



# Introduction to Python for Kids

Learn Python the Fun Way by  
Completing Activities and  
Solving Puzzles

—  
Aarthi Elumalai

Apress®

# **Introduction to Python for Kids**

**Learn Python the Fun Way by  
Completing Activities  
and Solving Puzzles**

**Aarthi Elumalai**

**Apress®**

# *Introduction to Python for Kids*

Aarathi Elumalai  
Chennai, Tamil Nadu, India

ISBN-13 (pbk): 978-1-4842-6811-7  
<https://doi.org/10.1007/978-1-4842-6812-4>

ISBN-13 (electronic): 978-1-4842-6812-4

Copyright © 2021 by Aarathi Elumalai

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Managing Director, Apress Media LLC: Welmoed Spahr  
Acquisitions Editor: Celestin Suresh John  
Development Editor: Laura Berendson  
Coordinating Editor: Aditee Mirashi

Cover designed by eStudioCalamar

Cover image designed by Freepik ([www.freepik.com](http://www.freepik.com))

Distributed to the book trade worldwide by Springer Science+Business Media New York, 1 New York Plaza, Suite 4600, New York, NY 10004-1562, USA. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail [orders-ny@springer-sbm.com](mailto:orders-ny@springer-sbm.com), or visit [www.springeronline.com](http://www.springeronline.com). Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail [booktranslations@springernature.com](mailto:booktranslations@springernature.com); for reprint, paperback, or audio rights, please e-mail [bookpermissions@springernature.com](mailto:bookpermissions@springernature.com).

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at <http://www.apress.com/bulk-sales>.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub via the book's product page, located at [www.apress.com/978-1-4842-6811-7](http://www.apress.com/978-1-4842-6811-7). For more detailed information, please visit <http://www.apress.com/source-code>.

Printed on acid-free paper

*Dedicated to My Daughter and to the rest of my family  
who believed me when no one would and stood by me  
through everything.*

# Table of Contents

<b>About the Author .....</b>	<b>xix</b>
<b>About the Technical Reviewer .....</b>	<b>xxi</b>
<b>Introduction .....</b>	<b>xxiii</b>
<b>Chapter 1: Did You Know? .....</b>	<b>1</b>
What is programming?.....	1
Why should your kids learn to code? .....	2
Why Python?.....	7
Python is fun!.....	8
Games! .....	9
Graphics and animation.....	9
Websites .....	10
Apps.....	10
Getting the most out of this book.....	11
Summary.....	12
<b>Chapter 2: Let's Install Python! .....</b>	<b>13</b>
Speak the computer's language .....	13
Get started – install Python .....	14
Installing Python on a Windows computer .....	14
Download Python.....	15

TABLE OF CONTENTS

- Install Python..... 16
- Installing Python on a Mac device..... 18
- Download Python..... 18
- Install Python..... 20
- Summary..... 25
- Chapter 3: Your First Python Program.....27**
- Creating and running programs in Python ..... 27
- Python interactive mode (Python Shell) ..... 30
- Your Shell can do Math ..... 30
- Print with Python..... 32
- IDLE script mode..... 33
- Python activity: Print your name (and some more) ..... 36
- Summary..... 37
- Chapter 4: Python Loves Numbers .....39**
- Numbers in Python..... 39
- Store your numbers ..... 40
- Comments..... 46
- Your numbers come in different forms ..... 47
- Integers ..... 48
- Floating-point numbers ..... 49
- Complex numbers..... 50
- Type conversion between numbers ..... 53
- Mini project – Do you understand numbers?..... 57
- Summary..... 58

<b>Chapter 5: Let's Play with Our Numbers!</b> .....	<b>59</b>
Get your numbers out to play.....	59
Basic Math operations .....	60
Special Math operations in Python .....	62
Assignment operations .....	65
What comes first? .....	67
Cool stuff with numbers.....	70
Floor and ceiling of a number .....	71
Power and square root.....	72
Factorial of a number.....	73
Sin, cos, tan, and more .....	73
More numerical operations.....	74
Working with random numbers.....	75
Mini project – multiples of a number.....	78
Summary.....	81
<b>Chapter 6: Drawing Cool Stuff with <i>Turtle</i></b> .....	<b>83</b>
Let's get started .....	83
Make your <i>Turtle</i> move.....	86
Move forward and backward .....	86
Make your turtle change directions .....	89
Mini project – draw a square .....	91
Mini project – draw a hexagon .....	93
Shortcuts.....	95
Go to random points on the screen.....	96
Draw a square with goto.....	98
Mini project – draw a mandala (with just straight lines) .....	100
Summary.....	105

TABLE OF CONTENTS

<b>Chapter 7: A Turtle Deep Dive .....</b>	<b>107</b>
Customize your screen .....	107
Customize your graphics .....	109
Shapes without lines.....	112
Circles .....	112
Dots.....	113
Arcs.....	114
More options!.....	116
Draw text on screen.....	120
Mini project – circle within a square .....	124
Change directions of your drawing.....	126
Mini project – smiley .....	131
Summary.....	135
<b>Chapter 8: Play with Letters and Words .....</b>	<b>137</b>
What are strings?.....	137
Let’s create some strings.....	138
I want lines and lines of strings!.....	139
My string has quotes! :O .....	141
Let’s join two or more strings .....	143
Concatenation in print() .....	144
Empty string.....	145
Accessing characters in strings.....	145
Negative indices .....	147
Slicing a part of a string.....	148
String methods – magic with strings!.....	149
Capital and small .....	150



Misc methods .....	152
True? False?.....	155
String formatting.....	157
Getting input from the users (start automation).....	161
String to int or float conversion .....	162
Mini project – take <i>Turtle</i> text to the next level! .....	164
Mini project – shout at the screen .....	166
Mini project – reverse your name .....	169
Mini project – colorful and dynamic Math .....	171
Summary.....	174
<b>Chapter 9: Follow My Command! .....</b>	<b>175</b>
True or False .....	175
Compare and decide .....	179
If this happens, do this (command!).....	180
Else? .....	183
More than one condition! :0.....	184
Mini project – guess the number game .....	186
The conditions keep stacking up! .....	189
Summary.....	193
<b>Chapter 10: Automate a Little.....</b>	<b>195</b>
Magic loops!.....	195
For loops .....	197
If statements within for loops .....	200
Nested for loops.....	201
Iterating over strings.....	204
While loops.....	205

TABLE OF CONTENTS

Abort mission! Break and continue..... 207

Mini project – guess the number game version 2 ..... 209

Mini project – automate your square..... 211

Mini project – automate any basic shape..... 212

Mini project – automatically draw a mandala design..... 216

Mini project – arc spirals ..... 218

Summary..... 221

**Chapter 11: Lots and Lots of Information!.....223**

Store more than one value..... 223

Lists ..... 225

Accessing values in a list..... 226

Slice a list! ..... 227

List manipulation on fire! ..... 228

    Copy and append..... 229

    Count and clear ..... 229

    Concatenation..... 230

    Search inside your list..... 231

    Add and remove elements..... 232

    Reverse and sort ..... 234

More fun with lists! ..... 235

Mini project – multi-colored automated star ..... 238

Tuples..... 240

Sets..... 244

Dictionaries ..... 247

Mini project – never-ending colors ..... 252

Mini project – first and last name reversal ..... 255

Summary..... 258

<b>Chapter 12: Fun Mini Projects Galore!</b> .....	<b>259</b>
Project 12-1: Odd or even .....	259
Part 1 – Is your number odd or even? .....	260
Part 2 – print odd or even numbers within a range.....	261
Project 12-2: Is your mom tipping enough?.....	262
Project 12-3: Draw a Christmas tree.....	264
Project 12-4: Spirals! .....	268
Square spiral.....	269
Random spiral.....	271
Triangular spiral .....	272
Star spiral.....	273
Circular spiral.....	274
Project 12-5: Complex mandala – completely automated .....	276
Project 12-6: <i>Turtle</i> race with loops .....	277
Summary.....	281
<b>Chapter 13: Automate with Functions .....</b>	<b>283</b>
True automation .....	283
Our first function .....	285
Why do we need functions?.....	286
Do different things every time! .....	287
Create (define) your functions .....	288
You can reuse your code! .....	289
No arguments? .....	291
Give an answer .....	292
No arguments? What to do! .....	295
Too many arguments! .....	295

TABLE OF CONTENTS

Global vs. local .....	297
Variables within functions .....	298
Return local variables.....	299
Global variables .....	299
Lambda .....	301
Mini project – do your Math homework with Python.....	302
Mini project – automated shapes – next level.....	306
Summary.....	309
<b>Chapter 14: Let’s Create Real-World Objects.....</b>	<b>311</b>
What is object-oriented programming? .....	312
Let’s prove it! .....	313
Classes.....	314
Objects with their own values.....	315
Manipulate your objects.....	317
Objects do stuff.....	318
<i>Turtle</i> race with objects .....	319
Summary.....	322
<b>Chapter 15: Python and Files.....</b>	<b>323</b>
Why files? .....	323
Opening and reading existing files.....	324
Line by line.....	328
Create new files .....	330
Manipulate files .....	330
Mini project – introduce with files .....	332
Summary.....	333

<b>Chapter 16: Create Cool Apps with <i>Tkinter</i>.....</b>	<b>335</b>
<i>Tkinter</i> – let’s set it right up!.....	335
Labels, buttons, and packing them .....	337
Packing in detail .....	342
Lots of inputs .....	348
One line of text.....	349
Line after line .....	352
<i>Tkinter</i> variables .....	355
Lots of options! .....	357
Menus .....	361
The perfect layout – grid.....	363
Mini project – tip calculator app .....	365
Summary.....	368
<b>Chapter 17: Project: Tic-tac-toe Game with <i>Tkinter</i>.....</b>	<b>369</b>
Bind events – make your apps dynamic!.....	369
Tic-tac-toe game – explanation .....	372
Set up <i>Tkinter</i> .....	373
Create global variables .....	374
Create the buttons .....	376
When the button is clicked, draw on it.....	378
Check if a player won during each turn .....	383
New game button.....	385
Entire program .....	387
Summary.....	390

TABLE OF CONTENTS

**Chapter 18: Project: Paint App with *Tkinter* .....391**

- Paint app – explanation ..... 391
- Get started ..... 392
- Set up the screen ..... 393
- Create the canvas ..... 394
- Create your first menu (shapes)..... 395
- Let's make our draw options work! ..... 396
- Get the mouse position ..... 398
- Let's draw our lines..... 399
- Squares and rectangles! ..... 401
- Circles and ovals! ..... 402
- Select size!..... 403
- Lots and lots of colors!..... 404
- I've finished drawing! ..... 407
- Entire program ..... 408
- Summary..... 412

**Chapter 19: Project: Snake Game with *Turtle*.....413**

- Snake game ..... 413
- Import the required modules ..... 415
- Set up the turtle screen ..... 415
- Create and initialize the required variables ..... 417
- Draw the head..... 417
- Draw the first apple ..... 419
- Is my screen registering my arrow presses?..... 421
- Make our snake head move ..... 423
- Get the scoreboard going..... 426

Our snake's eating! .....	428
Make the entire snake move.....	431
Collision check .....	434
Entire code .....	439
Summary.....	444
<b>Chapter 20: Become a Game Developer with <i>Pygame</i>.....</b>	<b>445</b>
What is <i>Pygame</i> ? .....	445
Install and import <i>Pygame</i> .....	446
Set up your gaming screen!.....	448
Make your screen pretty .....	450
Create your characters on the screen.....	452
Move your characters .....	457
Keyboard press events.....	459
Mini project – bouncing ball .....	462
Summary.....	465
<b>Chapter 21: Project: Space Shooters with <i>Pygame</i>.....</b>	<b>467</b>
Space shooter game .....	467
Import the required modules .....	469
Initialize everything.....	469
Game loop .....	470
Create the spaceship .....	471
Move the spaceship .....	473
Create and move the enemies .....	475
Fire the bullet.....	479
Create and display the scoreboard .....	482
Kill the enemies .....	484

TABLE OF CONTENTS

Kill the spaceship!.....487

Re-draw the enemies.....488

Game over!.....489

Entire code.....491

Summary.....496

**Chapter 22: Web Development with Python .....497**

Python and web development.....497

Building blocks – HTML .....499

Pretty things up – CSS .....502

Front-end dynamic – JavaScript.....504

Python’s Flask.....507

Summary.....510

**Chapter 23: More Mini Projects.....511**

Project 23-1: Calculator with *Tkinter*.....511

Project 23-2: Random story generator.....518

Project 23-3: Rock Paper Scissors game.....521

Project 23-4: Bouncing ball (off the four walls) with *Pygame*.....527

Project 23-5: Temperature conversion app .....531

Project 23-6: Introduce with files and *Tkinter*.....534

Summary.....538

**Chapter 24: What’s next?.....539**

Mini project ideas you can try.....539

    Currency conversion app.....539

    Race in *Pygame*.....540

    More patterns in *Turtle*.....540



TABLE OF CONTENTS

Capstone project ideas you can try.....540

- Snake game in *Pygame*.....540
- Dodge the bullet .....541
- Memory game in *Pygame*.....541

Looking ahead.....542

- OOPs in detail .....542
- Regular expressions .....542
- Web development.....543
- Packages in detail .....543

Summary.....543

**Index.....545**

# About the Author

**Aarthi Elumalai** is a programmer, educator, entrepreneur, and innovator. She has a Bachelor of Engineering degree in Computer Science from Anna University, Chennai.

Since then, she has managed a team of programmers and worked with hundreds of clients. She has also launched a dozen web apps, plug-ins, and software that are being used by thousands of customers online.

She has over 15 years of experience in programming. She started coding in Basics at the age of 12, but her love for programming took root when she came across C programming at the age of 15.

She is the founder of DigiFisk, an e-learning platform that has more than 60,000 students all over the world. Her courses are well received by the masses, and her unique, project-based approach is a refreshing change to many. She teaches the complex world of programming by solving a ton of practical exercises and puzzles along the way. Her courses and books always come with hands-on training in creating real-world projects using the knowledge learned, so her students get better equipped for the real world.

When she is not working on her next course or book, you'll see her researching her next product idea and refining her existing products. She is currently committed to bringing the sheer power of artificial intelligence to make life easier for small business owners.

# About the Technical Reviewer

**Ashwin Pajankar** holds a Master of Technology from IIIT Hyderabad. He started programming and tinkering with electronics at the tender age of 7. BASIC was the first programming language he worked with. He was gradually exposed to C programming, 8085, and x86 assembly programming during his higher secondary schooling. He is proficient in x86 assembly, C, Java, Python, and Shell programming. He is also proficient with Raspberry Pi, Arduino, and other single-board computers and microcontrollers. Ashwin is passionate about training and mentoring. He has trained more than 60,000 students and professionals in live and online training courses. He has published more than a dozen books with many international and Indian publishers. He has also reviewed numerous books and educational video courses. This is his fifth book with Apress, and he is working on more books. He regularly conducts programming bootcamps and hands-on training for software companies in Nashik, India.

He is also an avid YouTuber with more than 10,000 subscribers to his channel. You can find him on LinkedIn.

# 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 real-world 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.

## INTRODUCTION

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?

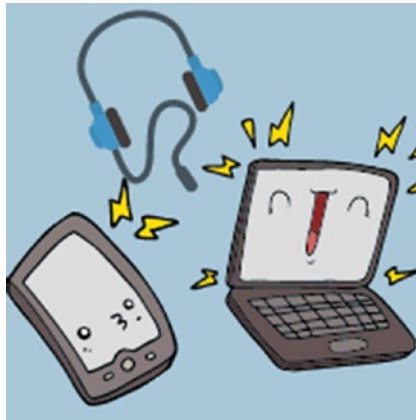
I have written the first part of this chapter for parents and the rest for the kids. I hope to convince you of the importance of programming and Python as your kid's first programming language in this chapter. If your kid is older (10+), they can read those topics themselves. In the latter half of the 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.

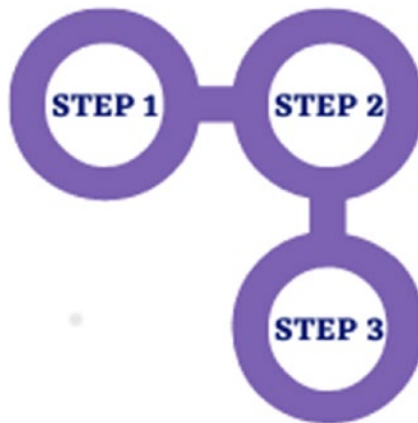
Even art is digitized. So regardless of the field your kid is getting into, their knowledge in programming is going to give them a leg up.

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.

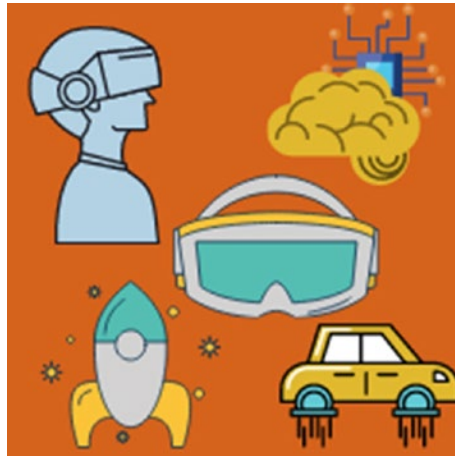
The best of both worlds, don't you think?

## **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.