



Packt>

Template:
MASTERING

USE CASE: IMPROVE

Once you've implemented a technology and learnt all of the associated tasks; how do you make it **faster, better, more efficient**? An advanced, 'expert' level book!

Not just how it works, but **why** it works the way it does. This *Mastering* book needs to include:

- › Examples
- › Best practice
- › Detailed explanations
- › Expert techniques

The *Mastering* series follows on from the *Hands-On series*.

An example:

Hands-On Machine Learning with R would enable the reader to build a scalable machine learning model with Big Data using R. It would show the user how to work with a machine learning algorithm and use it to build a Machine Learning model from raw data.

Mastering Machine Learning with R would allow a reader to take machine learning to an **advanced level** – that might include building and deploying predictive models. It might require the introduction of additional technologies, such as Apache Hadoop, Hive, Pig and Spark.

Features of the *Mastering Series*



Covers **high performance techniques**: for example, maximize the performance of code and applications.



Teaches **how to solve performance issues**, and focuses on techniques and tools that result in applications that perform **faster**



Focus on **niche areas** or **integrating new tools**

The *Mastering* series is pitched at the advanced developer:

Mastering Reader Profile

Knowledge Level	Book Objective	Starting Point	End Point
Advanced	Covers niche areas or specialised applications	Professional proficiency Good working knowledge of tech	<i>A power user</i> <i>Refinement and mastery</i> <i>Focus on niche areas</i>

Remember to think about the following:

› **Prerequisites:** knowledge that the customer will need to have in order to use the book.

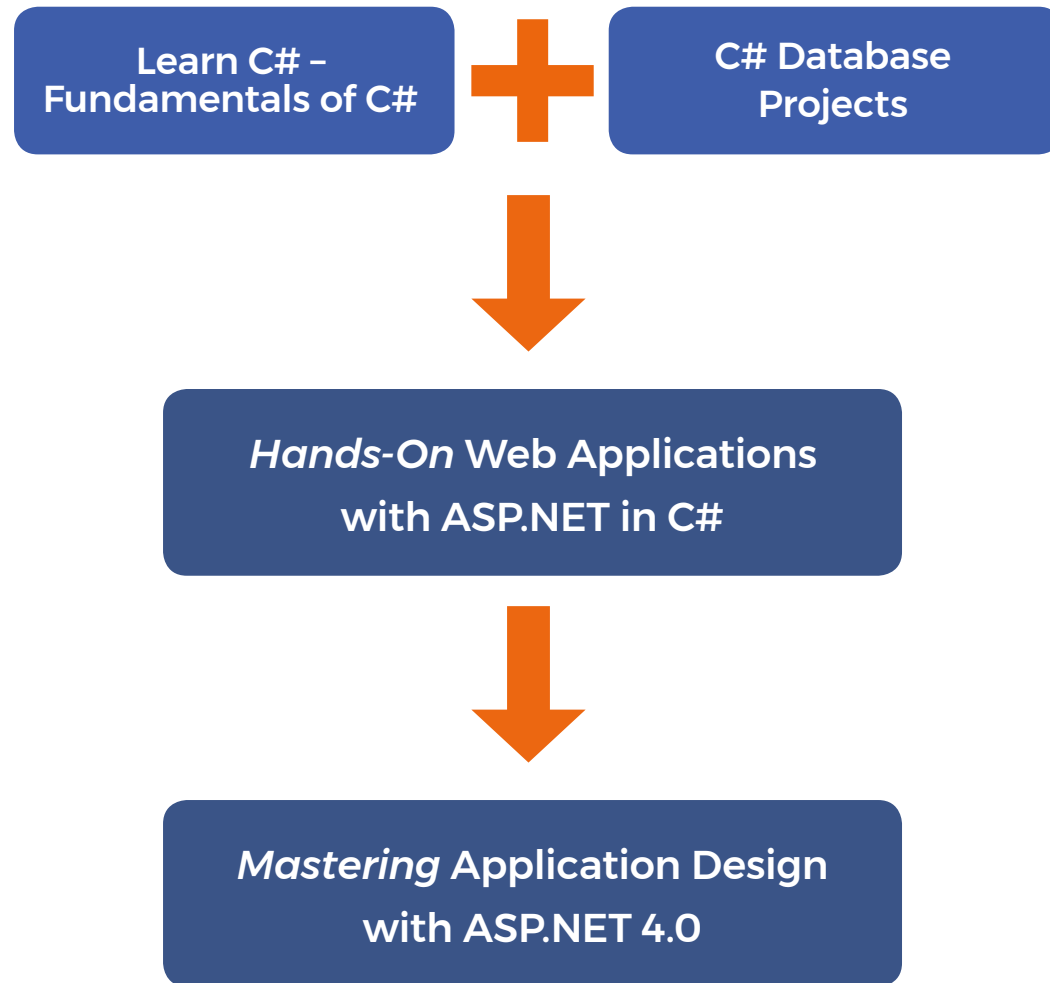
This needs to be **explicit** in the product's promise. For example: *Mastering Data Mining with R* requires knowledge of both R **and** of data mining. The customer should have read *Hands-On Data Mining with R*.

It is important to be explicit in the product information (i.e. description) about the prerequisite knowledge required by the customer.

› **Learning outcome:** the knowledge the customer will have and the level they will be at once they've completed your book.

On finishing the book, the customer will be a **power user** who is able to meet their own requirements. They'll also then be a **source of expertise** to other users.

For example: A customer learning path/product map might be as follows:



Content

Your *Mastering* book builds on the *Hands-On* title. If you haven't seen an example of the appropriate *Hands-On* title, then please ask your editor for a copy. This will really make sure you're aware of the concepts that have (and haven't) been covered previously.

The *Mastering* series explores **specific applications** and is **more granular** than *Hands-On*. It might address the area of sub-topics or scenarios where more than one technology is required. It might be about a more specific topic inside the main area.

This series is concerned with **implementation** in a real-world context, and with the *methodology* needed for this implementation. In addition to this, *Mastering* books **drill down to a deep level, focus on niche areas** and provide the reader with **mastery of a concept**.

- **Page length:** 350-550 pages
- **Builds on the fundamentals** of a concept (e.g. Machine Learning) that the reader already understands giving them greater depth of knowledge
- Includes key **troubleshooting** help

Think of each chapter like a **lesson plan** – it should relate to the chapter previously, but should also **sit entirely independently** of it. It needs to facilitate micro-learning. Not many readers will go through the book from cover to cover – they will dip in

and out looking for the content that is relevant to them. For this reason, Packt books have a modular structure that lets the reader **pick and choose the content** they need, when they need it. This is a book that will be used ‘on the job’.

Introduction

In the introduction you need to establish the following:

1. The problem that your book exists to solve

Give the reader a clear idea of what they’ll learn and achieve by reading your book. Describe the skills they will gain and the things they will build.

2. Introduction to the topic

Familiarize the audience with the big picture and major concepts that you’ll explore

3. Inform the reader about **the things that they need to know before they start**. Remind the reader of the things that they should already know and then show them where you are going to take them.

a. Prerequisite knowledge: spell out what knowledge you are assuming in relation to the technology that you are using.

b. The learning outcome: tell the reader what they will know and be able to do at the end of the book. Remember, by this stage they will be **an expert** and **source of expertise** to others.

4. Any **installation guidance and information** they need to get set up.

5. Summary

Bring together everything you’ve mentioned in the introduction, and summarize it in around 300 words.

Book Structure

Chapter 1

Remind the reader of any underlying core concepts that would have been covered in the *Hands-On* title. This is a **refresher** that will gather the readers and ensure that they are building on the same foundation – it should be **‘fast-track information’** to bring the user up-to-speed.

Middle Chapters

There are a number of ways in which you could order the content:

1. **Increase in complexity** as you progress through the book – this gives the reader the feeling that they are getting better and better as they progress. Go from simple to advanced.
2. **Go from common to rare**, put the most useful information at the start of the book.
3. **Go from early to late**, put topics in the order of their natural progression.

If there isn't an obvious order to the content, then do what is logical to you.

If appropriate, have a **‘bringing it together’** chapter towards the end. You might like to build an app that draws on knowledge from previous chapters.

The Last Chapter

Finally, **tell the reader what they now know** – and **congratulate them** on finishing your book!

Suggest the **next steps** that they might want to take in order to move onto the next level. **Your Editor is here to help**, so be sure to discuss this with them – we've got plenty of helpful books to suggest.



Chapter Structure

06

Chapter Header: Main topic – include SEO keyword(s), if possible

- Introduction
- Learning Outcomes

Technical Requirements

- List of Technologies, installations required
- Include GitHub URL for code bundle.

Topic A

- Explanation and overview

Subtopic A

- Explanation/essential concepts
- Examples, code, illustrations: explain complex concepts in clear, simple language
- Advice/Guidance/Pain points: Address common pain points and areas of confusion.

Subtopic B

- Explanation/essential concepts
- Examples, code, illustrations: explain complex concepts in clear, simple language
- Advice/Guidance/Pain points: Address common pain points and areas of confusion.

Subtopic...

Topic B

- Explanation and overview

Subtopic A

- Explanation/essential concepts
- Examples, code, illustrations: explain complex concepts in clear, simple language
- Advice/Guidance/Pain points: Address common pain points and areas of confusion.

Subtopic B

- Explanation/essential concepts
- Examples, code, illustrations: explain complex concepts in clear, simple language
- Advice/Guidance/Pain points: Address common pain points and areas of confusion.

Subtopic...

Summary

Q&A

Further Reading

Opening Pages

The opening page of a chapter should start on a right-hand page, and include:

1. A **chapter number**
2. The **chapter header**: Clear chapter titles that explicitly state the topic of the chapter. Include an **SEO keyword**, because your book's **table of contents** (and therefore the chapter headings) are important for making sure your book can be **found easily on Amazon, Google** and other sites.
3. An **introduction** (50–100 words) explaining the content of the chapter. Start with a couple of sentences summing up the goals of the chapter, relating it to the overall thread of the book (where we've come from, where we're going).

Be careful here – remember that the chapter needs to sit independently of the others. Don't assume the reader has read the book from start to finish. They haven't – they'll dip in and out as they need to!

4. A **bulleted list of learning outcomes** – these will correspond with the main H1 headings in the chapter (this list should include **page numbers** for ease of reference)
5. A **bulleted list of clear technical requirements**: list of technologies, installations required. Provide the Git Repo for the code bundle for the chapter.



Topics and Subtopics

Sections: Each chapter should have around 3–7 main sections, each introduced with a heading – this is labeled **Heading 1** in Type Cloud.

Packt’s research and experience over many years has highlighted the **importance of skim reading** to readers. For this reason, sections shouldn’t be too long. If the subject matter requires a very lengthy section, then make sure that it’s **broken down into logical sub-sections** (using **Heading 2**, **Heading 3** etc., in Type Cloud)

In a *Mastering* title there are four main kinds of content:

- 1. Explanation and Overview:** Describes how the content contributes to the overall goal of the book, and to the reader’s objectives. Informs the reader *why* they need to know something, and *how* it will fit into what they want to do. How will it be *useful*?
- 2. Essential Concepts:** Provide essential theory (what things are and how they work) and tell the reader how it is used (provide some instructions on how to go about doing something). Remember that the reader of a *Mastering* title already has a foundational understanding of both the tool - for example, Python - and the concept - for example, machine learning. There is no need to go over that.

3. Explanation: Using case studies, examples, code and illustrations, explain the concept previously described. Take the reader through it, helping the reader understand what they see or what happened.

4. Pain points: Provide advice and guidance by giving the reader the benefit of your experience and expertise. Packt has **tips** and **warnings** icons that you can use to highlight anything particular.

Aim for an even balance of the above. You might want to use the above in the order that it has been presented. Or feel free to mix it up!

Once you’ve done this, then move on to the next topic/subtopic.



Definitions

Chapter Summaries

The **chapter summary** captures the key points and reinforces the structure. You need to remind the reader of what they have just learnt. Aim for a maximum of 200–300 words.

Q&A

End each chapter with 7–10 questions. Questions are a way of letting your reader test how well they've learned the material. They are also a useful learning aid, helping the reader to reflect on and think about what they've read.

1. Start with factual and **comprehension-based questions**.
2. Include some **application-based questions** – when, how, why, where.
3. End with some **'challenge'** questions – questions that are not directly answered in the text and require the reader to try things out.

These might be practical challenges: how would you write code to solve problem x?

Remember to provide answers to the questions in the section at the back of the book called **Assessments**.

Further Reading

You should include additional references to useful third-party resources, or other Packt books that might help explain a particular concept in further detail in this section.

If you've assumed knowledge of the reader that would be covered by Packt's *Learn* or *Hands-On* titles, then remember to mention these books here.

For example: 'For more information on the principles of compressing data via dimensionality reduction, please refer to *Hands-On Machine Learning with Python*.'



Want to get in touch with Packt?

Want to get in touch and let us know your thoughts, feedback or anything else which springs to mind? We'd love to hear from you – drop as a line at authorsupport@packt.com

Useful Contact Details

- Author Support email: authorsupport@packt.com
- Packt Author Twitter handle: [@packtauthors](https://twitter.com/packtauthors)
- Packt Live email: packtlive@packt.com
- [Packt Experts Facebook Author Community](#)

Packt

