

Sample pages from the  
Professional IT Foundation in  
simple terms book

Sandor A. Nagy

# Professional IT Foundation

---

in simple terms



Published by: solidCode Ltd.  
itfoundation@solidcode.eu

[www.solidcode.eu](http://www.solidcode.eu)

2021

First edition

ISBN 978-615-81989-0-5

Book cover and graphics design: Petra Fias

Proofreading: Aleksandra Konsek-Nagy

Technical review: Zoltan Kerekes

Testing section review: Andrea Kis

Printed by: DePress Nyomdaipari Ltd.

Copyright ©2021 Sandor A. Nagy

All rights reserved!

No part of this book may be reproduced, stored in a retrieval system, or transmitted by any means, electronic, mechanical, photocopying, recording, or otherwise, without written permission from the copyright holder.

# Table of Contents

## **FOUNDATION**

Computing basics	10
Platform	18
Languages	21

## **NETWORKS**

Network building blocks	26
IT infrastructure	29
Telecommunication	37
Web	40

## **ARCHITECTURE**

Overview	47
Building a software solution	59
Roles	68
Data Layer	72

## **WHAT IS THE CLOUD?**

Cloud	78
Cloud services	83

## **TRENDS**

Productivity	90
Technology	100
Current websites	105

## **AGILE**

Software Development Life Cycle	111
Practices	118

## **TESTING**

Requirements	126
Process optimisation	128
Testing jobs	145

## **PEOPLE**

Team	149
User experience	154
IT management skills	163
Hiring	165

QUIZ SOLUTIONS	173
----------------	-----

FIGURES	174
---------	-----

BIBLIOGRAPHY	178
--------------	-----

GLOSSARY	179
----------	-----

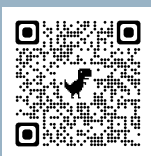
## About the author

Sandor A. Nagy has 20 years of experience in the software industry in many roles. His company is offering IT training and software development services [www.solidcode.eu](http://www.solidcode.eu).

Sandor started as a software developer working in various companies ranging from family business to the largest investment bank. He was a technical team lead on a financial trading platform. Sandor has fulfilled different hiring manager positions as well. He has worked with a wide variety of technologies during his career. Sandor has many Microsoft Certificates, and he is also a Certified Scrum Master. He has performed hundreds of technical interviews as a hiring manager. Sandor has built several software engineering teams as a leader. He is known as an approachable mentor who helped people to progress in their career. As a Head of Development and Development Centre Manager, he motivated many colleagues to widen their knowledge.

Sandor, together with Peter Gombas, translated "Introduction to PC hardware and troubleshooting" to Hungarian (*Michael Meyers, McGraw-Hill/Osborne., 2003*).

For fun, Sandor enjoys a motorbike ride on a good curvy road.



Sign up for the solidCode Academy YouTube channel for interesting technology videos. You can find introduction to Machine Learning, Software development process and many more.

# Preface

During my 20 years in the software industry, I saw that a knowledge gap is causing problems. Professionals without IT background have difficulties to understand software developers. It is hard for a newcomer to understand technology and software development processes. This results in daily frustration, misunderstandings and even project delays. I created an IT training material and wrote this book to fill the knowledge gap.

Once you gain IT knowledge, it brings an additional advantage. A special connection is formed, when people have things in common, share interests and speak the same technical language. This book will help you to understand software development better. It will provide you with a strong foundation to be successful in IT and connect with technical people. In this book, I will share a personal story about this special connection from the time I was working in the UK.

The last aspect I would like to mention about software development knowledge is authority. When IT professionals know that you understand their craft, they will trust you more. A strong foundation will be beneficial when doing business with a software company. Otherwise, how will you verify that you are getting the right solution?

Many participants of my training have several years of experience in the IT industry. They were surprised how much they did not know, and they also got motivated to learn more.

Enjoy learning IT with me! Please reach out if you have any comments or suggestions about the content of this book.

*Sandor*

## Testimonials



*Livia – Senior HRBP:* It was a great experience and lots of fun to participate in the solidCode training with Sandor. After so many years spent in IT industry, I thought I have a good understanding of the new trends and software development practices, and I speak developers' language, but it seems Sandor could teach me a lot more. Sandor was well prepared with customized training material, interactive games, regular checkpoints, summaries, visual tools, video materials and IT dictionary. The whole course is structured, easy to follow and very interesting. It has been delivered in a comprehensive way and a simple language. I would encourage everyone, especially HR people, to attend it, if they are eager to learn more about IT foundations, IT jobs, new agile trends, and development practices. Thank you, Sandor!



*Anna - Scrum Master / Agile consultant:* The training provided me with a map and a dictionary that helped me to place concepts that I had heard in everyday life but did not know how to relate to. It gave me a sense of security that Sandor guided us with great experience and patience and created an atmosphere where I dared to ask questions and to acknowledge what I did not know.

# Instructions for the book

Here are the different symbols you will see reading this book. The description of each symbol is included in each frame below.



This is an example of external links. The links point to recorded demos matching the topics in this book, or recommended videos and webpages. This specific QR code will take you to solidCode Academy's YouTube channel sign up.



This symbol represents information that is important to pay attention to.

QUIZ

This symbol contains Quiz questions. There are multiple choice quiz questions. If not indicated otherwise, only a single answer is correct. The quiz questions will suggest when there are multiple correct answers. There are statements where you need to decide whether they are True or False. The solutions can be found at the end of this book.



This symbol represents interesting information, or stories in connection with a given topic.



Removed content

readable. The machine code is machine readable, so it can execute the instructions. People writing source code using programming languages are the software developers (Devs), or software engineers.

This is how JavaScript source code looks like in a text editor:

```
src > JS index.js > ...
1  import React from "react";
2  import { render } from "react-dom";
3  import { BrowserRouter as Router } from "react-router-dom";
4  import "bootstrap/dist/css/bootstrap.min.css";
5  import App from "../components/App";
6  import "../index.css";
7  import configureStore from "../redux/configureStore";
8  import { Provider as ReduxProvider } from "react-redux";
9
10 const store = configureStore();
11
12 render(
13   <ReduxProvider store={store}>
14     <Router>
15       <App />
16     </Router>
17   </ReduxProvider>,

```

Figure 2: Source code

This is how machine code looks like when viewed in a text editor:

[illegible]

Figure 3: Machine code

Removed content

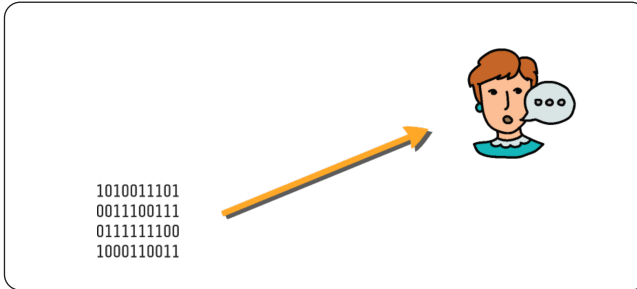


Figure 9: Low- and high-level languages

## QUIZ

**1. Markup Language: describes building blocks in a file and a software can construct desired output from the input file.**

- ☐ True
- ☐ False

**2. What is the focus of an object-oriented language?**

- ☐ A: Modelling real life objects in our code.
- ☐ B: Using “functions” to build software.

**3. A high-level language is human readable.**

- ☐ True
- ☐ False

There are over 600 programming languages that can be categorized in various ways. *Figure 10* shows a simplified table to give you an idea about some of the language groupings with a few specific examples. Note that PHP appears twice in the table. PHP 5 is used in scripting; PHP 8.0 is used in a managed environment. (*Wikipedia, 2021*)

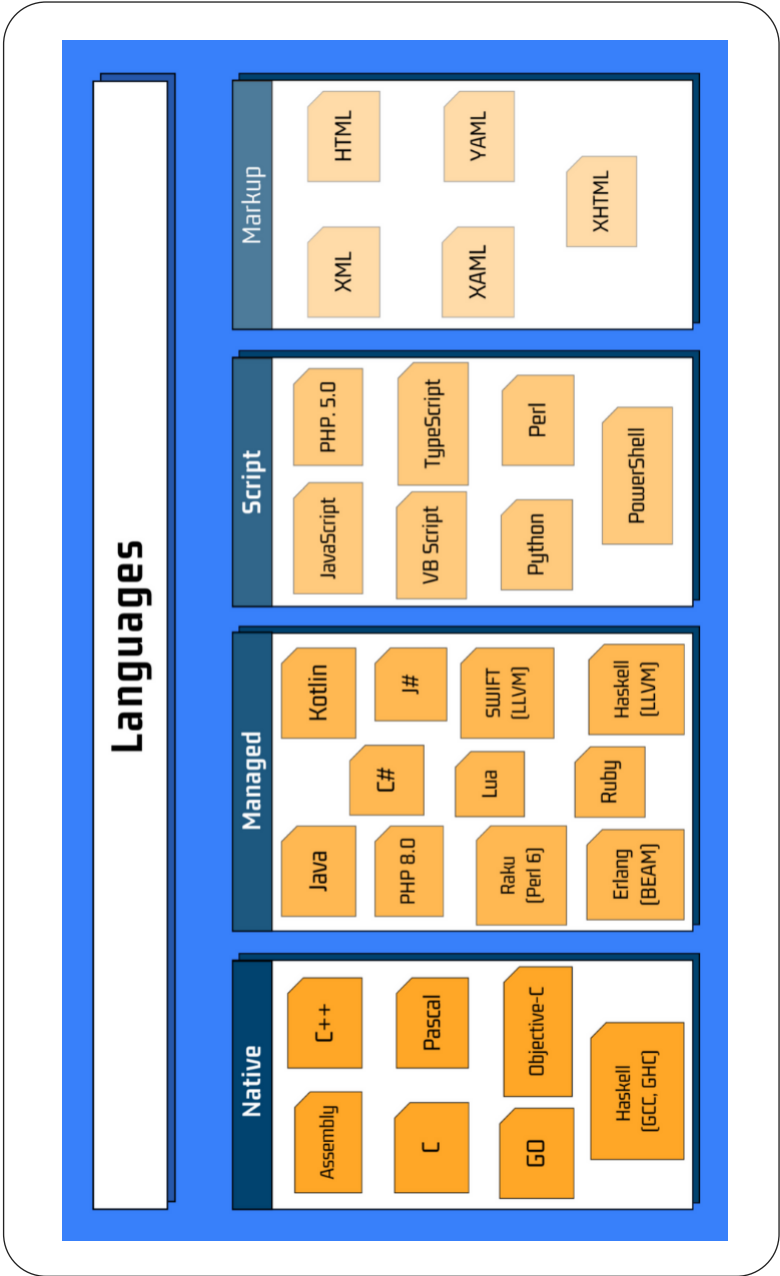
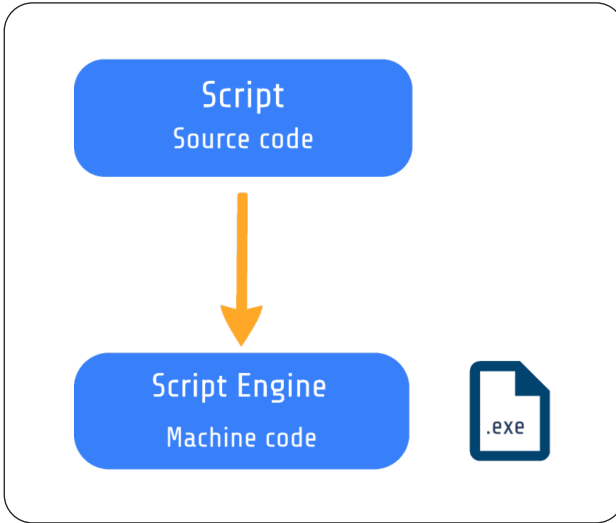


Figure 10: Programming language examples

Removed content



*Figure 26: Scripting*



Use this QR code link to do the scripting exercise. You will practice writing some basic JavaScript code and execute it in the browser.

Removed content



# QUIZ 18

**1. Full Stack JavaScript development can be performed using these (select three):**

- ☐ A: *NodeJS*
- ☐ B: *C++ / CLI*
- ☐ C: *Browser*
- ☐ D: *Web development environment*

**2. 'Time to Market' is (select three):**

- ☐ A: *Good if it is short*
- ☐ B: *Time that it takes from an idea to making the software available to the user*
- ☐ C: *The longer the better*
- ☐ D: *An indication of the competitiveness of a software company*

**3. Automation has increased productivity in these areas (select three):**

- ☐ A: *Testing*
- ☐ B: *DevOps*
- ☐ C: *Development tools*
- ☐ D: *Code review*

**4. Disadvantages of "cross-platform" development (select three):**

- ☐ A: *Build once and use it on multiple platforms*
- ☐ B: *Missing Native features*
- ☐ C: *Needs a development team per platform*
- ☐ D: *App is usually slower than native app*

## Technology

In order to reduce the Time-To-Market, development tools, team practices, and a lot of automation has been created. Additionally, quite a few productivity frameworks have emerged. We have seen cross-platform development, and full-stack JavaScript development so far. These allow software developers, teams, and companies to be more productive.

## Simplification

Now let's check some of the approaches to simplify software creation and enable a larger audience to build software applications.

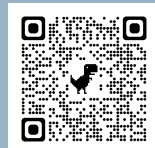
### Low-Code development

These tools will allow a person to build software without being a software developer. Some of the Low-Code development frameworks are Microsoft Power Apps and Google App maker.



For more information on low-code platforms please check this wiki page.

For a list of available platforms please visit this website.



### Websites building

In this category of low-code development platforms we can only build websites. However, these tools do a great job at making website building really easy. Website building tools are quite

capable of providing all the features you need to create a web shop and start selling online. WordPress, WIX, Joomla are examples of tools in this category.



If your goal is selling online, you might consider some of the SaaS options that offer web-shop features with zero coding.

### **Robotic Process Automation**

**Robotic Process Automation (RPA)** allows automation of manual tasks. Even in the 21<sup>st</sup> century many office jobs include a lot of repetitive manual work. Copying, formatting transforming, extracting parts of information and inputting into another software is quite common. RPA tools provide an automation solution for this type of tasks. The manual actions can be recorded and replayed. RPA allows to perform a whole workflow of processes that involve using multiple software in an automated way. Examples of RPA tools: Microsoft Power Automate, Blue Prism.

### **Trendy technologies**

Let's get familiar with some of the technologies you probably hear about quite often. *Figure 75* summarises a some of trendy technology buzzwords.



Figure 75: Technology trends

Removed content

I was astonished. It was definitely not the normal recruitment experience. I felt connected and convinced he would get me a good job as he also understood my craft quite well.

There are perfectly good candidates that are rejected simply because the interviewers did not feel connected with them. I have a colleague who is a super talented developer and one of the smartest guys I know, however he has difficulties to get his real values across on interviews.

As a Development Centre Manager, I have tried several techniques to attract candidates while building up a development centre from scratch. Some worked very well for us. Surprisingly many of these did not require a massive budget. Working with a single recruiter, we hired 3 people per month on average until we reached 137.

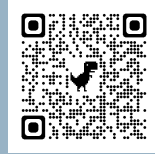


Check out the secrets from the science of persuasion video on the QR coded link. The video is based on Dr Robert Cialdini's book called Influence.

Congratulations! You have completed the IT foundation self-study material. Well done!

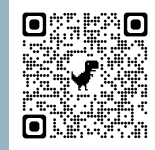
If you would like to participate in a live training or you would like me to speak at an event, please reach out to me using our website: [www.solidcode.eu](http://www.solidcode.eu), or write to [itfoundation@solidcode.eu](mailto:itfoundation@solidcode.eu)

If you would like to get the Professional IT Foundation certificate from solidCode Academy, you can get the process started here.



In case you would like to keep up to date with IT and further improve your skills, sign up to our YouTube channel.

Lastly, let me invite you to our LinkedIn group called You can do IT!



Removed content



## Figures

Figure 1: Computer peripherals .....	2
Figure 2: Source code .....	2
Figure 3: Machine code .....	2
Figure 4: Native development .....	2
Figure 5: Scripting .....	2
Figure 6: Managed software development .....	2
Figure 7: Native development .....	2
Figure 8: Cross-platform development .....	2
Figure 9: Low- and high-level languages .....	2
Figure 10: Programming language examples .....	2
Figure 11: Computer network .....	2
Figure 12: Local Area Network (LAN) .....	2
Figure 13: Wide Area Network (WAN) .....	2
Figure 14: Physical computer-based software solution .....	2
Figure 15: VMs running on a physical host computer .....	2
Figure 16: Containers running on a host computer .....	2
Figure 17: Virtualization summary .....	2
Figure 18: Network nodes “handshake” .....	2
Figure 19: OSI layers of protocols.....	2
Figure 20: User, Client, Server .....	2
Figure 21: Static webpage .....	2
Figure 22: Dynamic webpage process - part one .....	2
Figure 23: Dynamic webpage process - part two .....	2
Figure 24: Dynamic webpage process - part three.....	2
Figure 25: Dynamic webpage process - part four.....	2
Figure 26: Scripting.....	2
Figure 27: Peer-to-peer architectural style .....	2
Figure 28: Client-server architectural style .....	2
Figure 29: Tiny computers: Arduino Nano(left), ESP8266(right) ..	2
Figure 30: Client and server-side.....	2
Figure 31: Frontend and backend .....	2

Figure 32: Thick client .....	2
Figure 33: Less thick client .....	2
Figure 34: Thinner client .....	2
Figure 35: Thin client.....	2
Figure 36: Distributed computing .....	2
Figure 37: Microservices .....	2
Figure 38: Software solution .....	2
Figure 39: A software company's responsibilities .....	2
Figure 40: Production environment .....	2
Figure 41: Deployment.....	2
Figure 42: Hosting .....	2
Figure 43: Technologies .....	2
Figure 44: Mini web server.....	2
Figure 45: Infrastructure versus roles .....	2
Figure 46: Developer's roles.....	2
Figure 47: DevOps's roles .....	2
Figure 48: IT Sysadmin's roles .....	2
Figure 49: Data layer in the infrastructure.....	2
Figure 50: Data layer .....	2
Figure 51: DB jobs .....	2
Figure 52: Data warehouse .....	2
Figure 53: Data science .....	2
Figure 54: CLOUD .....	2
Figure 55: Cloud infrastructure .....	2
Figure 56: Cloud vs Company infrastructure.....	2
Figure 57: Public cloud .....	2
Figure 58: Private Cloud 1 .....	2
Figure 59: Private Cloud 2 .....	2
Figure 60: IaaS .....	2
Figure 61: Infrastructure as a Service.....	2
Figure 62: Platform as a Service .....	2
Figure 63: Platform as a Service .....	2
Figure 64: Software as a Service.....	2

Figure 65: Software as a Service..... 2

Figure 66: X as a Service ..... 2

Figure 67: Time to Market – part one ..... 2

Figure 68: Time to Market - part two ..... 2

Figure 69: Different platforms..... 2

Figure 70: Managed development ..... 2

Figure 71: User, Client, Server ..... 2

Figure 72:Client-Server architecture ..... 2

Figure 73: Web browsers hosting V8 JavaScript engine..... 2

Figure 74: Full-stack JavaScript development. .... 2

Figure 75: Technology trends ..... 2

Figure 76: Single Page Application ..... 2

Figure 77: Progressive Web App ..... 2

Figure 78: PWA offline..... 2

Figure 79: Typical website nowadays..... 2

Figure 80: Sections of a website ..... 2

Figure 81: From idea to realization ..... 2

Figure 82: From idea to creation actionable tasks. .... 2

Figure 83: From dev to end-user. .... 2

Figure 84: From coding to verification ..... 2

Figure 85: Software build ..... 2

Figure 86: Release ..... 2

Figure 87: Full SDLC process ..... 2

Figure 88: Continuous Integration ..... 2

Figure 89: Waterfall..... 2

Figure 90: Pair programming..... 2

Figure 91: Mob programming ..... 2

Figure 92: SCRUM framework ..... 2

Figure 93: Kanban board ..... 2

Figure 94: Functional requirements ..... 2

Figure 95: Shifting left ..... 2

Figure 96: Test pyramid - Mike Cohn ..... 2

Figure 97: Extended test pyramid ..... 2

Figure 98: White box testing .....	2
Figure 99: Black box testing (architecture is unknown) .....	2
Figure 100: Focus areas .....	2
Figure 101: AB testing .....	2
Figure 102: Test Driven Development (TDD).....	2
Figure 103: Behaviour Driven Development (BDD).....	2
Figure 104: Testing categories by place of testing .....	2
Figure 105: Functional testing.....	2
Figure 106: Non-Functional testing.....	2
Figure 107: Testing perspective from developer to user .....	2
Figure 108: Testing jobs .....	2
Figure 109: Testing jobs from a different perspective .....	2
Figure 110: Development team.....	2
Figure 111: Proximity exercise .....	2
Figure 112: Proximity and colour .....	2
Figure 113: Gutenberg diagram .....	2
Figure 114: Crowded Street .....	2
Figure 115: Savanna .....	2
Figure 116: Natural light.....	2
Figure 117: The more on the screen the harder to find.....	2
Figure 118: Less is more (better).....	2

# Bibliography

Lallemand, C., & Gronier, G. (2012). Enhancing User eXperience during waiting time in HCI: Contributions of cognitive psychology. (pp. 751-760). Newcastle (UK): Public Research Centre Henri Tudor.

Andrew, S. T. (1999). *Computer networks*. Prentice-Hall Inc.

Wikipedia. (2021). *List of programming languages*. Retrieved from Wikipedia:  
[https://en.wikipedia.org/wiki/List\\_of\\_programming\\_languages](https://en.wikipedia.org/wiki/List_of_programming_languages)

Wikipedia. (2021). *Computer Architectures*. Retrieved from Wikipedia:  
[https://en.wikipedia.org/wiki/Software\\_architecture](https://en.wikipedia.org/wiki/Software_architecture)

Wikipedia. (2021). *ETL based data warehousing*. Retrieved from Wikipedia:  
[https://en.wikipedia.org/wiki/Data\\_warehouse#ETL-based\\_data\\_warehousing](https://en.wikipedia.org/wiki/Data_warehouse#ETL-based_data_warehousing)

Microsoft. (2021). *Azure Admin Portal*. Retrieved from Azure Portal: <https://portal.azure.com/#home>

Wikipedia. (n.d.). *Low-level programming language*. Retrieved from Wikipedia: [https://en.wikipedia.org/wiki/Low-level\\_programming\\_language](https://en.wikipedia.org/wiki/Low-level_programming_language)

Wikipedia. (2021). *As a Service*. Retrieved from Wikipedia:  
[https://en.wikipedia.org/wiki/As\\_a\\_service](https://en.wikipedia.org/wiki/As_a_service)

ELTE Faculty of Informatics. (2021). *Data Science*. Retrieved from ELTE: <https://www.inf.elte.hu/en/content/data-science-in-computer-science-msc.t.1732?m=361>

Fowler, M. (n.d.). *Test Driven Development*. Retrieved from  
<https://martinfowler.com/>:  
<https://martinfowler.com/bliki/TestDrivenDevelopment.html>

## Glossary



Here is an online dictionary that contains all expressions used in this book. If you notice anything missing, please add the expression to the dictionary and I will provide the definition for it.