

# PYTHON FAST

---

A 12-Week Guide to  
Mastering Python in 2023



# The key to learning Python Fast

Are you ready to unlock the power of programming and take your skills to the next level and learn Python fast in 2023?

Look no further than Python, the versatile and dynamic language that's taking the tech world by storm.

Python has a user-friendly syntax and can be the perfect choice for you if you are looking to break into the exciting and ever-evolving world of coding.

Whether you're interested in building your own software, creating web applications, or analyzing data, Python has the tools you need to succeed.

Start your journey today and discover the endless possibilities that await Python!



Python's power lies in its simplicity, versatility, and vast community-driven ecosystem of libraries and tools that make it suitable for various applications such as web development, data analysis, machine learning, and artificial intelligence.

Are you ready to be part of that?

# The key to learning Python Fast

---

In this guide, you will discover how you can learn Python fast in 2023.



- **Set a clear goal:** Determine why you want to learn Python and what you want to accomplish with it. This will help you focus your learning efforts and motivate you to keep going.



- **Learn the basics:** Start with the fundamentals of Python programming such as variables, data types, functions, loops, and conditionals.



- **Practice coding regularly:** The more you code, the faster you'll learn. Start with simple exercises and gradually move on to more complex programs.



- **Find a mentor or join a community:** Learning from someone who has experience in Python can help you learn faster and avoid common pitfalls.



- **Use Python libraries and frameworks:** Python has many powerful libraries and frameworks that can simplify your coding and help you build more complex applications faster.



- **Work on projects:** Try building real-world applications using Python. This will help you gain practical experience and learn how to apply your knowledge to solve real problems.



- **Be patient and persistent:** Learning Python takes time and effort. Don't expect to become an expert overnight. Stay patient, be persistent, and keep practicing consistently.

# What you will learn in this guide?

---

First the guide will cover key aspects of learning Python fast.



**Set a clear goal for your Python journey**



**Learn the basics of Python fast**



**Practice coding regularly**



**Find a mentor or join a community**



**Use Python libraries and frameworks**



**Work on projects**



**Be patient and persistent**



**And how to learn Python in 12 weeks!**

# Hi, I'm Rune

---

I may not have learned **Python** quickly or possess any special talent in coding.

In fact, I **struggled** just like many others when I was starting out.

However, what sets me apart is my **passion for sharing** what I've learned and helping others succeed.

I take pride in being able to guide people with **no technical background** towards achieving their coding goals.

Through perseverance and a willingness to learn, I've been able to help others **succeed faster than me**.

The truth is, **coding can be easy**, and I'm passionate about sharing that message with others.

With over 20 years of experience and a Ph.D. in Computer Science, our instructor Rune has dedicated countless hours to creating a comprehensive resource for both beginners and advanced learners of Python.



# Set a clear goal for your Python journey

---

I know it can be difficult to set a goal when you are just starting to learn to program in Python. But this is one of the things that will speed up your learning.

If you want to learn Python fast, you need to know what to learn and more importantly, what not to learn.

Python is known for its versatility, which is a core strength of the language, but it also makes it difficult when you are a learner. There are so many options and possibilities to dive into.

## The importance of a Python goal

### Motivation



Goals motivate

### Relevance



Knowing the WHY

### Efficiency



Focus on result

### Measurability



Milestones to goals

### Practicality



Learn what matters

# Set a clear goal for your Python journey

---

A clear goal will help you both to learn faster and stay focused.



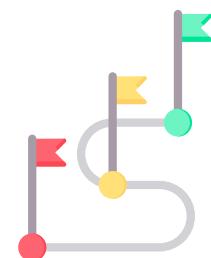
- **Motivation:** Having a specific goal in mind can help you stay motivated and focused as you learn Python. It gives you a reason to continue learning and practicing, even when the material may be challenging or confusing.



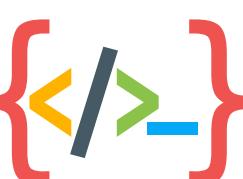
- **Relevance:** Knowing why you are learning Python and what you hope to accomplish with it can help you stay engaged with the material. It makes the learning process more relevant to your needs and interests.



- **Efficiency:** By having a clear goal, you can focus your learning on the specific skills and concepts that are most relevant to your goal. This can help you learn more efficiently and avoid wasting time on topics that are not important to you.



- **Measurability:** Having a goal can help you track your progress and measure your success as you learn Python. You can set milestones or benchmarks for yourself and celebrate your achievements along the way.



- **Practicality:** Learning Python without a goal can leave you with a bunch of disconnected knowledge that is hard to apply in the real world. Having a goal can help you see how the things you are learning can be used to solve real problems and create practical solutions.

# Set a clear goal for your Python journey

---

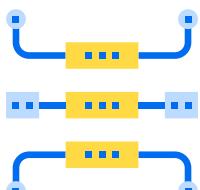
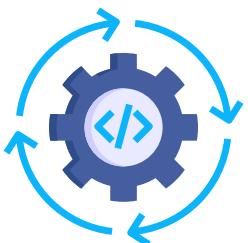
Here are ten common goals for using Python

- **Web development:** Python has various web frameworks such as Django and Flask, making it an excellent choice for building web applications.
- **Data analysis and visualization:** Python has a rich ecosystem of libraries such as NumPy, Pandas, and Matplotlib, which are widely used for data analysis and visualization.
- **Machine learning:** Python has several libraries for machine learning, such as Scikit-learn, TensorFlow, and PyTorch, making it a popular choice for building machine learning models.
- **Automation:** Python's ease of use and versatility make it an excellent choice for automating repetitive tasks such as data cleaning, scraping, and file manipulation.
- **Scripting:** Python is a popular choice for writing scripts for system administration, automation, and other tasks.
- **Scientific computing:** Python has libraries such as SciPy, SymPy, and Biopython, which make it an excellent choice for scientific computing.
- **Game development:** Python has several game development libraries such as Pygame and Arcade, making it a popular choice for building games.
- **Desktop application development:** Python can be used for building desktop applications with libraries like PyQt, PyGTK, and wxPython.
- **Artificial intelligence:** Python's vast library ecosystem and ease of use make it an excellent choice for building artificial intelligence applications.
- **Education:** Python's simplicity and readability make it an excellent choice for teaching programming to beginners.

# Learn the basics of Python fast

---

If you want to learn Python fast you need to focus on the Python Essentials, which are the basics that you need in order to master programming and avoid learning fancy constructions that will not enable you to do anything you can't do without.



- **Learn variables, data types, operators, and control structures.** This is the basics of Python programs and will enable you to solve simple problems and will give you a solid foundation to build on.
- **Practice data structures,** like lists, dictionaries, sets, and tuples. Data is the core of a program, and mastering the Python built-in data structures will help you work with data.
- **Learn functions,** which will enable you to organize your code into smaller blocks of code. They enable you to solve bigger problems in a structured way, where you can test if each function works as expected.
- **Learn how to work with files.** Learn how to read from and write to files in Python. This is a fundamental skill that you'll need for many programming tasks.
- **Learn how to work with libraries.** Python has a vast collection of libraries that you can use to extend its functionality. Learn how to install and use popular libraries like NumPy, Pandas, and Matplotlib.

This will enable you to create powerful applications with Python.



# Practice coding regularly to learn Python fast

While it shouldn't be a surprise, you need to code to learn to code. And code a lot.

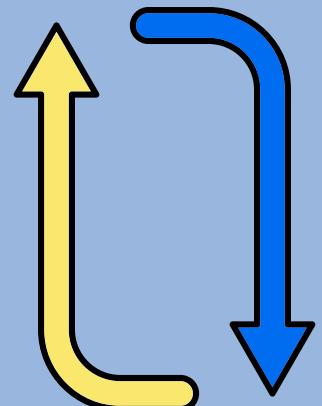
**Code**



**Sleep**



**Repeat**



Dedicate time and a great environment to learn in.



# Practice coding regularly to learn Python fast

---

The main reasons you should code a lot include the following.

- **Practice makes perfect.** The more you code, the more comfortable and efficient you will become with the language.
- **A better understanding of the language.** Regular coding will help you better understand the syntax, rules, and concepts of Python.
- **Problem-solving skills.** Coding regularly challenges you to solve problems and think critically, which improves your problem-solving skills.
- **Access to resources.** When you code regularly, you become more involved in the programming community and have access to a wide range of resources, tutorials, and support.
- **Better retention.** Regular coding ensures that you retain the information you learn and build on it over time.
- **Keeping up with updates.** Python is an evolving language, and coding regularly ensures that you stay up-to-date with the latest updates and changes.
- **Career opportunities.** Learning Python through regular coding can open up various career opportunities in fields such as data science, machine learning, web development, and more.
- **Fun and rewarding.** Coding in Python can be a fun and rewarding experience, especially when you see the results of your efforts come to life.

Please remember the last one. Keep it fun and rewarding.



# Find a mentor or join a community

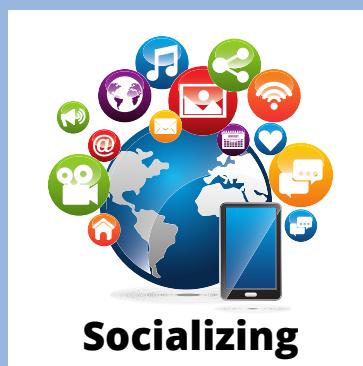
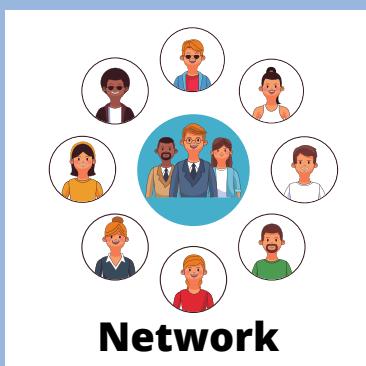
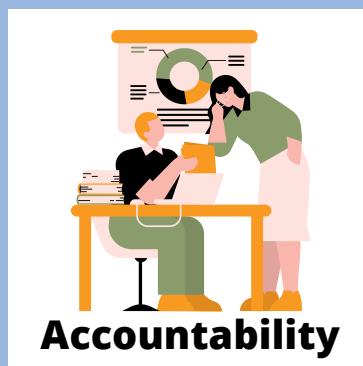
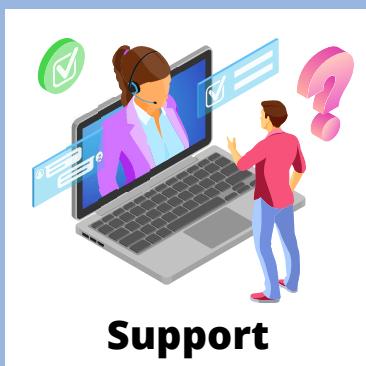
---

You sit with your Python program that just doesn't work as you expect it to work.

This problem bothers you.

You asked an A.I. service, but the answer is not fulfilling and the solution does not work.

In reality, what you need is an experienced programmer to ask and join a community.



# Find a mentor or join a community

---



- **Access to a support system.** Being part of a community or having a mentor provides access to a support system of like-minded individuals who can provide feedback, answer questions, and offer guidance.
- **Faster learning.** Being part of a community or having a mentor can help accelerate the learning process by providing access to resources, tips, and strategies that you may not have discovered on your own.
- **Accountability.** A community or mentor can help keep you accountable and motivated to continue learning and practicing Python regularly.
- **Networking opportunities.** Being part of a community can help you expand your network and connect with other professionals in the field, which can lead to new opportunities.
- **Exposure to new ideas.** Being part of a community or having a mentor can expose you to new ideas, perspectives, and approaches that you may not have considered on your own.
- **Feedback on your work.** A community or mentor can provide valuable feedback on your code and projects, helping you improve and grow as a Python programmer.
- **Avoiding common mistakes.** By learning from experienced programmers, you can avoid common mistakes and pitfalls that many beginners make.
- **Personalized learning.** A mentor can provide personalized guidance and support tailored to your specific learning needs and goals.
- **Fun and socializing.** Being part of a community can also be a fun and social experience, as you connect with others who share your passion for Python programming.

# Use Python libraries and frameworks

---

The key to really utilizing the power of Python is by using libraries and frameworks.

- **Reduced Development Time.** When a programmer uses a library or framework, they can leverage pre-built code modules that solve common problems, which means they can complete their work faster. For example, instead of writing code from scratch to handle HTTP requests, a programmer can use the popular `requests` library, which is optimized and pre-built, saving a significant amount of time.
- **Higher Productivity.** By using libraries and frameworks, programmers can focus on high-level tasks and logic, rather than spending time on low-level implementation details. This results in increased productivity, allowing programmers to get more done in less time.
- **Increased Efficiency.** Libraries and frameworks are often optimized for performance and efficiency, meaning that they can execute tasks faster than if a programmer were to write the code from scratch. For example, if a programmer needs to process large datasets, they can use the `pandas` library, which is specifically designed for high-performance data analysis, and can process large datasets much faster than if they were to write the code themselves.
- **Access to Expertise.** Libraries and frameworks are often created and maintained by expert developers, who have already solved common problems and can provide solutions that are optimized and tested. This means that by using these pre-built tools, programmers can leverage the expertise of others, which can help them achieve results faster.
- **Community Support.** Libraries and frameworks often have large and active communities of developers who share knowledge and provide support. This means that if a programmer encounters a problem or needs help, they can often find answers quickly, saving time and helping them achieve results faster.

# Use Python libraries and frameworks

---

While this explains why you want to start using libraries and frameworks there are some other benefits.



- **Reusable Code.** Libraries and frameworks provide pre-built and reusable code modules that can be easily integrated into your own projects. This saves time and effort that would otherwise be spent building everything from scratch.



- **Abstraction.** Libraries and frameworks often abstract away the low-level details of a task, allowing you to focus on the high-level logic of your project. This allows you to be more productive and make progress faster.



- **Robustness.** Libraries and frameworks are often tested and debugged by a large number of users, which can help to identify and resolve issues faster than if you were working alone.



- **Standardization.** Many libraries and frameworks follow standardized best practices and coding conventions, which can help to ensure that your code is consistent and maintainable over time.

# Work on projects and learn Python fast

---

With your community make projects that resemble real-world projects done with Python.



- **Build a website using Django or Flask.** Learn to develop web applications using Python web frameworks, which can help you build robust, scalable, and maintainable web applications.
- **Develop a chatbot** using Natural Language Processing techniques. Learn to use NLP techniques to build intelligent chatbots that can understand and respond to natural language queries.
- **Create a data visualization** dashboard using libraries like Matplotlib, Plotly, or Bokeh. Learn to use data visualization libraries to create interactive and informative data dashboards, which can help you present your data in a visually appealing manner.
- **Develop a machine learning model** to predict customer churn or classify images. Learn to use machine learning algorithms to build predictive models, which can help you solve real-world business problems such as customer churn prediction or image classification.



# Work on projects and learn Python fast

---

With your community make projects that resemble real-world projects done with Python.



**BeautifulSoup**



**Pygame**



- **Using OpenCV**, you can create a face recognition system that detects faces in real-time video streams and identifies them based on a pre-trained model or user-defined dataset.
- **Develop a web scraper** to collect data from websites using libraries like BeautifulSoup or Scrapy. Learn to extract data from websites by writing web scrapers, which can help you automate data collection tasks.
- **Implement a blockchain-based application** using Python. Learn to use Python to build decentralized applications using blockchain technology, which can help you build secure and transparent applications.
- **Create a game using Pygame or Unity game engine** with Python scripting. Learn to use Python for game development, which can help you build engaging and interactive games.

# Be patient and persistent

---

While we live in a world where everything is fast-paced getting new skills takes time.

Learning Python can sometimes feel like a Marathon.  
But remember, everyone, can finish a Marathon.

Just slow down and walk the distance, then you will eventually reach the goal.



# 12-week Python plan

---



## From zero Python to using libraries and frameworks in 12 weeks.

- This guide is designed to take you from a beginner with no prior experience in Python to a proficient user who can leverage libraries and frameworks for various applications in just 12 weeks.

This will enable you to use big frameworks and libraries on your own after 12 weeks.

- Python has a vast array of libraries and frameworks, which offer pre-written code and tools for solving complex problems and building applications with greater ease and speed.
- Frameworks like Django, Flask, and Pyramid make it easier to build web applications in Python, reducing the amount of boilerplate code and providing a consistent structure for development.
- Python libraries like NumPy, Pandas, and Matplotlib make it easier to analyze and visualize data, which is essential for scientific computing and machine learning.
- Python frameworks and libraries can help you leverage the collective knowledge of the Python community, providing access to a wealth of pre-built code, tutorials, and best practices.
- Using Python frameworks and libraries can reduce development time and costs, enabling you to build more robust applications with fewer resources.

# 12-week Python plan

---

## Week 1. Variables

- Learn about variables, types, and conversions in Python, which makes data reusable, recallable, and reassignable.
- Get comfortable with Python environment

## Week 2. Strings

- Learn about Strings, which enables you to work with text and raw data.
- How to get input from the user, which enables your programs to interact with the user.
- Also how to format the output to the user.

## Week 3. Data Structures

- Python lists will enable you to have any number of items in a list, just like a list.
- Dictionaries can be used for a lookup data structure, which can be used in many powerful ways.
- Sets and tuples enable you to do some awesome magic.

## Week 4. Python flows

- If statements enable you to make things depending on circumstances, which enables your programs to do different things each time, depending on what you feed it with.
- For-loops enable you to iterate over all items in a list, which enable you to process them all.
- While-loops enable you to do something until the user quits.

# 12-week Python plan

---

## Week 5. Libraries

- Get randomness into your programs, which makes your programs fun and do different things.
- Guess a number game, with randomness this is suddenly possible.
- Work with datetime objects, which is a core skill when working with data.

## Week 6. Functions and methods

- Learn how functions help you solve bigger problems, which is the next step of your programming.
- How functions are building blocks of something bigger, which enables you to solve bigger problems.
- How functions ensure your code is correct because you have a piece of code isolated.

## Week 7. Milestone #1 project

- Understanding building bigger projects is a key aspect to understand.
- Create simple board games as a warmup exercise.
- Create a Tic-Tac-Toe game, your first Milestone project.

## Week 8. Files

- Learn to read and write files, which is crucial to work with data from various sources.
- How to append to existing files, which teaches you how to update files.
- The principles of data processing of files, where you will learn best practices and how to keep it simple.
- Learn about CSV and JSON, the most common data structures that are used in the world of programming.

# 12-week Python plan

---

## Week 9. Errors and debugging

- Learn about errors and exceptions, which is a great source to understand bugs (why programs fail).
- How to debug, which helps you correct errors.
- Learn simple debugging with errors, which is the majority of problems you face as a beginner.

## Week 10. Object-Oriented Programming

- Learn the power of Object-Oriented Programming (OOP), this is crucial, as all data in Python are objects.
- Why do you need OOP, even though you will never use OOP directly?
- Mastering the basics of OOP that you need to know.

## Week 11. Milestone #2 Project

- Create your own big Object-Oriented Program (OOP).
- First, you create a simple card game as a warmup.
- Then you create the famous casino game BlackJack

## Week 12. Capstone project with ideas

- Working with API, like OpenAI (ChatGPT, GPT.4).
- Create a link scraper that gets all the links from a web page.
- Create your own REST-API
- Create an SQL connector or client.
- Create your own Excel to CSV file converter.
- Create an RSS-feed reader.
- Convert text files to PDF files.

# Python in 12 weeks

**From Zero to Python Hero in Just 12 Weeks:** Mastering Libraries and Frameworks for Web Scraping, PDF Generation, Excel Automation, and More!

- This Python program is **designed for beginners** who want to learn the language from scratch and quickly become proficient in **using libraries and frameworks**.
- In just **12 weeks**, you will learn the basics of Python programming, including syntax, data types, control structures, and functions.
- You will then move on to more advanced topics, such as **object-oriented programming, file handling, web scraping, and data analysis**.
- We will teach you how to use popular Python libraries and frameworks, including **BeautifulSoup**, automate tasks, generate **PDFs**, and manipulate **Excel** files.
- This program includes **hands-on projects** and **real-world examples** that will help you apply your newfound knowledge and build a portfolio of work to showcase your skills.

By the end of the program, you will have the **confidence and expertise to use Python libraries and frameworks** to solve complex problems and build powerful applications.

[Get 67% discount and join](#)



# Make a change in your life

---

- Learning **Python in 12 weeks** requires a commitment to consistent practice and a willingness to challenge yourself.
- By following the above guide, you can **acquire a solid foundation in Python programming** and gain hands-on experience with coding exercises and projects.
- Learning **Python is a valuable investment** in your personal and professional development, offering opportunities to build web applications, analyze and visualize data, and build machine learning models, among other applications.
- With **dedication and perseverance**, you can become proficient in Python and leverage its vast library ecosystem and community resources to build sophisticated applications.
- **Start your journey today** and in just 12 weeks, you'll be surprised at how much you've learned and how far you've come in your Python programming skills.

Rune