Packt

Template: HANDS-ON

Overview

USE CASE: APPLY TECHNOLOGY IN A REAL-WORLD SCENARIO

The objective of the *Hands-On* series is to help **established professionals** and **tech powered pros** put their knowledge to work, and figure out how to apply a core technology to a specific concept.

Hands-On is for developers working with a particular technology who want to focus on its practical implementation and associated methodologies. It is application based – for example: the application of Python to Machine Learning.

Hands-On series example: *Python Machine Learning Second Edition (9781787125933)*

Features of the *Hands-On* Series

- Aimed at those who already use a particular tool but want information on **how to apply it in a specific realworld situation**. For example; a Python programmer who wants to use Python to do Machine Learning
- **Application based:** comprehensive coverage of a concept
- Addresses **specific applications** and sophisticated, deep level questions
- Delivers **proven techniques** and skills and unlocks deeper insight
- Includes **working examples** and **case studies**

Overview

The *Hands-On* series is pitched at the intermediate level:

Hands-On Reader Profile

Knowledge Level	Objective	Starting Point	End Point
Working Knowledge	Covers the fundamentals of a concept; for example, web applications.	Understand fundamental principles of a language but not the concept being addressed. Have working proficiency with underlying tech	*A professional standard.* *Able to use the technology for their job*

Audience

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, *Hands-On Data Mining with R* requires the reader to have knowledge of R, but assumes that have little knowledge of data mining concepts and processes.

  The *Hands-On* series follows on from the ***Learn*** series, so you should assume that the reader understands the fundamentals of an underlying technology.

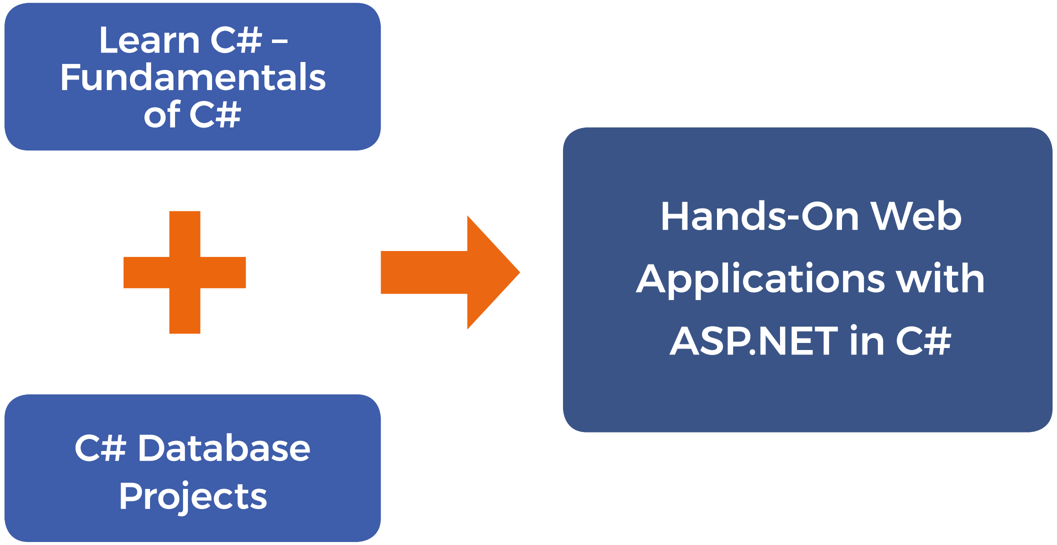
- **Learning outcome:** the knowledge/level the customer will be at having completed the book.

  The customer should finish the book and be capable of using the technology in a real-world situation and able to adapt it to meet their own requirements.

  In the example, *Hands-On Web Applications with ASP.NET in C#*, we clearly have an **implementation** scenario. The prerequisite to this title might be reading the two books *Learn C# – Fundamentals of C#* and *C# Database Projects* but at this stage, there should be an understanding of both.

Overview

The product roadmap might look like this:



Learn C# – Fundamentals of C#

+

C# Database Projects

Hands-On Web Applications with ASP.NET in C#

The learning outcome is that the reader is **capable of doing web applications with ASP.NET in C#**. They should be able to customize the technology to fit their own requirements.

Content

The *Hands-On* series explores specific applications. It might address the area of sub-topics or scenarios where more than one technology is required. It's concerned with **implementation** in a real-world context and with the **methodology** needed for this implementation.

- **Page length:** 350-550 pages
- **Builds on the fundamentals** of a technology (e.g. Python) that the reader already understands giving them greater depth of knowledge as they cover the concept (e.g. Machine Learning)
- A masterclass on **how to apply a technology to a concept**
- Includes key **troubleshooting** help
- Provides **help and guidance**
- Focus is on **practical** application

Book Structure

Think of each chapter like a **lesson plan** – it should relate to the previous chapter, 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.

A ***Hands-On*** book is one 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. Describe the rationale behind your topic inclusion and the sequence that they are presented in. This should:

   a. Spell out context: General concepts of the technology or topic

   b. Drill down to the area that you're going to be focusing on. This might also include an explanation of terminology, for example.

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're assuming the reader has

   b. The learning outcome: tell the reader what they will know at the end of the book

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

05

Chapter 1

Introduce the **basic methods**, and remind your reader of any underlying core concepts. This is a refresher that will gather the readers and make sure that they're building on the same foundation.

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 that your reader might enjoy next!

Chapter Structure

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

Definitions

07

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.

Definitions 08

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 *Hands-On* 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 *Hands-On* title already has a foundational understanding of a tool – for example, Python. There is no need to provide any information about that. The **concept** in question – Machine Learning – requires explanation.
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

09

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.

Provide answers to the questions in a section at the back of the book called **Assessments**.

Further Reading

Additional references to useful third-party resources, or other Packt books that might help explain a particular concept in further detail. 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**
- Packt Live email: **packtlive@packt.com**
- **Packt Experts Facebook Author Community**