A Roadmap to Learning Python

Contents

Something about me 3
Where to start 4
What is expected of you 5
Programming 6
Bibliography 8

Rosalie de Klerk – Bambara
www.bambara.nl
Version 1.0
26-2-2023

Something about me

I have been enthusiastically involved with information and communication technology (ICT) since 1996. My passion is creating and designing digital applications. In 1998 I graduated in Multimedia Design from the Grafisch Lyceum Rotterdam, where I had immersed myself in website design, 3D drawing and animation and I also learnt to work with Adobe Photoshop 4.0. I look back on that time with nostalgia.

After leaving the Lyceum, I started my own business developing websites. It took ten years before I decided to go back to studying again. In 2014 I graduated with an applied sciences degree in Business Informatics and I also gained my teaching certificate (*tweedegraads bevoegdheid*) in Informatica as well, which allowed me to share my knowledge of ICT with others. I am now studying Education Sciences at the Dutch Open University.

As well as ICT and education, I also love drawing (both digital and analogue) and travelling. At the beginning of 2021 I decide to give up my job as a lecturer at the Rotterdam University of Applied Sciences, so that I could focus entirely on drawing, running my business and my studies. At the university of applied sciences, I taught the programming language Python and I was a career coach. The passion for sharing my knowledge and supporting students is as strong as ever, and it is precisely that that has led me to create this roadmap.

This roadmap is aimed at everyone, including children and their parents or carers, who is interested in learning to program in Python. It outlines the essential information I have always taught my students, like *how to start programming*, *what you have to do* and *what is expected of you as a budding programmer*.

Let’s get started...

Where to start

"*Miss, the book is enormous. Can't you just show us? Then we can watch and then we'll learn it much faster. Really*." This is one of the remarks that students often make. Experience and various research, however, including Dunlosky (Dunlosky, Rawson, Marsh, Nathan, & Willingham, 2013), have shown that watching passively while a teacher writes a piece of code does not lead to knowledge. If students do not have any experience of devising code, they will not be able to pass the test. This is why you won't find solutions for homework here, but a roadmap explaining how you can learn to program in Python.

"How then? What is the best way to learn how to program?"

Learning strategy

According to Dunlosky, there are various learning strategies that vary in their effectiveness, but how do you know which strategy is best for you? One of the strategies discussed is 'elaboration', where the student continually asks him or herself questions about the subject matter, endeavours to make connections and checks what they think they know. The questions that you ask yourself when 'elaborating' depend on the subjects that you are studying. In the case of programming Python, they would be questions like: how does this piece of code work? Why does the robot go to the left? When can I add two numbers together? What, exactly, is causing the conflict? What is the result of this code?

The basics first!

I think that students should know what programming is first, what you can do with it and how you can master the basic principles before you start with thick textbooks and assignments.

Computational thinking

This roadmap is about learning to program in Python, but whatever programming language you choose, you first have to train yourself in what is called 'computational thinking'. You don't suddenly become a programmer by being able to think this way, but you do learn to divide a problem into smaller parts. You learn to think in the abstract. This is what a programmer does.

Some definitions to start with

Seymour Papert (1928 - 2016), a South African born American professor of mathematics and education at Massachusetts Institute of Technology, introduced the concept ‘computational thinking (CT)’ (Papert, *Mindstorms: Children,Computers and Poweful Ideas*, 1980). He was a pioneer in artificial intelligence.

Jeanette Wing, professor at the Univerity of Pittsburgh, defines computational thinking as something that: ‘...involves solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science’ (Wing, 2006 p.33).

The Netherlands Institute for Curriculum Development (Stichting Leerplan Ontwikkeling or SLO) defines computational thinking on its website as: ‘...the process orientated formulation and reformulation of problems in such a way that it is possible to solve them with computer technology. It involves a collection of thought processes that use problem formulation and data organisation, analysis and representation to solve problems using ICT techniques and tools.’ (SLO, 2015).

The United Kingdom is often named as a forerunner in teaching computational thinking skills. Programming has been a compulsory subject for all state schools there since 2014. Training school children in solving problems in computer terms is central to the educational system. British children learn computational thinking from their fifth school year. Similar to ‘reading comprehension’, it is also known as ‘program comprehension’. When children leave primary school in their 11th year, they know two computer languages and they can operate robots and security systems. In secondary schools, they expand their knowledge under the supervision of physics, mathematics and IT specialists (Kennisnet, 2015).

This is quite different to my experience of Dutch education in recent years…

What is expected of you

Showing interest by downloading this roadmap is already a good start. The answer to the question ‘what is expected of you?’ is actually very simple: that you remain eager to learn, keep thinking for yourself, don’t give up if you don’t understand something right away and, above all, that you just make a start. These qualities are key to mastering the basics of programming.

In my experience as a programming teacher, many students get bogged down by using heavy textbooks in small print, by having to do too many exercises in too little time and teachers doing exercises for them. With this roadmap, however, you can master the basics at your own pace.

Thousands of people have gone before you, so I know that you can too. Stay interested and committed, and before you know it, you will be a real, live, programming expert.

Good luck and enjoy yourself.

Programming

I think it's clear by now that I'm not an advocate of thick textbooks on theory and demonstrating everything for students. So, as promised, I'm not going to do that. But I'm not going to throw you in at the deep end either. Did you know that YouTube, Instagram, Netflix and numerous other well-known web-based companies are all built on the programming language Python? Yes, really! If you don't believe it, see **this** video by Buvesa (in Dutch only). **Which route do you have to take to learn to programme?** Whatever you do, don't start by buying a book, one of those thick textbooks with lots of exercises I've been going on about. No, just follow this roadmap.

Route explanation

Step 1 Visual foundation of programming with RoboMind Academy

Learn how to give orders to the little robot car by using a visual program interface.

1. Go to the RoboMind Academy website (www.robomindacademy.com) and create a free account.
2. Start with the course *Hour of Code* and continue with *Basics 1 Course* and *Basics 2 Course*.
3. And if you enjoyed it, try this course as well: *Advanced 1 Course*.

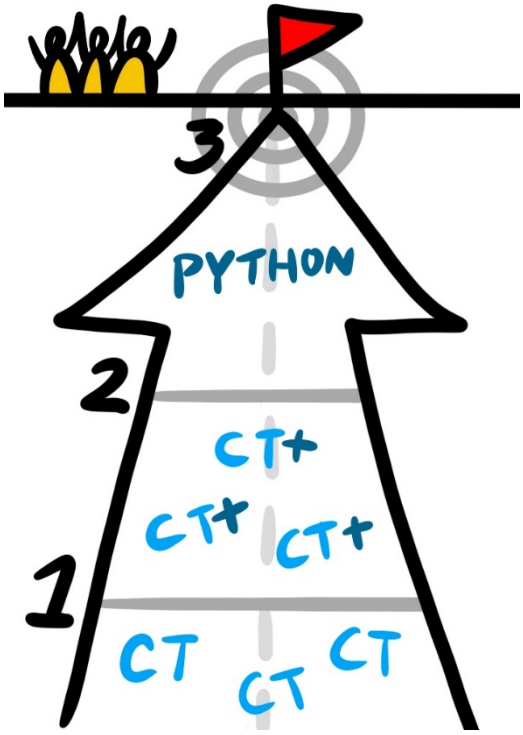


Figure 1 Roadmap to Learning Programming

Step 2 Visual programming in Python with RoboMind Academy

Learn the basics of Python (if-else, while loops, etc). Functions and variables will soon be familiar concepts.

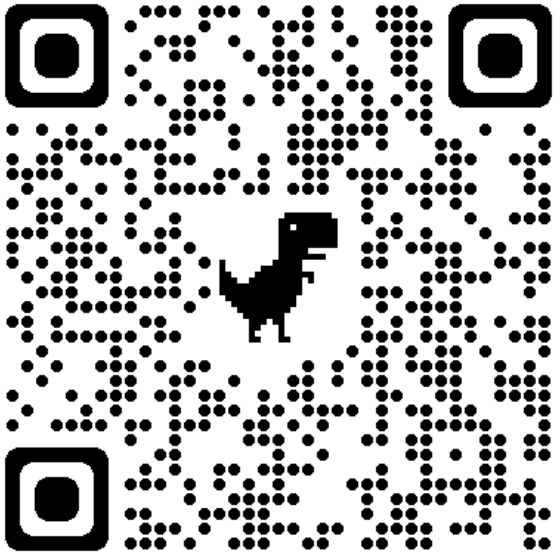
1. See the next course in RoboMind Academy called Python Course.
2. Run through the different lessons and learn the basic principles of the language Python.
3. Try out different exercises and refine your knowledge and skills.

Step 3 Python's 'real' programming environment

Use the book in the link for the next three steps: www.spronck.net/pythonbook/pythonboektablet.pdf

1. After completing the course Programming in Python in RoboMind Academy, it's time to make the move to Python's 'real' programming environment.
2. Download a popular Python development environment, like IDLE or PyCharm.
3. Experiment with writing simple scripts in Python and practice carrying out tasks in the environment. Keep practising and learning to improve your skills.

Did you find this roadmap useful? Please show your appreciation by making a donation.



Bibliography

Dunlosky, J. (2013). *Strengthening the Student Toolbox*. American Educator. https://www.academia.edu/29950252/Strengthening_the_Student_Toolbox_study_strategies_to_Boost_learning

Papert, S. (1978). Mindstorms: Children, Computers and Powerful Ideas by Seymour. . . Biblio.co.uk. https://biblio.co.uk/book/mindstorms-children-computers-powerful-ideas-papert/d/1508638816?aid=frg

SLO. (2015). *Computational Thinking*. Accessed from http://curriculumvandetoekomst.slo.nl/21e-eeuwse-vaardigheden/digitale-geletterdheid/computational-thinking

Wing, Jeannette. (2006). 'Computational Thinking', *Communications of the ACM.* ACM