
[root@localhost src]# ldconfig  -p | grep mysql
        libmysqlclient_r.so.15 (libc6,x86-64) =>
/usr/lib64/mysql/libmysqlclient_r.so.15
        libmysqlclient_r.so.15 (libc6) =>
/usr/lib/mysql/libmysqlclient_r.so.15
        libmysqlclient_r.so (libc6,x86-64) =>
/usr/lib64/mysql/libmysqlclient_r.so
        libmysqlclient_r.so (libc6) =>
/usr/lib/mysql/libmysqlclient_r.so
        libmysqlclient.so.15 (libc6,x86-64) =>
/usr/lib64/mysql/libmysqlclient.so.15
        libmysqlclient.so.15 (libc6) =>
/usr/lib/mysql/libmysqlclient.so.15
        libmysqlclient.so (libc6,x86-64) =>
/usr/lib64/mysql/libmysqlclient.so
        libmysqlclient.so (libc6) =>
/usr/lib/mysql/libmysqlclient.so
```

3. C Library

3.1. lib

syslog.h

```
# cat syslog.c
#include <stdio.h>
#include <unistd.h>
#include <syslog.h>

int main(void) {

 openlog("slog", LOG_PID|LOG_CONS, LOG_USER);
 syslog(LOG_INFO, "A different kind of Hello world ... ");
 closelog();

 return 0;
}
```

```
[root@dev1 test]# gcc syslog.c
[root@dev1 test]# ls
a.out  syslog.c
[root@dev1 test]# ./a.out

[root@dev1 test]# tail /var/log/messages
Jan 11 23:52:27 dev1 slog[5056]: A different kind of Hello world
...
```

stdio.h

fscanf/fprintf

```
#include<stdio.h>
int main()
{
        FILE *file;
        char name[20][20];

        int a[10]={0};

        int i,j;
        if((file=fopen("1.txt","rt"))==NULL)
        {
                printf("Cannot open file strike any key exit!");
                return 0;
        }
        i=0;
        while(fscanf(file,"%s %d\n",name[i],&a[i])!=EOF)
        {
                i++;
        }
        i=0;
        while(a[i]!=0)
        {
                printf("%s %d\n",name[i],a[i]);
                i++;
        }
        fclose(file);
}
```

```
#include <stdio.h>
#include <string.h>

typedef struct _Address
{
        char *name;
        int age;
}Address;

int main()
{
        FILE *file;
        int num;
        char str[256];
        int i;
        char *p;
        Address addr[10]={0};

        if((file=fopen("1.txt","rt"))==NULL)
        {
                printf("Cannot open file strike any key exit!");
                return 0;
        }
        i=0;

        while(fscanf(file,"%s %d\n",str,&num)!=EOF)
        {
                asprintf(&addr[i].name, "%s", str);
                addr[i].age = num;
                i++;
        }
        fclose(file);
        addr[i].name = NULL;

        i=0;
        while(1){
                if(addr[i].name == NULL) break;
                printf("%d: %s %d\n",i,
addr[i].name,addr[i].age);
                i++;
        }
}
```

3.2. libssh2

http://www.libssh2.org/

3.3. libconfig – C/C++ Configuration File Library

http://www.hyperrealm.com/main.php?s=libconfig

libconfig 可用于处理 *.conf 配置文件

3.4. libuv

提供Socket,进程线程处理等等

http://nikhilm.github.io/uvbook/

3.5. newt

https://fedorahosted.org/newt/

http://sourcecodebrowser.com/newt/0.52.10/windows_8c.html

http://gnewt.sourceforge.net/tutorial-4.html#ss4.1

3.6. Spdylay - SPDY C Library

http://spdylay.sourceforge.net/

3.7. libPhenom

http://facebook.github.io/libphenom/

libPhenom is an eventing framework for building high performance and high scalability systems in C

3.8. curl

```


#include <stdio.h>
#include <curl/curl.h>

int main(void)
{
  CURL *curl;
  CURLcode res;

  curl = curl_easy_init();
  if(curl) {
    curl_easy_setopt(curl, CURLOPT_URL, "http://example.com");
    /* example.com is redirected, so we tell libcurl to follow
redirection */
```

```
    curl_easy_setopt(curl, CURLOPT_FOLLOWLOCATION, 1L);

    /* Perform the request, res will get the return code */
    res = curl_easy_perform(curl);
    /* Check for errors */
    if(res != CURLE_OK)
      fprintf(stderr, "curl_easy_perform() failed: %s\n",
              curl_easy_strerror(res));

    /* always cleanup */
    curl_easy_cleanup(curl);
  }
  return 0;
}
```

get content

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <curl/curl.h>

struct string {
  char *ptr;
  size_t len;
};

void init_string(struct string *s) {
  s->len = 0;
  s->ptr = malloc(s->len+1);
  if (s->ptr == NULL) {
    fprintf(stderr, "malloc() failed\n");
    exit(EXIT_FAILURE);
  }
  s->ptr[0] = '\0';
}

size_t writefunc(void *ptr, size_t size, size_t nmemb, struct
string *s)
{
  size_t new_len = s->len + size*nmemb;
  s->ptr = realloc(s->ptr, new_len+1);
  if (s->ptr == NULL) {
    fprintf(stderr, "realloc() failed\n");
    exit(EXIT_FAILURE);
  }
  memcpy(s->ptr+s->len, ptr, size*nmemb);
  s->ptr[new_len] = '\0';
  s->len = new_len;

  return size*nmemb;
}

int main(void)
{
  CURL *curl;
  CURLcode res;

  curl = curl_easy_init();
  if(curl) {
    struct string s;
    init_string(&s);

    curl_easy_setopt(curl, CURLOPT_URL, "curl.haxx.se");
    curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, writefunc);
    curl_easy_setopt(curl, CURLOPT_WRITEDATA, &s);
    res = curl_easy_perform(curl);

    printf("%s\n", s.ptr);
    free(s.ptr);

    /* always cleanup */
    curl_easy_cleanup(curl);
  }
  return 0;
}
```

url encode / decode

encode

```
char *subject= curl_easy_escape(curl, "测试主题", 0);
```

decode

```
char *subject= curl_easy_unescape(curl, "测试主题", 0);
```

3.9. libxml

example

创建xml

```
#include <stdio.h>
#include <libxml/parser.h>
#include <libxml/tree.h>

int main(int argc, char **argv)
{
        xmlDocPtr doc = NULL;
        xmlNodePtr root_node = NULL, node = NULL, node1 = NULL;

        doc = xmlNewDoc(BAD_CAST "1.0"); // create a new xml
document.
        root_node = xmlNewNode(NULL, BAD_CAST "root"); //
create a root node.
        xmlDocSetRootElement(doc, root_node);

        xmlNewChild(root_node, NULL, BAD_CAST "node1", BAD_CAST
"content of node1");
        //xmlNewChild(root_node, NULL, BAD_CAST "node2", NULL);

        node = xmlNewChild(root_node, NULL, BAD_CAST "node3",
BAD_CAST "node3 has attributes");
        xmlNewProp(node, BAD_CAST "attribute", BAD_CAST "yes");

        node = xmlNewNode(NULL, BAD_CAST "node4");
        node1 = xmlNewText(BAD_CAST
                   "other way to create content (which is also
a node)");
        xmlAddChild(node, node1);
        xmlAddChild(root_node, node);

        xmlSaveFormatFileEnc(argc > 1 ? argv[1] : "-", doc,
"UTF-8", 1);

        xmlFreeDoc(doc);

        xmlCleanupParser();

        xmlMemoryDump();
        return(0);
}
```

Creating string with libxml2

```
        xmlChar *s;
        int size;
        xmlDocDumpMemory(doc, &s, &size);
        xmlFree(s);

        printf("xml: %s", (char *)s);
```

第 27 章 C++

1. g++ - The GNU C++ compiler

$ sudo apt-get install g++

```
$ sudo apt-get install g++
```

2. C++ library

2.1. Boost C++ Libraries

www.boost.org

2.2. google-perftools

Fast, multi-threaded malloc() and nifty performance analysis tools

http://code.google.com/p/google-perftools/

2.3. TreeFrog Framework

High-speed C++ MVC Framework for Web Application

第 28 章 Objective-C

```
$ sudo apt-get install gobjc gobjc++

$ sudo apt-get install gnustep-make
```

例 28.1. Objective-C hello world

```


$ cat hello.m
#import <stdio.h>

int main( int argc, const char *argv[] ) {
    printf( "hello world\n" );
    return 0;
}
```

```
$ gcc hello.m

$ ./a.out
hello world
```

第 29 章 调试工具

1. file 命令

file 命令将帮助你确定你所处理的文件类型。

```
[root@localhost ~]# file /bin/ls
/bin/ls: ELF 64-bit LSB shared object, x86-64, version 1
(SYSV), dynamically linked, interpreter /lib64/ld-linux-x86-
64.so.2, for GNU/Linux 3.2.0,
BuildID[sha1]=bed150f5bac8514e69b3b98eb6fd3e75d6b277c8,
stripped, too many notes (256)

neo@MacBook-Pro-Neo ~ % file /bin/ls
/bin/ls: Mach-O 64-bit executable x86_64
```

2. ftop - Tool to show progress of open files and file systems

ftop 是一个显示进程打开文件的工具

```
Tue Jan 20 16:25:50 2015
ftop 1.0
Processes:  47 total, 0 unreadable                                                Press h for
help, o for options
Open Files: 57 regular, 0 dir, 149 chr, 0 blk, 12 pipe, 59 sock, 15 misc

_  PID    #FD  USER      COMMAND
-- 413    10   root      /sbin/udevd -d
|  +- 3    -rw  --      935/935      /dev/.udev/queue.bin
-- 982    7    root      auditd
|  +- 5    --W  --     3.1M/3.1M     /var/log/audit/audit.log
-- 1002   5    root      /sbin/rsyslogd -i /var/run/syslogd.pid -c 5
|  +- out  --W  --     8059/8059     /var/log/messages
|  +- err  --W  --    16443/16443    /var/log/cron
|  +- 3    -r-  --        0/0         /proc/kmsg
|  +- 4    --W  --    14976/14976    /var/log/secure
-- 1106   17   www       nginx: worker process
|  +- err  --W  --   538920/538920   /var/log/nginx/error.log (fd 11 for PID 1106)
```

3. ldd - print shared library dependencies

如何找到可执行程序所依赖的库？这就是 ldd 命令的作用了。对动态链接的二进制文件运行该命令会显示出所有依赖库和它们的路径。

```
$ ldd /bin/ls
        linux-gate.so.1 =>  (0xffffe000)
        librt.so.1 => /lib/tls/i686/cmov/librt.so.1
(0xb7f13000)
        libacl.so.1 => /lib/libacl.so.1 (0xb7f0d000)
        libselinux.so.1 => /lib/libselinux.so.1 (0xb7ef9000)
        libc.so.6 => /lib/tls/i686/cmov/libc.so.6 (0xb7dc4000)
        libpthread.so.0 => /lib/tls/i686/cmov/libpthread.so.0
(0xb7db1000)
        /lib/ld-linux.so.2 (0xb7f22000)
        libattr.so.1 => /lib/libattr.so.1 (0xb7dad000)
        libdl.so.2 => /lib/tls/i686/cmov/libdl.so.2
(0xb7da9000)
        libsepol.so.1 => /lib/libsepol.so.1 (0xb7d6c000)
```

举例

```
                        # ./boinc
                        ./boinc: error while loading shared
libraries: libssl.so.1.0.0: cannot open shared object file: No
such file or directory

                        # ldd ./boinc | grep libssl
                        ./boinc: /lib64/libcurl.so.4: no
version information available (required by ./boinc)
                        libssl.so.1.0.0 => not found
                        libssl3.so => /lib64/libssl3.so
(0x00007f1f46998000)
                        libssl.so.10 => /lib64/libssl.so.10
```

```
(0x00007f1f44ba1000)
```

4. strace - trace system calls and signals

跟踪系统调用和信号。

```
-tt  在每行输出的前面,显示毫秒级别的时间
-T   显示每次系统调用所花费的时间
-v   对于某些相关调用,把完整的环境变量,文件stat结构等打出来.
-f   跟踪目标进程,以及目标进程创建的所有子进程.
-e  控制要跟踪的事件和跟踪行为,比如指定要跟踪的系统调用名称.
     trace=file     跟踪和文件访问相关的调用(参数中有文件名)
     trace=process  和进程管理相关的调用,比如fork/exec/exit_group
      trace=network  和网络通信相关的调用,比如socket/sendto/connect
      trace=signal    信号发送和处理相关,比如kill/sigaction
      trace=desc   和文件描述符相关,比如write/read/select/epoll等
      trace=ipc 进程见通信相关,比如shmget等

-o  把strace的输出单独写到指定的文件
-s  当系统调用的某个参数是字符串时,最多输出指定长度的内容, 默认是32个字节.
-p  指定要跟踪的进程pid,要同时跟踪多个pid,重复多次-p选项即可.
```

```
strace -tt -T -v -f -e trace=file -o ~/strace.log.2 -s 1024 -p
25849
```

strace -v ps -e 2

strace -v ls

```
neo@netkiller:~/workspace/Document$ strace -c ls
Docbook  makedoc  Tex
% time     seconds  usecs/call     calls    errors syscall
------ ----------- ----------- --------- --------- -------------
---
```

```
  -nan    0.000000           0        11           read
  -nan    0.000000           0         1           write
  -nan    0.000000           0        38        13 open
  -nan    0.000000           0        27           close
  -nan    0.000000           0        25           fstat
  -nan    0.000000           0        39           mmap
  -nan    0.000000           0        16           mprotect
  -nan    0.000000           0         4           munmap
  -nan    0.000000           0         3           brk
  -nan    0.000000           0         2           rt_sigaction
  -nan    0.000000           0         1
rt_sigprocmask
  -nan    0.000000           0         2           ioctl
  -nan    0.000000           0         9         9 access
  -nan    0.000000           0         1           execve
  -nan    0.000000           0         1           fcntl
  -nan    0.000000           0         2           getdents
  -nan    0.000000           0         1           getrlimit
  -nan    0.000000           0         1           statfs
  -nan    0.000000           0         1           arch_prctl
  -nan    0.000000           0         3         1 futex
  -nan    0.000000           0         1
set_tid_address
  -nan    0.000000           0         1
set_robust_list
------ ----------- ----------- --------- --------- -------------
---
100.00    0.000000                   190        23 total
```

```


neo@netkiller:~/workspace/Document$ strace -f -e open ls
>/dev/null
open("/etc/ld.so.cache", O_RDONLY)      = 3
open("/lib/librt.so.1", O_RDONLY)       = 3
open("/lib/libselinux.so.1", O_RDONLY)  = 3
open("/lib/libacl.so.1", O_RDONLY)      = 3
open("/lib/libc.so.6", O_RDONLY)        = 3
open("/lib/libpthread.so.0", O_RDONLY)  = 3
open("/lib/libdl.so.2", O_RDONLY)       = 3
open("/lib/libattr.so.1", O_RDONLY)     = 3
open("/proc/filesystems", O_RDONLY)     = 3
open("/usr/lib/locale/locale-archive", O_RDONLY) = -1 ENOENT (No
such file or directory)
```

```
open("/usr/share/locale/locale.alias", O_RDONLY) = 3
open("/usr/lib/locale/en_US.UTF-8/LC_IDENTIFICATION", O_RDONLY)
= -1 ENOENT (No such file or directory)
open("/usr/lib/locale/en_US.utf8/LC_IDENTIFICATION", O_RDONLY) =
3
open("/usr/lib/gconv/gconv-modules.cache", O_RDONLY) = 3
open("/usr/lib/locale/en_US.UTF-8/LC_MEASUREMENT", O_RDONLY) =
-1 ENOENT (No such file or directory)
open("/usr/lib/locale/en_US.utf8/LC_MEASUREMENT", O_RDONLY) = 3
open("/usr/lib/locale/en_US.UTF-8/LC_TELEPHONE", O_RDONLY) = -1
ENOENT (No such file or directory)
open("/usr/lib/locale/en_US.utf8/LC_TELEPHONE", O_RDONLY) = 3
open("/usr/lib/locale/en_US.UTF-8/LC_ADDRESS", O_RDONLY) = -1
ENOENT (No such file or directory)
open("/usr/lib/locale/en_US.utf8/LC_ADDRESS", O_RDONLY) = 3
open("/usr/lib/locale/en_US.UTF-8/LC_NAME", O_RDONLY) = -1
ENOENT (No such file or directory)
open("/usr/lib/locale/en_US.utf8/LC_NAME", O_RDONLY) = 3
open("/usr/lib/locale/en_US.UTF-8/LC_PAPER", O_RDONLY) = -1
ENOENT (No such file or directory)
open("/usr/lib/locale/en_US.utf8/LC_PAPER", O_RDONLY) = 3
open("/usr/lib/locale/en_US.UTF-8/LC_MESSAGES", O_RDONLY) = -1
ENOENT (No such file or directory)
open("/usr/lib/locale/en_US.utf8/LC_MESSAGES", O_RDONLY) = 3
open("/usr/lib/locale/en_US.utf8/LC_MESSAGES/SYS_LC_MESSAGES",
O_RDONLY) = 3
open("/usr/lib/locale/en_US.UTF-8/LC_MONETARY", O_RDONLY) = -1
ENOENT (No such file or directory)
open("/usr/lib/locale/en_US.utf8/LC_MONETARY", O_RDONLY) = 3
open("/usr/lib/locale/en_US.UTF-8/LC_COLLATE", O_RDONLY) = -1
ENOENT (No such file or directory)
open("/usr/lib/locale/en_US.utf8/LC_COLLATE", O_RDONLY) = 3
open("/usr/lib/locale/en_US.UTF-8/LC_TIME", O_RDONLY) = -1
ENOENT (No such file or directory)
open("/usr/lib/locale/en_US.utf8/LC_TIME", O_RDONLY) = 3
open("/usr/lib/locale/en_US.UTF-8/LC_NUMERIC", O_RDONLY) = -1
ENOENT (No such file or directory)
open("/usr/lib/locale/en_US.utf8/LC_NUMERIC", O_RDONLY) = 3
open("/usr/lib/locale/en_US.UTF-8/LC_CTYPE", O_RDONLY) = -1
ENOENT (No such file or directory)
open("/usr/lib/locale/en_US.utf8/LC_CTYPE", O_RDONLY) = 3
open(".", O_RDONLY|O_NONBLOCK|O_DIRECTORY|O_CLOEXEC) = 3
```

4.1. -o file -- send trace output to FILE instead of stderr

```
$ strace -o strace.log php --version
$ grep php.ini strace.log
```

5. ltrace - A library call tracer

它的作用：库调用跟踪器。一个库可以包含数百个函数。在这几百个函数中，哪些是我们的二进制程序正在使用的实际函数？

ltrace 命令可以显示运行时从库中调用的所有函数。在下面的例子中，你可以看到被调用的函数名称，以及传递给该函数的参数。你也可以在输出的最右边看到这些函数返回的内容。

ltrace ls

```
neo@netkiller:~/workspace/Document$ ltrace ls
__libc_start_main(0x407bb0, 1, 0x7fff827aea38, 0x413730,
0x413720 <unfinished ...>
strrchr("ls", '/')
= NULL
setlocale(6, "")
= "en_US.UTF-8"
bindtextdomain("coreutils", "/usr/share/locale")
= "/usr/share/locale"
textdomain("coreutils")
= "coreutils"
__cxa_atexit(0x40abb0, 0, 0, 0x736c6974756572, 1)
= 0
isatty(1)
= 1
getenv("QUOTING_STYLE")
= NULL
getenv("LS_BLOCK_SIZE")
= NULL
getenv("BLOCK_SIZE")
= NULL
getenv("BLOCKSIZE")
= NULL
getenv("POSIXLY_CORRECT")
= NULL
getenv("BLOCK_SIZE")
= NULL
getenv("COLUMNS")
```

```
= NULL
ioctl(1, 21523, 0x7fff827ae910)
= 0
getenv("TABSIZE")
= NULL
getopt_long(1, 0x7fff827aea38,
"abcdfghiklmnopqrstuvw:xABCDFGHI:"..., 0x00416a60, -1)
= -1
__errno_location()
= 0x7f89323f16a8
malloc(40)
= 0x02543870
memcpy(0x02543870, "", 40)
= 0x02543870
__errno_location()
= 0x7f89323f16a8
malloc(40)
= 0x025438a0
memcpy(0x025438a0, "", 40)
= 0x025438a0
malloc(18400)
= 0x025438d0
malloc(32)
= 0x025434c0
strlen(".")
= 1
malloc(2)
= 0x025480c0
memcpy(0x025480c0, ".", 2)
= 0x025480c0
__errno_location()
= 0x7f89323f16a8
opendir(".")
= 0x025480e0
readdir(0x025480e0)
= 0x02548108
readdir(0x025480e0)
= 0x02548120
readdir(0x025480e0)
= 0x02548138
readdir(0x025480e0)
= 0x02548150
strlen("Tex")
= 3
malloc(4)
```

```
= 0x02550110
memcpy(0x02550110, "Tex", 4)
= 0x02550110
readdir(0x025480e0)
= 0x02548168
readdir(0x025480e0)
= 0x02548188
strlen("makedoc")
= 7
malloc(8)
= 0x02550130
memcpy(0x02550130, "makedoc", 8)
= 0x02550130
readdir(0x025480e0)
= 0x025481a8
readdir(0x025480e0)
= 0x025481c8
strlen("Docbook")
= 7
malloc(8)
= 0x02550150
memcpy(0x02550150, "Docbook", 8)
= 0x02550150
readdir(0x025480e0)
= NULL
closedir(0x025480e0)
= 0
free(NULL)
= <void>
malloc(72)
= 0x025480e0
_setjmp(0x61c040, 0x25480e0, 0x2543af8, 3, 1)
= 0
__errno_location()
= 0x7f89323f16a8
strcoll("makedoc", "Docbook")
= 9
__errno_location()
= 0x7f89323f16a8
strcoll("Tex", "Docbook")
= 16
__errno_location()
= 0x7f89323f16a8
strcoll("Tex", "makedoc")
= 7
```

```
memcpy(0x025480f0, "\3208T\002", 8)
= 0x025480f0
realloc(NULL, 144)
= 0x02548130
malloc(168)
= 0x025481d0
__errno_location()
= 0x7f89323f16a8
__ctype_get_mb_cur_max(0x7fff827ac0e0, 8192, 0x2550150, -1, 0)
= 6
__ctype_get_mb_cur_max(0x7fff827ac0e0, 8192, 0x2550150,
0x7fff827ac0e0, 0)                                     = 6
__errno_location()
= 0x7f89323f16a8
__ctype_get_mb_cur_max(0x7fff827ac0e0, 8192, 0x2550130, -1, 0)
= 6
__ctype_get_mb_cur_max(0x7fff827ac0e0, 8192, 0x2550130,
0x7fff827ac0e0, 0)                                     = 6
__errno_location()
= 0x7f89323f16a8
__ctype_get_mb_cur_max(0x7fff827ac0e0, 8192, 0x2550110, -1, 0)
= 6
__ctype_get_mb_cur_max(0x7fff827ac0e0, 8192, 0x2550110,
0x7fff827ac0e0, 0)                                     = 6
__errno_location()
= 0x7f89323f16a8
__ctype_get_mb_cur_max(0x7fff827ac110, 8192, 0x2550150, -1, 0)
= 6
__ctype_get_mb_cur_max(0x7fff827ac110, 8192, 0x2550150,
0x7fff827ac110, 0)                                     = 6
__errno_location()
= 0x7f89323f16a8
__ctype_get_mb_cur_max(0x7fff827ac050, 8192, 0x2550150, -1, 0)
= 6
__ctype_get_mb_cur_max(0x7fff827ac050, 8192, 0x2550150,
0x7fff827ac050, 0)                                     = 6
fwrite_unlocked("Docbook", 1, 7, 0x7f8931bab780)
= 7
__overflow(0x7f8931bab780, 32, 0, 8, 0xffffffff)
= 32
__overflow(0x7f8931bab780, 32, 1, 8, 0xffffffff)
= 32
__errno_location()
= 0x7f89323f16a8
__ctype_get_mb_cur_max(0x7fff827ac110, 8192, 0x2550130, -1, 0)
```

```
= 6
__ctype_get_mb_cur_max(0x7fff827ac110, 8192, 0x2550130,
0x7fff827ac110, 0)                                    = 6
__errno_location()
= 0x7f89323f16a8
__ctype_get_mb_cur_max(0x7fff827ac050, 8192, 0x2550130, -1, 0)
= 6
__ctype_get_mb_cur_max(0x7fff827ac050, 8192, 0x2550130,
0x7fff827ac050, 0)                                    = 6
fwrite_unlocked("makedoc", 1, 7, 0x7f8931bab780)
= 7
__overflow(0x7f8931bab780, 32, 1, 8, 7)
= 32
__overflow(0x7f8931bab780, 32, 2, 8, 7)
= 32
__errno_location()
= 0x7f89323f16a8
__ctype_get_mb_cur_max(0x7fff827ac110, 8192, 0x2550110, -1, 0)
= 6
__ctype_get_mb_cur_max(0x7fff827ac110, 8192, 0x2550110,
0x7fff827ac110, 0)                                    = 6
__errno_location()
= 0x7f89323f16a8
__ctype_get_mb_cur_max(0x7fff827ac050, 8192, 0x2550110, -1, 0)
= 6
__ctype_get_mb_cur_max(0x7fff827ac050, 8192, 0x2550110,
0x7fff827ac050, 0)                                    = 6
fwrite_unlocked("Tex", 1, 3, 0x7f8931bab780)
= 3
__overflow(0x7f8931bab780, 10, 0, 120, 3Docbook  makedoc  Tex
)
= 10
free(0x025480c0)
= <void>
free(NULL)
= <void>
free(0x025434c0)
= <void>
exit(0 <unfinished ...>
__fpending(0x7f8931bab780, 0, 0x7f8931bac330, 0x7f8931bac330,
0x25434b0)                                      = 0
fclose(0x7f8931bab780)
= 0
__fpending(0x7f8931bab860, 0, 0x7f8931bacdf0, 0,
0x7f89323f17a0)                                              =
0
fclose(0x7f8931bab860)
= 0
+++ exited (status 0) +++
```

6. nm - list symbols from object files

列出对象文件中的符号。

```
$ nm libzeromq.so
                 U asprintf@@GLIBC_2.2.5
00000000002020d0 B __bss_start
00000000002020d0 b completed.6992
0000000000000f25 T concat
                 w __cxa_finalize@@GLIBC_2.2.5
0000000000000e40 t deregister_tm_clones
0000000000000eb0 t __do_global_dtors_aux
0000000000201de8 t __do_global_dtors_aux_fini_array_entry
00000000002020c8 d __dso_handle
0000000000201df8 d _DYNAMIC
00000000002020d0 D _edata
00000000002020d8 B _end
0000000000001710 T _fini
0000000000000ef0 t frame_dummy
0000000000201de0 t __frame_dummy_init_array_entry
0000000000001ab8 r __FRAME_END__
                 U free@@GLIBC_2.2.5
0000000000202000 d _GLOBAL_OFFSET_TABLE_
                 w __gmon_start__
0000000000000ca8 T _init
                 w _ITM_deregisterTMCloneTable
                 w _ITM_registerTMCloneTable
0000000000201df0 d __JCR_END__
0000000000201df0 d __JCR_LIST__
                 w _Jv_RegisterClasses
                 U malloc@@GLIBC_2.2.5
                 U memcpy@@GLIBC_2.14
                 U memset@@GLIBC_2.2.5
0000000000000e70 t register_tm_clones
                 U __stack_chk_fail@@GLIBC_2.4
                 U strlen@@GLIBC_2.2.5
                 U strncpy@@GLIBC_2.2.5
00000000002020d0 d __TMC_END__
0000000000001399 T zmq_client
000000000000151f T zmq_client_deinit
```

```
000000000001346 T zmq_client_init
                U zmq_close
                U zmq_connect
                U zmq_ctx_destroy
                U zmq_ctx_new
                U zmq_msg_close
                U zmq_msg_data
                U zmq_msg_init
                U zmq_msg_init_size
                U zmq_msg_size
000000000000157d T zmq_publish
0000000000001703 T zmq_publish_deinit
000000000000152a T zmq_publish_init
0000000000000fae T zmq_read
0000000000001157 T zmq_read_deinit
0000000000000f5b T zmq_read_init
                U zmq_recvmsg
                U zmq_sendmsg
                U zmq_socket
00000000000011b5 T zmq_write
000000000000133b T zmq_write_deinit
0000000000001162 T zmq_write_init
```

7. strings - print the strings of printable characters in files.

打印文件中的可打印字符的字符串。

二进制文件中有很多文本信息，比如打印信息、调试信息、帮助信息、错误等。只要这些信息都存在于二进制文件中，就可以用 strings 命令将其转储到屏幕上。

```
[root@localhost ~]# strings /bin/ls | grep "list"
%s: not listing already-listed directory
  -A, --almost-all           do not list implied . and ..
  -B, --ignore-backups       do not list implied entries ending
with ~
  -C                         list entries by columns
  -d, --directory            list directories themselves, not
their contents
  -g                         like -l, but do not list owner
  -G, --no-group             in a long listing, don't print
group names
                             follow symbolic links listed on
the command line
      --hide=PATTERN         do not list implied entries
matching shell PATTERN
  -I, --ignore=PATTERN       do not list implied entries
matching shell PATTERN
  -l                         use a long listing format
  -m                         fill width with a comma separated
list of entries
  -n, --numeric-uid-gid      like -l, but list numeric user and
group IDs
  -o                         like -l, but do not list group
information
  -R, --recursive            list subdirectories recursively
  -U                         do not sort; list entries in
directory order
  -x                         list entries by lines instead of
by columns
  -1                         list one file per line.  Avoid
```

```
'\n' with -q or -b
```

8. hexdump

以 ASCII、十进制、十六进制或八进制显示文件内容。

```
[root@localhost ~]# hexdump -C /bin/ls | head
00000000  7f 45 4c 46 02 01 01 00  00 00 00 00 00 00 00 00
|.ELF............|
00000010  03 00 3e 00 01 00 00 00  00 5e 00 00 00 00 00 00
|..>......^......|
00000020  40 00 00 00 00 00 00 00  38 0b 04 00 00 00 00 00
|@.......8.......|
00000030  00 00 00 00 40 00 38 00  0a 00 40 00 1f 00 1e 00
|....@.8...@.....|
00000040  06 00 00 00 04 00 00 00  40 00 00 00 00 00 00 00
|........@.......|
00000050  40 00 00 00 00 00 00 00  40 00 00 00 00 00 00 00
|@.......@.......|
00000060  30 02 00 00 00 00 00 00  30 02 00 00 00 00 00 00
|0.......0.......|
00000070  08 00 00 00 00 00 00 00  03 00 00 00 04 00 00 00
|................|
00000080  70 02 00 00 00 00 00 00  70 02 00 00 00 00 00 00
|p.......p.......|
00000090  70 02 00 00 00 00 00 00  1c 00 00 00 00 00 00 00
|p...............|

neo@MacBook-Pro-Neo ~ % hexdump -C /bin/ls | head
00000000  cf fa ed fe 07 00 00 01  03 00 00 00 02 00 00 00
|................|
00000010  13 00 00 00 10 07 00 00  85 00 20 00 00 00 00 00
|.......... .....|
00000020  19 00 00 00 48 00 00 00  5f 5f 50 41 47 45 5a 45
|....H...__PAGEZE|
00000030  52 4f 00 00 00 00 00 00  00 00 00 00 00 00 00 00
|RO..............|
00000040  00 00 00 00 01 00 00 00  00 00 00 00 00 00 00 00
|................|
00000050  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
|................|
00000060  00 00 00 00 00 00 00 00  19 00 00 00 28 02 00 00
```

```
|............(...|
00000070  5f 5f 54 45 58 54 00 00  00 00 00 00 00 00 00 00
|__TEXT..........|
00000080  00 00 00 00 01 00 00 00  00 50 00 00 00 00 00 00
|.........P......|
00000090  00 00 00 00 00 00 00 00  00 50 00 00 00 00 00 00
|.........P......|
```

9. objdump - display information from object files.

从对象文件中显示信息。

```
[root@localhost ~]# objdump -d /bin/ls | head

/bin/ls:     file format elf64-x86-64


Disassembly of section .init:

0000000000003538 <.init>:
    3538:       f3 0f 1e fa             endbr64
    353c:       48 83 ec 08             sub    $0x8,%rsp
    3540:       48 8b 05 69 da 21 00    mov
0x21da69(%rip),%rax        # 220fb0 <__gmon_start__>
```

10. readelf - Displays information about ELF files.

显示有关 ELF (Executable and Linkable File Format 可执行和可链接文件格式) 文件的信息。

```
[root@localhost ~]# readelf -h /bin/ls
ELF Header:
  Magic:   7f 45 4c 46 02 01 01 00 00 00 00 00 00 00 00 00
  Class:                             ELF64
  Data:                              2's complement, little
endian
  Version:                           1 (current)
  OS/ABI:                            UNIX - System V
  ABI Version:                       0
  Type:                              DYN (Shared object file)
  Machine:                           Advanced Micro Devices
X86-64
  Version:                           0x1
  Entry point address:               0x5e00
  Start of program headers:          64 (bytes into file)
  Start of section headers:          265016 (bytes into file)
  Flags:                             0x0
  Size of this header:               64 (bytes)
  Size of program headers:           56 (bytes)
  Number of program headers:         10
  Size of section headers:           64 (bytes)
  Number of section headers:         31
  Section header string table index: 30
```

11. Valgrind

http://valgrind.org/

```
valgrind --tool=memcheck --leak-check=full ./test
```

第 30 章 GNU Development Tools

1. strip - Discard symbols from object files.

给C程序减肥

```
# cp nginx nginx.old
# strip nginx
# ll
total 4984
-rwxr-xr-x  1 root root  545080 Oct 18 10:48 nginx
-rwxr-xr-x. 1 root root 4554524 May  7 17:18 nginx.old
```

部分 V. D Programming Language

第 31 章 D Lang

http://dlang.org/

1. dmd install

Ubuntu

```
$ wget http://ftp.digitalmars.com/dmd_2.061-0_amd64.deb
$ sudo apt-get install libc6-dev

$ sudo dpkg -i dmd_2.061-0_amd64.deb
```

CentOS

```
wget http://ftp.digitalmars.com/dmd-2.062-0.fedora.x86_64.rpm
yum localinstall dmd-2.062-0.fedora.x86_64.rpm
```

2. helloworld

```
$ cat hello.d
import std.stdio;
void main() {
  writeln("Hello, world!");
}
```

```
$ chmod u+x hello.d
$ ./hello.d
Hello, world!
```

```
$ dmd hello.d
$ ./hello
Hello, world!
```

```
$ scp hello root@172.16.0.3:/tmp

# cd /tmp/
# ./hello
Hello, world!
```

3. dmd - Digital Mars D2.x Compiler

3.1. -cov do code coverage analysis

$ dmd hello.d -cov

```
$ cat hello.d
#!/usr/bin/rdmd
import std.stdio;
void main() {
    writeln("Hello, world!");
}

$ dmd hello.d -cov

$ ./hello
Hello, world!

$ cat hello.lst
       |#!/usr/bin/rdmd
       |import std.stdio;
       |void main() {
      1|    writeln("Hello, world!");
       |}
hello.d is 100% covered
```

4. Open Source Development for the D Programming Language

http://www.dsource.org/

4.1. DDBI - A database independent interface.

```
$ git clone http://github.com/aaronc/ddbi/
```

第 32 章 FAQ

1. /lib64/libc.so.6: version `GLIBC_2.14' not found

```
# strings /lib64/libc.so.6 |grep GLIBC_
GLIBC_2.2.5
GLIBC_2.2.6
GLIBC_2.3
GLIBC_2.3.2
GLIBC_2.3.3
GLIBC_2.3.4
GLIBC_2.4
GLIBC_2.5
GLIBC_2.6
GLIBC_2.7
GLIBC_2.8
GLIBC_2.9
GLIBC_2.10
GLIBC_2.11
GLIBC_2.12
GLIBC_PRIVATE
```

第 33 章 数据可视化

1. matplotlib

1.1. 直方图

```
import matplotlib.pyplot as plt
import pandas as pd

df = pd.read_excel("data1.xlsx", "Sheet1")

fig = plt.figure()
ax = fig.add_subplot(1, 1, 1)
ax.hist(df['年龄'], bins=7)
plt.title('Age distribution')
plt.xlabel('Age')
plt.ylabel('Employee')
plt.show()
```

1.2. 显示中文

```
import matplotlib.pyplot as plt
import pandas as pd
from matplotlib.font_manager import FontProperties

title =
FontProperties(fname=r"/System/Library/Fonts/PingFang.ttc",
size=14)
font =
FontProperties(fname=r"/System/Library/Fonts/PingFang.ttc",
size=10)
```

```
df = pd.read_excel("data.xlsx", "Sheet1")

fig = plt.figure()
ax = fig.add_subplot(1, 1, 1)
ax.hist(df['年龄'], bins=7)
plt.title('年龄分布图', fontproperties=title)
plt.xlabel('年龄', fontproperties=font)
plt.ylabel('员工数量', fontproperties=font)
plt.show()
```

2. pyecharts

https://pyecharts.org/

安装

```
$ pip install pyecharts
```

第 34 章 UML

UML中的视图分为5大类(每一类的名称都有好几种说法，但表示的意思是差不多的，下面主要是按照EA中的分法）：

a)　　用例视图(Use Case View)，强调从用户角度看到的或需要的系统功能，是被称为参与者的外部用户所能观察到的系统功能的模型图。

b)　　动态视图(Dynamic View)，体现了系统的动态或者行为特征，也称为行为模型视图(Behavioral Model View)或并发视图(Concurrent View)。

c)　　逻辑视图(Logical View)，展现系统的静态或结构组成及特征，也被称为结构模型视图(Structural Model View)或者静态视图(Static View)。

d)　　组件视图(Component View)，体现了系统实现的结构和行为特征，也称为实现模型视图(Implementation Model View)。

e)　　配置视图(Deployment View)，体现了系统实现环境的结构和行为特征，也被称为环境模型视图(Environment Model View)或者物理视图(Physical View)。

在EA中还有一个Custom,其相当于设计者自己定义的一个视图，并不是UML的定义。

UML中的图有9种：

a)　　用例图(Use Case Diagram)，描述系统功能；

b)　　类图(Class Diagram)，描述系统的静态结构；

c)　　对象图(Object Diagram)，描述系统在某个时刻的静态结构；

d)　　时序图(Sequence Diagram)，按时间顺序描述系统元素间的交互；

e)　　协作图(Collaboration Diagram)，按照时间和空间顺序描述系统元素间的交互和他们之间的关系；

f)　　状态图(State Diagram)，描述了系统元素的状态条件和响应；

g)　　活动图(Activity Diagram)，描述了系统元素的活动；

h)　　组件图(Component Diagram)，描述了实现系统的元素的组织；

i)　　配置图(Deployment Diagram)，描述了环境元素的配置，并把实现系统的元素映射到配置上。

在UML中视图是由图构成的，视图和图之间的对应关系：
用例视图：用例图
动态视图：时序图、协作图、状态图和活动图
逻辑视图：类图和对象图
组件视图：组件图
配置视图：配置图

第 35 章 Regular expression (正则表达式)

1. Network 网络地址处理

```
$ wget -q -O - checkip.dyndns.org|sed -e 's/.*Current IP
Address: //' -e 's/<.*$//'
202.130.101.34

$ curl -q -s http://checkip.dyndns.org | egrep -o '[0-9]+\.[0-
9]+\.[0-9]+\.[0-9]+'
202.130.101.34

$ curl -q -s http://checkip.dyndns.org | egrep -o '[0-9]{1,3}\.
[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}'
202.130.101.34
```

2. HTML 处理

url 抓取

```
$ curl -sq http://www.163.com | grep -o -E 'http://([^"#]+)'
```

第 36 章 Miscellaneous

1. php function check

```
#!/bin/bash
LOGFILE=/tmp/my.log
echo > $LOGFILE
for helper in `ls -1 class/helper/`
do
    echo ========================== $helper
============================ >> $LOGFILE
    class=`grep '^class' class/helper/$helper | awk -F ' '
'{print $2}'`
    for fun in `grep 'public function [a-zA-Z]'
class/helper/$helper | awk -F ' ' '{print $3}' | awk -F '('
'{print $1}'`
    do
        count=`grep -r "$class->$fun(" *|wc -w`
        if [ $count == 0 ]; then
               echo "[ unused ] $class->$fun" >> $LOGFILE
        else

               echo "[  used  ] $class->$fun" >> $LOGFILE
        fi
        echo "[`date`] [$helper] $class->$fun (checked:
$count)"
    done
done
```

2. whois 域名查询

```
<html>
<head>
<meta http-equiv="content-type" content="text/html;
charset=UTF-8">
<title>whois</title>
</head>
<body>
<fieldset>
<legend>whois</legend>
<form name="form1" method="post" action="<? $PHP_SELF ?>">
<input type="text" name="domainname">
.cn
<input type="submit" name="Submit" value="查询">
</form>
</fieldset>

查询域名:
<?echo $domainname;?>
.cn
<?php
$fp = fsockopen ("whois.cnnic.cn", 43 , $errno, $errstr, 30);
if (!$fp) {
echo "$errstr ($errno)<br>\n";
} else {
fputs ($fp, "$domainname".".cn"."\r\n");
echo "<pre>";
while (!feof($fp)) {

$data = fgets ($fp,1024);
$data = str_replace("no matching record", "该域名没有被注册\n<a
href='http://www.cnwwwcn.com'>我想注册该域名</a>", $data);
/*
$data = fgetc ($fp);

if($data == "\n"){
echo "<br>";
}
*/
echo $data;
//no matching record
}
echo "</pre>";
fclose ($fp);
}
?>
</body>
</html>
```

3. 身份证校验

```
<?php

function check_id_number($no){
        if (strlen($no) != 18){
                return false;
        }
        $sigma = 0;
        $wi = array(7, 9, 10, 5, 8, 4, 2, 1, 6, 3, 7, 9, 10,
5, 8, 4, 2);
        $ai = array('1', '0', 'X', '9', '8', '7', '6', '5',
'4', '3', '2');
        for ($i = 0;$i < 17;$i++) {
            $sigma += ((int) $no{$i}) * $wi[$i];
        }

        if (substr($no,17) == $ai[($sigma % 11)]){
                return true;
        }else{
                return false;
        }
}

echo check_id_number('330702198003090915');
```

4. PHP PDF 处理库

TCPDF is a FLOSS PHP class for generating PDF documents.

http://www.tcpdf.org/

5. Kint - a modern and powerful PHP debugging helper

http://raveren.github.io/kint/

6. snoopy 模拟浏览器操作

http://snoopy.sourceforge.net/

7. PHP Nightrain

Using PHP Nightrain you will be able to deploy and run HTML, CSS, JavaScript and PHP web applications as a native desktop application on Windows, Mac and the Linux operating systems. Popular PHP Frameworks (e.g. CakePHP, Laravel, Drupal, etc…) are well supported!

http://www.naetech.com/php-nightrain

第37章 FAQ

1. /lib64/libc.so.6: version `GLIBC_2.14' not found

```
# strings /lib64/libc.so.6 |grep GLIBC_
GLIBC_2.2.5
GLIBC_2.2.6
GLIBC_2.3
GLIBC_2.3.2
GLIBC_2.3.3
GLIBC_2.3.4
GLIBC_2.4
GLIBC_2.5
GLIBC_2.6
GLIBC_2.7
GLIBC_2.8
GLIBC_2.9
GLIBC_2.10
GLIBC_2.11
GLIBC_2.12
GLIBC_PRIVATE
```