



*The Newsletter of UKUUG, the UK's Unix and Open Systems Users Group*  
Published electronically at <http://www.ukuug.org/newsletter/>

---

**Volume 20, Number 1**

**ISSN 0965-9412**

**March 2011**

---

## **Contents**

<b>News from the Secretariat</b>	<b>3</b>
<b>From the Chairman</b>	<b>4</b>
<b>OpenTech 2011 – Saturday 21 May, London</b>	<b>4</b>
<b>Advanced Perl Course review</b>	<b>5</b>
<b>Review of the Perl tutorials: February 2011</b>	<b>7</b>
<b>Book review: Real World Instrumentation (with Python)</b>	<b>7</b>
<b>Book review: SQL Pocket Guide (Third Edition)</b>	<b>8</b>
<b>Book review: Data Analysis with Open Source Tools</b>	<b>9</b>
<b>Book review: Cassandra, The Definitive Guide</b>	<b>9</b>
<b>Book review: Head First Python</b>	<b>11</b>
<b>Book review: Building Android Apps with HTML, CSS and JavaScript</b>	<b>12</b>
<b>Book review: Pragmatic Guide to Git</b>	<b>12</b>
<b>Book review: Pragmatic Guide to JavaScript</b>	<b>13</b>
<b>Book review: Pragmatic Guide to Subversion</b>	<b>13</b>
<b>Contributors</b>	<b>14</b>
<b>Contacts</b>	<b>15</b>



## News from the Secretariat

*Jane Morrison*

Thank you to everyone who has kindly sent in their subscription payments promptly. We have received a number of early payments. Those remaining outstanding will be chased this month and any not paid at the end of April will not receive the next issue (June) Newsletter.

We now have everything in place for the UKUUG Spring Conference and Tutorials, being held in Leeds (22nd - 24th March). We have received sponsorship monies once again from Google and Bytemark which has enabled us to organise a Conference Dinner in the famous Oriental Gallery at the Royal Armouries.

We have also been able to keep the event delegate fees low for members through sponsorship from Novell.

A full list of talks can be found on the web site together with all abstracts and speakers biographies, see: <http://www.ukuug.org/events/spring2011/>.

Remember that by attending the Spring event you can:

- Keep abreast with new/emerging technologies
- Network with some of the people who are responsible for developing critical applications
- Become part of the UK Unix community – build up informal relationships that can be invaluable in problem solving
- Benefit from the experience of delegates with similar interests
- Keep staff happy and feeling valued
- Make valuable contacts with sponsor organisations such as Google, Bytemark and Novell
- Secure essential O'Reilly books at a significant discount to the cover price

At the time of writing I can confirm that the Intermediate and Advanced Perl tutorials being held from 15th–18th February are again popular and we hope the new format of 2-day hands-on courses will be successful.

Future events include OpenTech and tutorials jointly arranged with O'Reilly:

19th May: 'Debian Packaging Tutorial' – by Phil Hands. Bookings are already coming in for this event. Be sure to book soon: <http://www.ukuug.org/events/debian/>.

22nd September: 'Building the network you need with PF, the OpenBSD packet filter' – by Peter N. M. Hansteen. Full information is enclosed with this Newsletter.

OpenTech 2011 (ULU, Malet Street, London, WC1E 7HY) will be held on Saturday 21st May. Full details together with the facility to book ticket reservations on-line will be available soon at: <http://www.ukuug.org/events/opentech2011/>.

The UKUUG AGM will be held on Thursday 22nd September - please put this date in your diary now.

After the successful un-conference held last year in Birmingham, the event for 2011 will be held in Manchester on Saturday 8th October. Put the date in your diary now.

As you can see we have been working hard to bring you a full schedule of events for the coming year.

The next Newsletter will be the June issue and the copy date is: Friday 20th May.

Please feel free to feed back any comments about events, or anything else to: [newsletter@ukuug.org](mailto:newsletter@ukuug.org).

## From the Chairman

*Paul Waring*

### Upcoming Conferences

Our Spring conference in Leeds is only a few weeks away, and the full programme is available on our website. Topics including IPv6, MogileFS, OpenStack, DNSSEC and many more will be covered over two days of talks, along with a full day tutorial on Haskell.

There is still time to book a place for the tutorial and/or the conference, and I hope you will be able to join us for another enjoyable three day programme.

### Get Involved

UKUUG exists to serve its members and we are always on the lookout for people who are keen to get involved in any capacity, whether that be through volunteering to help with organising events, writing newsletter articles, or entirely new activities which haven't been tried before. We are particularly keen to get members involved in running the October unconference and next year's Spring conference. If you would like to help out in any capacity please do get in touch via [office@ukuug.org](mailto:office@ukuug.org).

---

## OpenTech 2011 – Saturday 21 May, London

*Sam Smith*

OpenTech 2011 is an informal, low-cost, one-day conference on slightly different approaches to technology, politics and transport.

### What's the day going to have?

- Technology
- Transport
- Democracy
- Now with added hardware!
- you...

### What are we looking for?

- Proposals from people who want to give a presentation, run a panel, organise a tutorial, or run a demo of something new and interesting.
- Publicity – please blog/tweet/forward and tell your friends!
- anything that we should have that we don't know we're looking for

### What topics do we hope to cover?

Technology, Democracy, Transport (and travel), London – and other cities, Sustainability – how to turn hackdays and demos into something more permanent, Open Hardware – doing things technologies were perhaps not intended to do.

If you've got an interesting proposal that doesn't fit into any of the categories above, please send it in anyway!

### What have we already got talks or sessions about?

- Community linked data
- FixMyTransport.org

- The Geek Manifesto
- Security: X is bonkers. For X in PHP, and more
- Talking to non-techies: experiences of Fossbox
- Open Source Hardware
- TroubleMakers Inc's latest (legal) project. . .
- Hard curves, soft electronics – code, tech and textiles
- The Quantified Self
- Media, Standards and Journalism
- Open Data Cities
- Innocent Enough Questions. . . – including “what does the government spend money on?”
- and other vagueish things being firmed up

We're still looking for more talks on all our topics, so if you want to offer something, we're waiting to hear your ideas.

#### How do I submit a proposal?

Online form at <http://www.ukuug.org/opentech>.

#### Can I buy or reserve a ticket to the event?

Join our mailing list at <http://www.ukuug.org/opentech> and we'll email you nearer the time with more information.

---

## Advanced Perl Course review

*Martin Houston*

This course was given by Dave Cross in London on 18th and 19th Feb.

I have been using Perl since the early 1990s and probably write ‘quick and dirty’ Perl code almost every day. If people want a quick bit of data munging they know where to come. . .

The problem is that where I have worked the past few years Perl is not one of the officially approved of languages, I am seen somewhat as a rebel so there has been no driving force to make the functional and effective Perl I do write ‘better’ and ‘modern’.

Perl has moved on a lot in the last few years, on the whole quietly without great fanfare, so I thought taking a couple of days out to see what ‘Advanced Perl’ means in 2011 would be a worthwhile exercise.

Like many people who don't have ‘Perl Developer’ officially part of their current job title I have been ‘stuck’ on the Perl 5.8 that comes as standard with RedHat EL5. This is a shame as I have been missing out on a lot of nice little improvements in Perl 5.10 and 5.12. Each one not earth shattering but together make writing good, easy to maintain programs that little bit easier.

The ‘Defined-Or’ // operator is one.

Instead of having to write things like:

```
$x = 'initial value' unless defined $x;
```

one can use far fewer key-presses:

```
$x //= 'initial value'
```

(like in the bash shell `{x:=initial value}` but nicer syntax.

`say` is another simple little time saver, its just like `print` but appends a newline on the end. The saving in typing mounts up:

```
say 'hello,world' is 16 characters; print 'hello,world' . "\n" is 24!
```

I am not going to take up space with all the new features, the one I will probably find most useful, rather than just time saving is the named capture buffers in regexes.

```
/(?<telephonenumber>\d+)\s+(?<name>\w+)/ if it matches will make the %+ hash contain telephonenumber and name fields.
```

The beauty of this is that unlike the old \$1 and \$2 way of matching capture buffers someone could decide that the data is better the other way round and the only thing that would have to change is the regular expression itself. It also makes the regular expressions far more meaningful in that the buffer names serve as comments as to the meaning of that particular string of numbers.

As Perl regular expression syntaxes tend to get stolen by other languages expect this new feature to arrive in Python and Java sometime soon too.

The course also covered testing – using `Test::Builder` and related modules to define the unit testing framework that has always been a great strength of the Perl core modules.

`Test::MockObject` so that the ‘outside world’ can be simulated enough for unit tests to exercise their code. Finding out just how much of your code your test suite has managed to test is revealed by the `Devel::Cover` tool which produces some nice reports that are in HTML so they can look great in a browser to show your boss!

Next the fast moving course went on to cover Object Relational Mapping – using the `DBIx::Class` tools to either turn Perl object relationships into matching SQL tables or working from an existing database and going the other way. Lots of tedious writing SQL code directly using DBI is now mostly a thing of the past. We did a practical exercise with `DBIx::Class` by turning a simple ‘CD Collection’ database into something accessible from Perl with a few simple commands.

Exception handling was next on the agenda – Modern Perl can now throw objects as exceptions – in fact it always could – the `$@` variable used for communication between `die` and an enclosing `eval` can be an object instead of a simple string. So we now have Python- and Java-like try/catch without needing fundamental changes to the language. Perl is just so flexible!

Day two started with benchmarking and the related subject of profiling. The wonderful module `Devel::NYTProf` was introduced to us. Another producer of lavish web pages full of detailed performance information about your code, what needs to have some work done to benchmark possible faster implementations and what code is so little used that just being correct rather than efficient is good enough.

Another bit of time and code saving that is in increasing use is Moose – a modern object system for Perl. Using Moose means a lot of the repetition is taken out of designing objects. The programmer can rely on the libraries to do some of the routine stuff and just concentrate on what is different about this particular program.

The whole theme comes together with an exploration of Catalyst and Plack. A way to produce a framework for a typical web based application in a way that allows the programmer to just concentrate on providing strategic islands of new code. Catalyst brings together `DBIx::Class` to access underlying database tables with other parts of the modern Perl armoury. We managed to take our humble CD database to a full web application with only small but strategic changes to the Perl code framework produced by Catalyst.

At each stage the tutorial sessions allowed us to get simple examples of each concept working – just enough to be able to build on in private study later.

I now feel I would not be intimidated to tackle a job that used any of the new Perl buzzwords and I think that is a good sign of a successful course.

---

## Review of the Perl tutorials: February 2011

*Howard Thomson*

I attended the tutorials on Perl, presented by Dave Cross, with two days on Intermediate Perl, and two days on Advanced Perl.

As an Eiffel and C (and APL) programmer, and one who has not written any Perl code, I attended both as courier for UKUUG's projector, and an interested observer.

I suspect that the main reasons why Perl is popular are the fact that the starting point for coding in Perl is minimal, and the draw of the huge number of packages in the CPAN archive.

Having since discovered that the Perl system can be embedded in a C, and therefore also Eiffel, program, I am interested in being able to write sufficient Perl to be able to use (more importantly, reuse) code that is already available, albeit not written in my preferred language.

Dave noted that the object-oriented aspects of Perl are 'bolted on', achieving the best of both worlds.

I have to agree about the 'bolted on' aspect in that, like attempting (or succeeding) in writing object oriented code directly in C, there is a lot of boiler-plate implementation stuff that has to be written explicitly in order to write using the OO capabilities of Perl.

Within the constraints of the language, I think Dave Cross did an admirable job of explaining Perl's concepts of packages, types, references, argument passing, sorting, and many of the less obvious traps and pitfalls for the unwary, most of which I would, and probably will, fall into if and when I start writing Perl code.

As with any language, proficiency in Perl depends on familiarity with the concepts, the syntax and expression of those concepts, and knowledge of common code combinations.

With coverage of new facilities in Perl 5.10 and 5.12, Testing, Database access, Exception handling, Profiling and benchmarking, Moose, MVC frameworks (Catalyst) and PSGI/Plack, there was an enormous amount to take in (or not as the case may be), especially for a non-Perl programmer. However, I learnt a lot about the capabilities of Perl combined with the CPAN archive, and while I don't have a desire to write more in Perl than I have to, the ability to reuse existing packages may prove irresistible.

---

## Real World Instrumentation (with Python)

**J M Hughes**

**O'Reilly Media**

**ISBN: 978-0-596-80956-0**

**624pp.**

**£ 42.50**

**Published: Nov 2010**

**reviewed by John Pinner**

As a resident UKUUG Python advocate, I was 'landed' with the task of reviewing this book for our Newsletter. However, the task, or chore, has proved to be a pleasure, and thanks are due to the author for that.

J M Hughes is an experienced systems engineer (he was responsible for surface imaging software on the Phoenix Mars Lander) the book reflects that experience, covering every area of automated data acquisition and control. Currently, he is using Python for verifying the alignment of mirror segments on the James Webb telescope, and I expect that this provided the motivation for writing this book.

However, this is much more than a book about Python programming, which is really incidental to a complete guide to Instrumentation, Control, Electronics (both theory and construction), Data Acquisition and Storage. Programming (with Python and C) is just a means to an end.

The book is planned well: each section deals with basic theory before moving onto hands-on practical work, and where appropriate finishes with a useful bibliography. I found myself thinking “I wish I’d had this book when I first started working with ...”, and this applied to Instrumentation, 3-term controllers, Project Planning, and even User Interfaces.

The Python content deliberately avoids the more esoteric features beloved of Python enthusiasts, allowing the engineer user to concentrate on getting the job done without having to learn advanced language features. Nonetheless the advice and examples given are good, although I did see one example with a ‘code smell’ using global variables to excess.

There is a useful introduction to extending Python with C, as might be needed to interface with a new instrument, and this itself builds on a Chapter about the C language – an excellent introduction starting with the basics and finishing with advice on libraries and build tools, all in 40 pages.

The author manages to strike an even-handed balance between ‘nix and Windows operating systems, so that people like me who have managed to avoid Windows as a development platform will find a lot to help them if they cannot avoid it any longer.

The Chapter on interfaces deals with serial (RS32/RS485/USB) and parallel (GPIB) interfaces, and having been mostly concerned with the GPIB lately, I jumped straight there, to remember that there is little Free Software to support GPIB in the Python world, and making me feel guilty that we have not publicised our own Python GPIB drivers enough (reminder to self – do something about this!).

Often, Data Storage and User Interfaces are neglected by engineers, so it was good to see these dealt with in dedicated chapters, and I found at least one thing to help me with a current project. This is one of the perks of reviewing books for UKUUG, so why not contact Jane at [office@ukuug.org](mailto:office@ukuug.org) if you would like to join the review team?

Highly recommended.

---

## **SQL Pocket Guide (Third Edition)**

**Jonathan Gennick**

**O’Reilly Media**

**ISBN: 978-1-4493-9409-7**

**208pp.**

**£ 11.50**

**Published: November 2010**

**reviewed by Mike Fowler**

This is the third edition of the pocket guide covering DB2, MySQL, Oracle, PostgreSQL and SQL Server. It covers everyday SQL statements in enough detail to be useful without going to the depths that SQL In A Nutshell does. All the examples in the book work on a sample database that can be downloaded from O’Reilly and I loaded both the PostgreSQL and MySQL versions without error.

Four topics in the book get very thorough coverage that are worth noting. Datatypes are grouped by type showing ANSI syntax with platform specific syntax where required. The other three are grouped by the platform and cover numeric conversions, datetime conversions and datetime functions. For each of these different conversions the book provides the full list of options, such as SQL Server’s annoying CONVERT styles.

Unfortunately for a pocket book, it does not lend itself well for a quick thumb through. Although organised alphabetically by topic, the headings and sub-headings are not very different. I often found myself checking the table of contents to determine whether I had found the start of topic or was midway through one.

Having test driven the book over the last few weeks I found it rather useful. I'm no stranger to SQL but I often jump between all of the platforms covered (except DB2) and usually fail to remember how each platform differs from the last. With this little book on my desk I won't be reading the online manuals every other day.

---

## **Data Analysis with Open Source Tools**

**Philipp K. Janert**

**O'Reilly Media**

**ISBN: 978-0-596-80235-6**

**536pp.**

**£ 30.99**

**Published: November 2010**

**reviewed by Mike Fowler**

This book provides a thorough treatment of the field of data analysis. It is divided into four parts covering graphical representations, data modelling, computational data mining and general applications. Each part ends with an "Intermezzo": a long worked example that puts all the theory covered in that part into practice. Each of the chapters of a part (including the intermezzo) end with a workshop that shows you how to put the topics to use with an open source tool. These workshops each introduce a new open source tool and gives you the opportunity to play with the techniques covered.

Although the author proclaims that not much mathematical knowledge is required, I think it is a considerable prerequisite to getting the most out of this book. A useful refresher of the major mathematical principles used in the book is provided as an appendix, but it can not be used to learn calculus from scratch for example. The ill-prepared non-mathematical reader will do well to follow the suggested further readings at the end of each chapter.

I was a touch disappointed by the overall lack of coverage of the open source tools. However this disappointment is short lived once you appreciate the effort that has clearly been put into covering the mathematical theory that underpins all that the open source tools do. After all, if you don't know how best to present and analyse your data, what good is a tool? Certainly having read this book I will no longer be simply showing the means and standard deviations of my data.

---

## **Cassandra, The Definitive Guide**

**Eben Hewitt**

**O'Reilly Media**

**ISBN: 978-1-4493-9041-9**

**336pp.**

**£ 30.99**

**Published: November 2010**

**reviewed by Mike Smith**

I had the opportunity to review an O'Reilly book on MongoDB for the last newsletter, which I enjoyed (and I hope you did) as my interest has been piqued recently by nosql (some people say "Not Only SQL") databases. When I saw this new Cassandra book I thought a review of this would be a fitting sequel (lol, pun not intended but I'll leave it in) – so I jumped at the chance. Cassandra appears to be one of the most scalable database options of its type at present, originating at Facebook, and now used within the infrastructures for Twitter, Digg and elsewhere. (Indeed it was Digg's troubles whilst moving to Cassandra that started me off on this path of discovery.)

---

I really liked the fact that this book recognises the reader is likely to have some RDBMS understanding. So it takes time to explain the differences. In particular chapter 3 describes the data model to help us on our way to learning what column-oriented databases are all about (this follows chapters that offer a general introduction and installation instructions).

We then move quickly on to a simple example application and we're up and running quickly with Cassandra. Again parallels are drawn with RDBMS concepts, and we use YAML to describe the database structure and code in Java. There are some leaps one has to overcome here – for instance how do you import YAML into the system to create the columns? (All is explained much later in the book.) I was a little disappointed when I read that the example provided in the book was not complete because of space reasons. However that quickly died away when I hit the O'Reilly website and found the full code for download.

Cassandra is used for very scalable architectures where, most likely, you're going to need multiple database nodes to store the data. Clustering is therefore inherently supported. When creating a keyspace you set the replication factor (how many nodes data is stored on) depending on performance and data consistency requirements. All this replication is managed and performed by a peer to peer protocols between the nodes. It's wonderful. I learnt all this from Chapter 3 too.

Make sure you investigate Brewer's CAP theorem for distributed data systems. That had passed me by, even though it's over 10 years old, but it's darned interesting – I had no idea of the relationship between consistency, availability and partition tolerance. You can only have 2 of the three – and actually, as you always get network issues (sorry network guys), you can only have data consistency OR availability – not both. That's a pretty awesome conclusion.

We've reached chapter 5 by now, and here we get serious. This chapter covers some Cassandra architecture and internals. It is actually quite interesting how Cassandra uses a bunch of advanced technologies to underpin its services, including a peer-to-peer distribution model that provides a masterless mode of clustering; a Gossip protocol for communication to overcome faulty networks; anti-entropy and read-pair techniques. There's a whole host of other exciting terms in this chapter too – you must read it to find out what they are! Suffice to say, all quite well explained.

The next chapter describes replicas and replica replacement strategies (how data is copied onto multiple nodes), snitches and partitioners. All about how the data is split up and managed across the cluster. An important aspect covered next is Cassandra's pluggable authentication system. (By default client access is possible without any authentication, so you'll be changing this in the real world.)

It's hard to believe we've got this far and haven't really talked about properly reading and writing to the database or about the Cassandra API. We rectify this in the next chapter, also learning about slices and ranges.

The client environment gets a look-in next, including mechanisms to connect to the cluster. You can connect to any node, it makes no difference – but maybe you want to arrange to connect to a local one in geographic or network terms. You don't use a driver (like JDBC) to connect to the database. You use either direct API calls or high level clients developed by others that you call from your own code, be it Ruby, PHP, Python, Perl etc. Some recommendations are made here too.

Operations next – logging, monitoring using JMX, maintenance activities – cleanly removing nodes and the like. Then a chapter on performance tuning.

There's also an entire chapter (the final one) dedicated to the integration of Cassandra with Hadoop. If you're not familiar with Hadoop, this is an open source implementation of Google's MapReduce algorithm (together with a distributed filesystem). There's some example code for a MapReduce job together with an explanation of some useful tools.

To round off the book there's an appendix that covered the wider nosql environment, with descriptions of the various databases (including MongoDB) or various types (XML, columnar, graph etc), and finally a useful glossary.

You can usually tell with my reviews whether I like a book or not. Have you guessed yet?

Well, of course in this case I approve. Readable, interesting, easy for newbies. In fact I've held back from telling you everything – there are other little treats in the book, so you jolly well need to go out and buy it.

---

## **Head First Python**

**Paul Barry**

**O'Reilly Media**

**ISBN: 978-1-4493-8267-4**

**496pp.**

**£ 38.50**

**Published: November 2010**

**reviewed by Roger Whittaker**

When I was a teacher of mathematics, I was dissatisfied with many of the textbooks that were available. There were some that presented the material in a dry and formal way, and did not attempt to go into the question of why any of it was useful or what its purpose was. There were others that tried to be more “accessible” by presenting examples of the type of problem that motivated a particular piece of mathematics. But these often failed the “rigour” test. It is very hard to get this balance right, because introducing a new abstract concept works best as a bottom-up process, while properly and accurately describing it works best top-down.

It seems to me that the “Head First” series of books is an attempt to bridge the gap between these two approaches, and the blurb on the back cover claims that the book “*uses a visually rich format designed for the way your brain works, not a text-heavy approach that puts you to sleep*”.

There is no doubt that it is visually rich in a sense: many of the pages contain a single cartoon or “retro” 1950s-style photo with speech bubble introducing the next topic. Tastes will vary: I began to find these slightly irritating rather than amusing or motivating at a fairly early stage.

But I do salute the attempt to introduce topics by example and motivate the discussion with concrete reasons for everything that is done. This is a good approach for beginners who have not been hardened by the frustrations of trying to understand things from more formal reference documentation.

This is a fairly “gentle” introduction to Python, and because of the style and layout, contains less information that you might guess purely from the book's size. Python 3 is used in most of the book. The first half of the book introduces the language and lists, functions, dictionaries, file I/O, exception handling, classes and other basic concepts. Then at Chapter 7 (of 11) we see the dreaded heading “Putting it all together” and we move into web development with python CGI scripts followed rather briskly by (among other things) some coding for Android and using Google App Engine.

The gradient of the learning curve seems to curve upwards towards the end, but I think this is a good general introduction to Python assuming no prior knowledge of Python itself, but a little knowledge of another programming language. It is meant to be worked through, hands-on, and there is a unifying example problem throughout the book of data about a group of athletes in training. Some readers will find the style of the “Head First” books annoying: others may find that the attempt to present material in a way that best suits your brain really works.

---

## **Building Android Apps with HTML, CSS and JavaScript**

**Jonathan Stark**

**O'Reilly Media**

**ISBN: 978-1-4493-8326-8**

**184pp.**

**£ 22.99**

**Published: September 2010**

**reviewed by John Collins**

I don't think the title of this book really describes its subject matter very accurately, but not so as to detract from the book. It talks about building web pages and forms for optimum use on web-enabled phones such as Android and "dressing them up" to look like Apps on the phone. It then describes how to use some open-source tools to convert those to "native" apps for Android and possibly other phones.

After a quick introduction to HTML, CSS and JavaScript (I don't really see the point of this as the book subsequently uses far more than is explained here – also the book has lumps of XML in it with little explanation), the book spends two chapters on styling issues, one on animation, one on client-side data storage and one on offline working. The book concludes with a chapter on converting web-based apps to "Native" apps and one on submitting apps thus created to the Android market.

The text doesn't go into much detail of the "nitty-gritty" of JavaScript in particular, relying heavily on the use of two open-source JavaScript libraries JQuery and JQTouch to provide a lot of the functionality rather than going into the details of explaining how those provide the functionality or how you could do it without them. I'm sure this is ample for many people, but readers will probably need to refer to other books if they need functionality not provided in those libraries.

The conversion to Native apps relies on using the PhoneGap software, (<http://www.phonegap.com>). This software allows apps to be constructed for other phones as well as Android (although the relevant SDKs will be required – sometimes at a charge).

I think this book will make a helpful introduction for people wanting to optimise web pages for display on various phones and go on to build apps for phones – and not just Android ones. To build comprehensive Apps just for Android, the user will have to get to grips with Java which this book does not deal with.

---

## **Pragmatic Guide to Git**

**Travis Swicegood**

**Pragmatic Bookshelf**

**ISBN: 978-1-934356-72-2**

**160pp.**

**£ 18.99**

**Published: November 2010**

**reviewed by John Collins**

This book give an overview of Git, one of the three popular source code control systems (the others being Subversion and Mercurial).

I already use Git all the time and little of what is talked about here was new to me. I don't really like the style of presenting a list of "recipes" with little variation for doing tasks considered important by the author. It would be much better, in my opinion, to list the various git commands with their options and follow those with common examples and a little more guidance about doing slightly different tasks. Some of the parts about installing Git on various computers could usefully be omitted.

---

I really don't think the book helps enough with the most complicated aspects, such as merging conflicting commits. It doesn't talk about "hooks" either, which I use all the time.

This book is neither a reference nor a pocket guide (especially as it's too big to fit in a pocket) and I have to say I don't think it's particularly useful.

The best introduction is probably Progit by Scott Chacon at <http://progit.org/book/> but beware of some elementary omissions.

---

## **Pragmatic Guide to JavaScript**

**Christophe Porteneuve**

**Pragmatic Bookshelf**

**ISBN: 978-1-934356-67-8**

**150pp.**

**£ 18.99**

**Published: November 2010**

**reviewed by John Collins**

This book describes how to do what the author considers to be common tasks in JavaScript, the scripting language which is supported in most browsers to do certain operations on the client side. Despite its name, JavaScript has only a superficial resemblance to Java.

I have to say that I think this book is too brief to be anything much more than a list of "recipes" for doing the tasks which the author considers most important and you may be in difficulty if you want to do slightly different things and only have this book to guide you. There really isn't enough to explain the interaction of JavaScript with different browsers, with the DOM, with CSS, with components of forms, with various windows and frames, and the nature of JavaScript objects so that the reader can properly understand what's going on.

It might be handy as a starting point for the reader to begin a project but to develop anything to more than a superficial level the reader would in my opinion need to refer to one of the more comprehensive volumes such as "JavaScript: The Definitive Guide" by Flanagan, published by O'Reilly. It is much more of a "Tome" but its reference section with members of objects and functions is invaluable. Another popular book is "Object Oriented Javascript" by Stoyan Stevanov.

---

## **Pragmatic Guide to Subversion**

**Mike Mason**

**ISBN: 978-1-934356-61-6**

**150pp.**

**£ 18.99**

**Published: November 2010**

**reviewed by John Collins**

This book describes how to do certain common operations in Subversion, one of the three popular source code control systems (the others being Git and Mercurial). It should not be confused with the other book "Pragmatic Version Control, Using Subversion" 2nd Edition by the same author.

I read through this not having used Subversion or being familiar with it. I have to say I don't really feel that familiar with it afterwards. I just feel I know how to use a few commands.

I think this book lacks a proper explanation of the principles behind Subversion and why the various commands operate the way they do, so the reader can get a handle on what is going on. It just seems

---

to give a list of “recipes” for doing tasks which the author considers most important. I think the reader will be a little at sea trying to do slightly different tasks.

Examples are given using Internet Explorer and the “Tortoise” utility, which is not widely available.

I don’t think this book is either good as a reference or as a “pocket guide”, especially as it’s too big to fit in a pocket.

---

## Contributors

**John Collins** is a former UKUUG Council member and runs Xi Software Ltd which specialises in System Management tools for UNIX and Linux systems. Xi Software is based in Welwyn Garden City and has been in existence for about 25 years.

**Mike Fowler** is a Senior Software Engineer at Dotted Eyes. He is a firm open-source advocate and has helped companies move from Microsoft based solutions to complete Linux/PostgreSQL stacks. He is a major contributor to YAWL, an open source BPM/Workflow system. When not melded with a keyboard, he spends his time playing as many different stringed instruments as he can.

**Martin Houston** has been using Unix since 1980 starting with a PDP11/70 and Linux since 1994. He was the founder of the original UKUUG Linux SIG and is currently a freelance Linux consultant. He lives in Essex with wife Teresa and 2 cats, 2 dogs and too many computers.

**Jane Morrison** is Company Secretary and Administrator for UKUUG, and manages the UKUUG office at the Manor House in Buntingford. She has been involved with UKUUG administration since 1987. In addition to UKUUG, Jane is Company Secretary for a trade association (Fibreoptic Industry Association) that she also runs from the Manor House office.

**John Pinner** is a member of UKUUG Concil and has been using Unix since 1978, and Linux since kernel 0.95. Through Thyme Software Ltd he specialises in producing bespoke systems, for both business and engineering applications, all using Free Software, and runs the Linux Emporium originally started by UKUUG member John Winters. His main development tools are C, Python, make and vim. John is active in the FLOSS community, is an elected member of the Python Software Foundadtion, and received the 2010 PSF Community Service Award.

**Mike Smith** works in the Chief Technology Office of a major European listed outsourcing company, setting technical strategy and working with hardware and software vendors to bring innovative solutions to its clients. He has 20 years experience in the industry, including mid-range technical support roles and has experience with AIX, Dynix/ptx, HP-UX, Irix, Reliant UNIX, Solaris and of course Linux.

**Sam Smith** runs Opentech with Dave and Anna, and does a load of other things as well. He’s on twitter as @smithsam and opentech can be found @opentechuk.

**Howard Thomson** writes: From being the Post-Graduate System Manager of a Unix V6 system at Westfield College London in the 1970s, including much kernel hacking and tweaking, through Unix and C code development and maintenance in a small business environment, to my current Free Software efforts, I have always preferred a Unix based development process. I am currently developing my own Eiffel IDE, and contributing to the Bacula backup project.

**Paul Waring** is chairman of UKUUG and a director of a wholesale insurance broker. Outside of work he can usually be found filing documentation bugs against various open source and free software projects.

**Roger Whittaker** works for Novell Technical Services at Bracknell supporting major Linux accounts in the UK. He is also the UKUUG Newsletter Editor, and co-author of three successive versions of a SUSE book published by Wiley.

---

## Contacts

Paul Waring  
UKUUG Chairman  
Manchester

Phil Hands  
Council member  
London

Kimball Johnson  
Council member  
Lancaster

Holger Kraus  
Council member  
Leicester

Niall Mansfield  
Council member  
Cambridge

John Pinner  
Council member  
Sutton Coldfield

Howard Thomson  
Treasurer; Council member  
Ashford, Middlesex

Jane Morrison  
UKUUG Secretariat  
PO Box 37  
Buntingford  
Herts  
SG9 9UQ  
Tel: 01763 273475  
Fax: 01763 273255  
[office@ukuug.org](mailto:office@ukuug.org)

Roger Whittaker  
Newsletter Editor  
London

Alain Williams  
UKUUG System Administrator  
Watford

Sam Smith  
Events and Website  
Manchester