Introduction\_to\_Python\_for\_Kids\_c01

Learn Python the Fun Way by Completing Activities and Solving Puzzles Aarthi Elumalai

**Introduction**

This book is the perfect blend of education and fun for kids 8 years and above looking to learn one of the easiest languages they can use to develop almost everything from websites to desktop apps to games to AI. It will include four big projects (or capstone projects): three games with Turtle, Tkinter, and Pygame and a desktop app with Tkinter.

The book starts with an overview of basic programming concepts such as variables, numbers, and strings while creating fun, personalized mini projects like “Print your name” and “Is your mom tipping enough.” It then dives right into Turtle, a Python library custom-made for kids, where they'll learn how to draw, animate, automate, and eventually make colorful mini projects based on the Python concepts learned. Once they have built a foundation in programming and the Python language, they will learn all about building desktop apps with Tkinter and games with Pygame.

There is also an entire chapter dedicated to more fun puzzles and activities that come with a step-by-step solution, and another chapter with cool ideas for more puzzles, and a section that gives them advice on where they can go from there. By the end of this book, kids will learn Python from the inside out while creating projects that they can showcase to their friends, families and teachers. They will develop problem-solving skills along with programming skills while doing the puzzles and activities described in the book.

This book is really jam-packed with information, but do not worry, I made sure that you will not be bored. You will not feel like sitting in another one of your boring classes at school, I promise.

I have included a lot of fun activities, mini as well as big projects throughout this book. There are also a lot of puzzles and even more activities at the end of the book, so you will not have a boring minute.



In Chapter 1, I’ll give you a brief introduction to Python, its uses, and an overview of what’s covered in this book and how to get the most out of it.

In Chapters 2 and 3, I will start by guiding you through the steps required to install Python in your system. It is quite easy to do, do not worry.

Then, you will create your very first program. Yes, you will start coding from the get-go! In

Chapters 4 and 5, you will learn all about numbers, performing mathematical calculations with Python, and the various cool tools Python equips you with to perform your magic with numbers. You will start coding real Python programs in this chapter.

Chapters 6 and 7 are where things get interesting. You will learn how to create a lot of cool graphics with Python’s built-in module, Turtle. Remember the add-ons I told you Python had to help you make awesome stuff? Turtle is one of the best of them. With Turtle, you can literally draw on the screen, and it will be automated! Once I have taught you Turtle, I will use it in the next chapters to make things more colorful.

In Chapter 8, you will learn how to play with alphabets, words, and sentences in Python. You will learn how to print things, extract words from sentences, find words in sentences, and a whole lot more.

Chapter 9 is where you will learn about conditions. There is always a cause and effect in life. If something happens, something else will happen because of it. “If I ace my test, my mom will praise me.” That is a cause and effect. You will learn how to apply scenarios like that and use it to create fun little mini games (you will see how) in this chapter. What if you want to print every number from 1 to 100? What if you do not want to write more than four lines of code to do that? You will learn how to do that and use the power of automation to automatically draw your graphics and animations in Chapter 10.

In Chapter 11, you will learn how to store a lot of information in one place. You will start using the real power of Python from this chapter onward.

In Chapter 12, we’re going to take a break from all the learning and create fun mini projects based on the concepts we’ve learned so far.

In Chapters 13 and 14, we will take a long look into real-world programming. We will look at true automation with functions and realworld problem-solving with objects.

In Chapter 15, you will learn how to automatically manipulate files with Python.

From Chapter 16, we will go back to having fun with Python! In Chapters 16, 17, and 18, you will learn all about a powerful package called Tkinter. You can use it to create desktop-based apps and games. You will learn how to use the package to create a paint app you can show off to your friends and a tic-tac-toe game you can play with your friends.

In Chapter 19, we will revisit Turtle and create a fun project with it. Let us create a snake game, shall we?

In Chapters 20 and 21, I will focus on making you a budding game developer. Let us learn how to use Pygame to create awesome games and make a space shooter game you can modify to your heart’s content!

In Chapter 22, we will learn the basics of web development with Python. We will not delve too much into it, but I will give you an introduction to get started.

In Chapter 23, we will go back to solving puzzles and activities. We will even make a couple more mini projects!

Chapter 24 is the final chapter. I’ll give you ideas on new projects and mini projects you can try with what we’ve learned in this book and also advice on what you can do next. What is covered in this book is just the starting point. There is a lot more to Python, and I will point you in the right direction to continue your journey

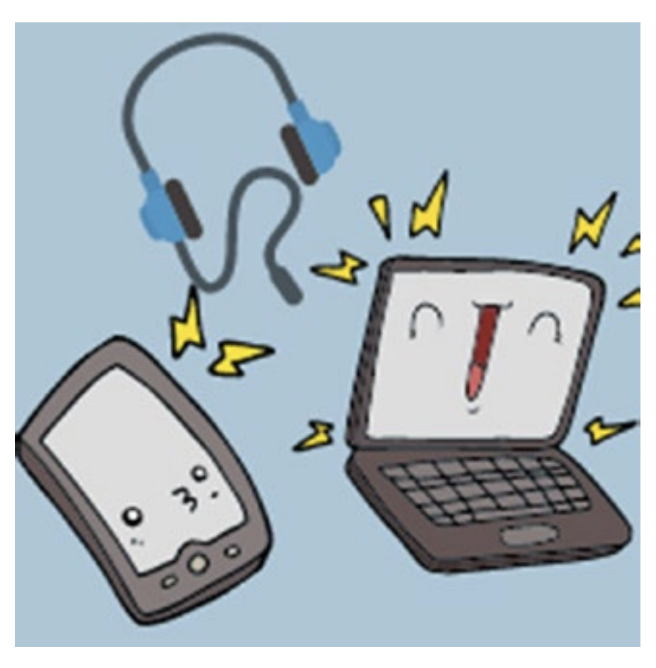
**CHAPTER 1 Did You Know?**

I have written the first part of this chapter for parents and the rest for the kids. I hope to convince you of the importance of programming and Python as your kid’s first programming language in this chapter. If your kid is older (10+), they can read those topics themselves. In the latter half of the chapter, I’ll address the kids and give them a brief overview of all the fun stuff they can do with Python, what they’ll learn from my book, and how to use my book to its full capacity.

So, let us get started.

**What is programming?**

You have your gadget – your laptops, PCs, tablets, mobile phones, and so on – and whenever you ask it to do something, within reason, it does it. How? Well, that is because every time you task your gadget with something, the pre-programmed set of instructions it has pertaining to that task will fire up in the background. Those set of instructions are called code.



You will find that your gadgets need a complete set of instructions to perform even the simplest tasks, like opening an application or performing a calculation. They are just machines after all, just 1s and 0s at the base of it all. They cannot think on their own, so with our code, we are making them think.

In other words, programming is the language your computer speaks, and the different programming languages are the different languages it understands/speaks. You might know English, French, and Mandarin, but you might not know Italian or Japanese. Similarly, out of the, literally, hundreds of programming languages out there (Python, JavaScript, C, C++, C#, Ruby, etc.), your computer might speak a few, or just one, and not understand the rest.

**Why should your kids learn to code?**

Now that you know what programming is and how it literally runs the digital world, I shouldn’t have to give you a lot of reasons to convince you to teach your kids coding, am I right?

But still, you might be wondering why your kid needs to learn programming, and why should they learn now. After all, in our time, people learned programming in college and only if they decided to become a programmer.

Well, I think I have a couple of reasons that will convince you why, in this day and age, kids, regardless of their future career aspirations, should learn programming, and why it is prudent that they start now.

**Programming is like Math**

Thirty years back, no one would have dared make such a statement, but now, times have changed, and yes, programming is, indeed, like Math. It is everywhere, just like Math is.



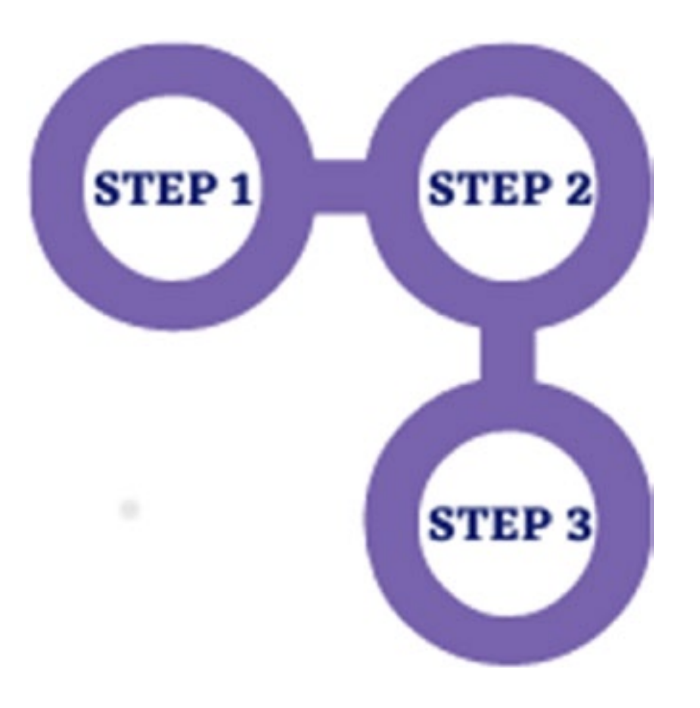
Math was a part of our educational requirement until we were at least 18, but not everyone of us is a mathematician today. Then why was Math forced down our throats? Well, because Math runs everything. We need basic mathematics in our daily life and certainly in most of our careers. So, we learned everything from calculus to algebra to geometry, knowing very well that we would probably not use 90% of our knowledge when we grew up.

That is exactly the case with programming today. Everything is digitized. There is an app for everything from food delivery to stock market prediction. Computers have entered every field, including traditional fields like construction and manufacturing. Most of the construction equipment are digitized nowadays, and what powers them? Programs, and thousands of lines of code.

Even art is digitized. So regardless of the field your kid is getting into, their knowledge in programming is going to give them a leg up. But apart from that, coding also improves a kid’s mathematical capabilities by fostering logical thinking and problem-solving.

**Coding improves logical thinking and creativity**

A contradictory statement, but true in this case. Every code blocks your child creates will be logically driven. Logic dictates programming, and once they start coding on their own, they will learn to dissect a problem into bite-sized components, apply logic to solve each component, and then finally combine all the components into a coherent solution.



That is how problem-solving works in the real world, regardless of field, and they will learn this invaluable skill as a child.

But let us be honest, your kid would be bored with just logic. That is where creativity comes in. The world runs on both creativity and logic, and the same holds true for programming.

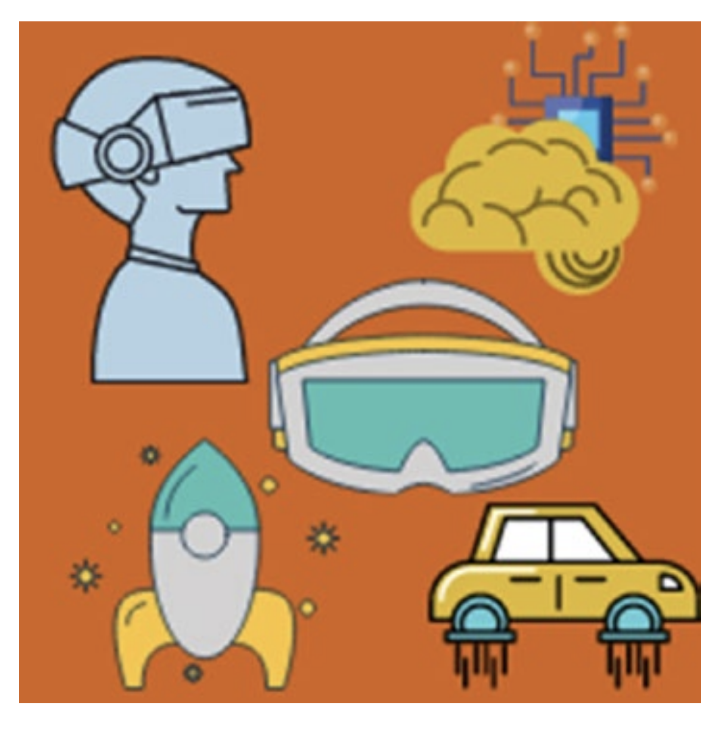
There is no right answer in programming. If they are solving the problem, how they are solving it does not matter. There are best practices, of course, but if you ask two programmers to solve the same problem, chances are, their code blocks will look completely different. So, while coming up with a solution, and multiple solutions for multiple problems down their programming journey, they will foster creativity as well.

The best of both worlds, don’t you think?

**Coding is the future**

Let us be very honest here. We are moving toward a completely digitized society at an amazingly fast pace. Everything is digitized. Apps are everywhere. Artificial intelligence is making new waves in the world every single day. Before we know it, we will have artificial intelligence–driven technologies cleaning our houses and driving our cars. It is no wonder that programming has become an invaluable skill in today’s world, and the demand for programmers is just going to increase every year.

So, coding is indeed the future, and by learning to code at a young age, your child will have a leg up over their competitor. Sounds good, but why does your kid need coding if they were going to become a mechanical engineer, for example? Or a financial analyst?



This brings me back to my original argument. Everything is digitized, and coding is everywhere, in every single field. So, if your kid has a programming base, then will they not stand out from the rest?

For example, a financial analyst with knowledge in programming can code a stock prediction application themselves and hence save the company tons of resources, or they would at least be fluent enough in tech talk that they’d guide the programmers better and save their boss hundreds of production hours and back and forth.

So, regardless of what your kid is going to study in the future, coding is going to help them out, and in the immediate future, they have a monetizable skill they can use to make extra cash.

In their teenage years, they do not have to flip burgers at minimum wage as a side gig. They can freelance as a software developer and make at least three to four times more money working half the time.

Or better yet, your kid could decide to become an entrepreneur. As you know, almost every startup relates to programming and software in some way, so as a programmer, your kid can code the app themselves and easily save tens of thousands in programmer fees. I believe those are very convincing arguments as to why your kid needs to start learning to code right now.

**Why Python?**



Alright, now you are convinced about the benefits of your kids learning to code, but why Python? Out of the tens of popular programming languages out there, why should they start with Python? I firmly believe that Python should be the first choice of real-world programming for a kid and let me convince you why.

**Python is easy**

Well, that is pretty much it. Kids want to have fun, and if we want them to learn programming, it needs to be fun and easy. Python is both. Unlike the other older languages where you need to learn a lot of syntaxes and theory before you can create your first program, Python is very straightforward. The syntax is easy to understand and logically sound. “Print” just prints something on the screen. Easy to remember, right? There is not a lot of memorizing to do, and your kid can start coding and creating from the get-go.

It is the perfect programming language for beginners with no prior knowledge of programming, and it is even better for kids because it is also fun

Python has a lot of built-in kid-friendly modules and libraries that will help them draw graphics and create games and fun apps with just a few lines of code.

**It can do a lot of things**

Now, do not underestimate Python because it is easy to learn. It can be used in literally everything from web development to desktop app development to artificial intelligence.

The language is immensely powerful, and the libraries and modules that come with it are even more powerful. You have add-ons for everything. You can create graphics with Turtle, beautiful desktop apps with Tkinter (like the calculator app you’re so fond of using on your laptop), create professional games with Pygame, develop full-blown websites and web apps with Django or Flask, and apply machine learning (artificial intelligence) algorithms with a host of easy to learn libraries. The possibilities are endless with Python.

By starting their programming journey with Python, your kids are not just stepping into the world of coding, they are also equipping themselves with skills in one of the most highly sought-after (and paid) programming languages of this time. What more? With the rate at which Python’s popularity, and its adoption into fast-moving fields like artificial intelligence, is growing, it is obvious that Python is here to stay, so your kids’ skills are unlikely to go stale in the future. From this point onward, I will be addressing the kids.

**Python is fun!**

Hi there! So, you are here to learn Python. Not only is it easy to get started with, it also comes with a lot of bells and whistles that make programming fun for you. Would you like to know all the cool things you can create with Python?

**Games!**

Who does not love games, am I right? But what if you can create your own games and then play them with your friends? You will be the most popular kid in class if you can do that.



What more? You can change the features of the games as you like. Want five lives instead of three? Great, add two more. Not enough levels and you are bored already? Code more levels into your game! Make those levels extra difficult to give yourself a challenge. You have the freedom to do anything you want with the games you create, and you can even get suggestions from your friends and apply them to your games. With just a little bit of coding, you will be able to modify the games you have always been dissatisfied with or create a brand-new game you can play with your friends.

So, you will have fun while coding and after coding (while playing the games).

**Graphics and animation**

For me, graphics and animations are the next best thing right after games. What about you?



Imagine running a program that draws a design you created in real time. What about an animation? What if you can create designs and animations and use them in that game you have always wanted to create and play? Like I always say, the possibilities are endless, and they are only limited by your creativity and imagination. Run wild with Python!

**Websites**

Do you use the Internet? Then you must have visited at least 100 websites by now. They look great, don’t they? What if you can create a website just like your most favorite website?

If you learn Python, you certainly can.

I am not talking about simple websites. I am talking about big, fullblown websites and web apps with a lot of cool features. You could even create websites and apps like Facebook and Instagram with enough practice.

**Apps**

Python comes with a lot of tools, just like the tools you use in your games. These tools are called libraries and packages in Python. With the help of these libraries, you can create almost anything, including apps.



Do you use a laptop or tablet? It comes with a lot of cool apps, right? There is a calculator app, a stopwatch/timer app, paint app, and so on.

What if you can create those exact apps? Well, with Python, you certainly can. In fact, you will learn to create some of those apps right in this book. Are you excited?

Not just that, with packages like Kivy and PyQt, you can even start creating mobile apps with Python. We will not be talking about those packages in this book, but as you can see, you have a lot of possibilities with Python. Whew! That was a big list indeed. The world is your oyster with Python, so come and play!

**Getting the most out of this book**

This chapter (and the last) would be the only two chapters with a lot of text. I have tried my best to keep things interesting and practical in the rest of the chapters. You will come across a lot of examples that illustrate every topic we cover. There will be a lot of coding, so I recommend you code the examples along with me. Try not to copy and paste. Type everything out so you get familiarized with coding.



Every chapter comes with a lot of activities, puzzles, and mini projects with detailed, step-by-step solutions too. I would recommend following along with the solutions in the first few chapters, but once you are confident enough, try solving the puzzles/activities on your own and then cross-verify with the given solution.

Remember, there are no wrong solutions in programming! If you get the desired result, you are good to go. This book includes four capstone projects (big projects) as well to solidify your knowledge of Python. I would recommend creating the projects, but do not stop there. Try changing things in every project to make it your own. Of course, do not forget to show your projects to your family, friends, and teachers as well! That is pretty much it. It is an easy-to-follow along book, so do not get overwhelmed by the size of it. Just get started and start coding.

**Summary**

In this chapter, which was addressed to the parents (in the first half) and the kids (in the second half), I gave a brief explanation of what programming is and why your kids need to learn to code at such a young age, regardless of their future aspirations. I also gave convincing arguments on why Python should be the first real-world programming language of choice for a kid and what a kid can do with Python. We ended the chapter with a brief overview of everything you will learn from this book and the best way to utilize this book to its full capacity. In the next chapter, we will learn how to install Python and create and execute our very first Python programs.