

Support our charity and our mission. Donate to freeCodeCamp.org.

DECEMBER 12, 2022 / #PROGRAMMING

What is Programming? A Handbook for Beginners



Estefania Cassingena Navone

Welcome to the amazing world of programming. This is one of the most useful and powerful skills that you can learn and use to make your visions come true.

In this handbook, we will dive into why programming is important, its applications, its basic concepts, and the skills you need to become a successful programmer.

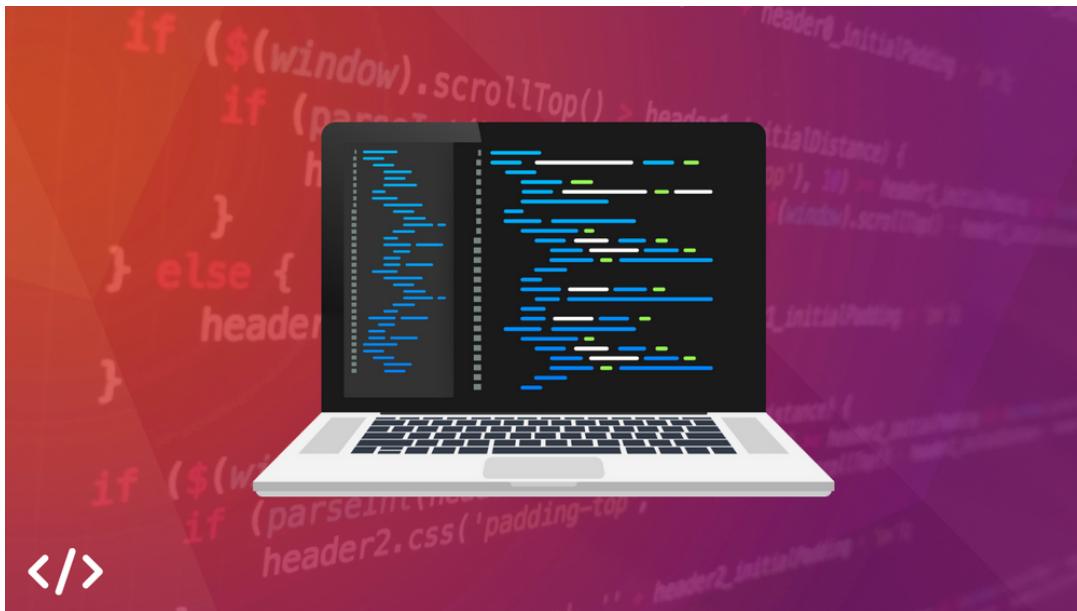
You will learn:

- What programming is and why it is important.
- What a programming language is and why it is important.
- How programming is related to binary numbers.
- Real-world applications of programming.
- Skills you need to succeed as a programmer.
- Tips for learning how to code.
- Basic programming concepts.
- Types of programming languages.

Support our charity and our mission. Donate to freeCodeCamp.org.

Are you ready? Let's begin! ⚡

◆ What is Programming?



Programming is essential for our everyday lives.

Did you know that computer programming is already a fundamental part of your everyday lives? Let's see why. I'm sure that you will be greatly surprised.

Every time you turn on your smartphone, laptop, tablet, smart TV, or any other electronic device, you are running code that was planned, developed, and written by developers. This code creates the final and interactive result that you can see on your screen.

That is exactly what **programming** is all about. It is the process of writing code to solve a particular problem or to implement a particular task.

Support our charity and our mission. Donate to freeCodeCamp.org.

Whenever you check your calendar, attend virtual conferences, browse the web, or edit a document, you are using **code** that has been written by developers.

"And what is code?" you may ask.

Code is a sequence of instructions that a programmer writes to tell a device (like a computer) what to do.

The device cannot know by itself how to handle a particular situation or how to perform a task. So developers are in charge of analyzing the situation and writing explicit instructions to implement what is needed.

To do this, they follow a particular syntax (a set of rules for writing the code).

A **developer** (or programmer) is the person who analyzes a problem and implements a solution in code.

Sounds amazing, right? It's very powerful and **you** can be part this wonderful world too by learning how to code. Let's see how.

You, as a developer.

Let's put you in a developer's shoes for a moment. Imagine that you are developing a mobile app, like the ones that you probably have installed on your smartphone right now.

What is the first thing that you would do?

Support our charity and our mission. Donate to freeCodeCamp.org.

Analyzing the problem. What are you trying to build?

As a developer, you would start by designing the layout of the app, how it will work, its different screens and functionality, and all the small details that will make your app an awesome tool for users around the world.

Only after you have everything carefully planned out, you can start to write your code. To do that, you will need to choose a **programming language** to work with. Let's see what a programming language is and why they are super important.

◆ **What is a Programming Language?**



Logos of popular programming languages.

Support our charity and our mission. Donate to freeCodeCamp.org.

We cannot just write English words in our program like this:

"Computer, solve this task!"

and hope that our computer can understand what we mean. We need to follow certain rules to write the instructions.

Every programming language has its own set of rules that determine if a line of code is valid or not. Because of this, the code you write in one programming languages will be slightly different from others.

 **Tip:** Some programming languages are more complex than others but most of them share core concepts and functionality. If you learn how to code in one programming language, you will likely be able to learn another one faster.

Before you can start writing awesome programs and apps, you need to learn the basic rules of the programming language you chose for the task.

 **Tip:** a **program** is a set of instructions written in a programming language for the computer to execute. We usually write the code for our program in one or multiple files.

For example, this is a line of code in Python (a very popular programming language) that shows the message "Hello, World!":

```
print("Hello, World!")
```

Support our charity and our mission. Donate to freeCodeCamp.org.

To do something very similar in JavaScript, we would write this line of code instead:

```
console.log("Hello, World!");
```

Visually, they look very different, right? This is because Python and JavaScript have a different syntax and a different set of **built-in functions**.

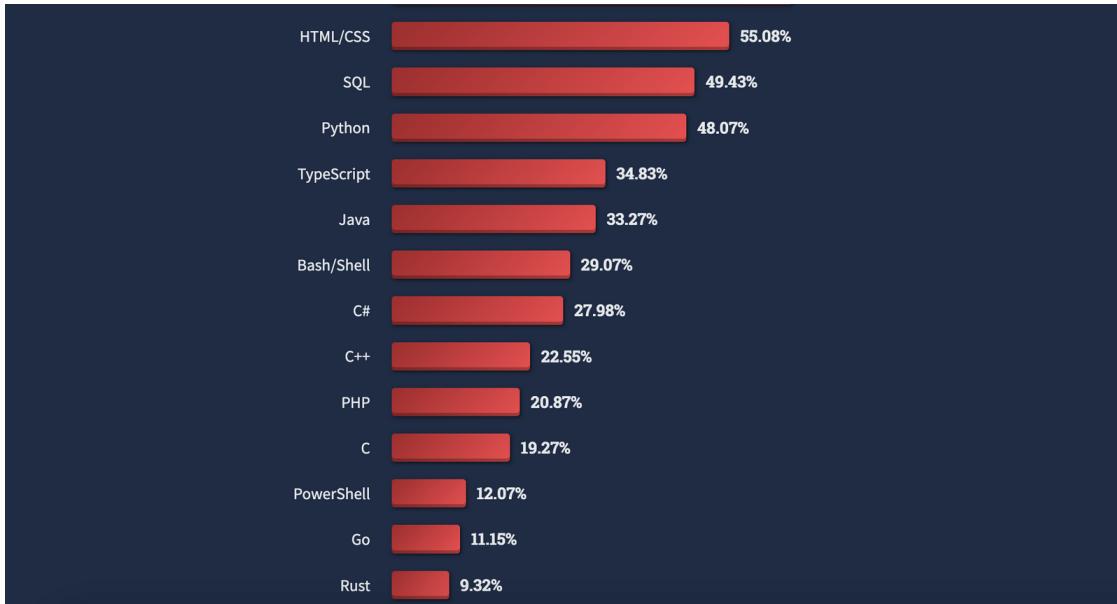
 **Tip:** **built-in functions** are basically tasks that are already defined in the programming language. This lets us use them directly in our code by writing their names and by specifying the values they need.

In our examples, `print()` is a built-in function in Python while `console.log()` is a function that we can use in JavaScript to see the message in the console (an interactive tool) if we run our code in the browser.

Examples of programming languages include Python, JavaScript, TypeScript, Java, C, C#, C++, PHP, Go, Swift, SQL, and R. There are many programming languages and most of them can be used for many different purposes.

 **Tip:** These were the most popular programming languages on the [Stack Overflow Developer Survey 2022](#):

Support our charity and our mission. Donate to freeCodeCamp.org.



The 14 most popular programming languages among all respondents of the [StackOverflow Developer Survey 2022](#). It's a yearly survey that collects information about popular technologies and trends in the developer community.

There are many other programming languages (hundreds or even thousands!) but usually, you will learn and work with some of the most popular ones. Some of them have broader applications like Python and JavaScript while others (like R) have more specific (and even scientific) purposes.

This sounds very interesting, right? And we are only starting to talk about programming languages. There is a lot to learn about them and I promise you that if you dive deeper into programming, your time and effort will be totally worth it.

Awesome! Now that you know what programming is and what programming languages are all about, let's see how programming is related to binary numbers.

Support our charity and our mission. Donate to freeCodeCamp.org.

When you think about programming, perhaps the first thing that comes to your mind is something like the below image, right? A sequence of 0 s and 1 s on your computer.



Binary numbers are 0 and 1.

Programming is indeed related to binary numbers (0 and 1) but in an indirect way. Developers do not actually write their code using zeros and ones.

We usually write programs in a high-level programming language, a programming language with a syntax that recognizes specific words (called keywords), symbols, and values of different data types.

Basically, we write code in a way that humans can understand.

For example, these are the keywords that we can use in Python:

Support our charity and our mission. Donate to freeCodeCamp.org.

and	del	import	return
as	elif	in	try
assert	else	is	while
async	except	lambda	with
await	finally	nonlocal	yield
break	for	not	

List of Python keywords.

Every programming language has its own set of keywords (words written in English). These keywords are part of the syntax and core functionality of the programming language.

But keywords are just common words in English, almost like the ones that we would find in a book.

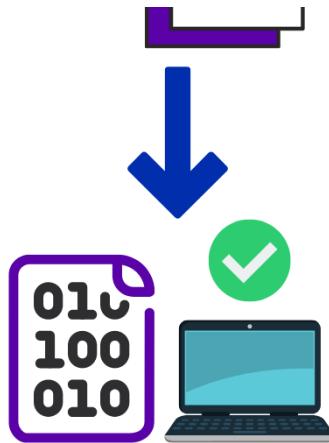
That leads us to two very important questions:

- How does the computer understand and interpret what we are trying to say?
- Where does the binary number system come into play here?

The computer does **not** understand these words, symbols, or values directly.

When a program runs, the code that we write in a high-level programming language that humans can understand is automatically transformed into binary code that the computer can understand.

Support our charity and our mission. Donate to freeCodeCamp.org.



Process of transforming a program into binary code.

This transformation of source code that humans can understand into binary code that the computer can understand is called **compilation**.

According to [Britannica](#), a **compiler** is defined as:

Computer software that translates (compiles) source code written in a high-level language (e.g., C++) into a set of machine-language instructions that can be understood by a digital computer's CPU.

Britannica also mentions that:

The term *compiler* was coined by American computer scientist [Grace Hopper](#), who designed one of the first compilers in the early 1950s.

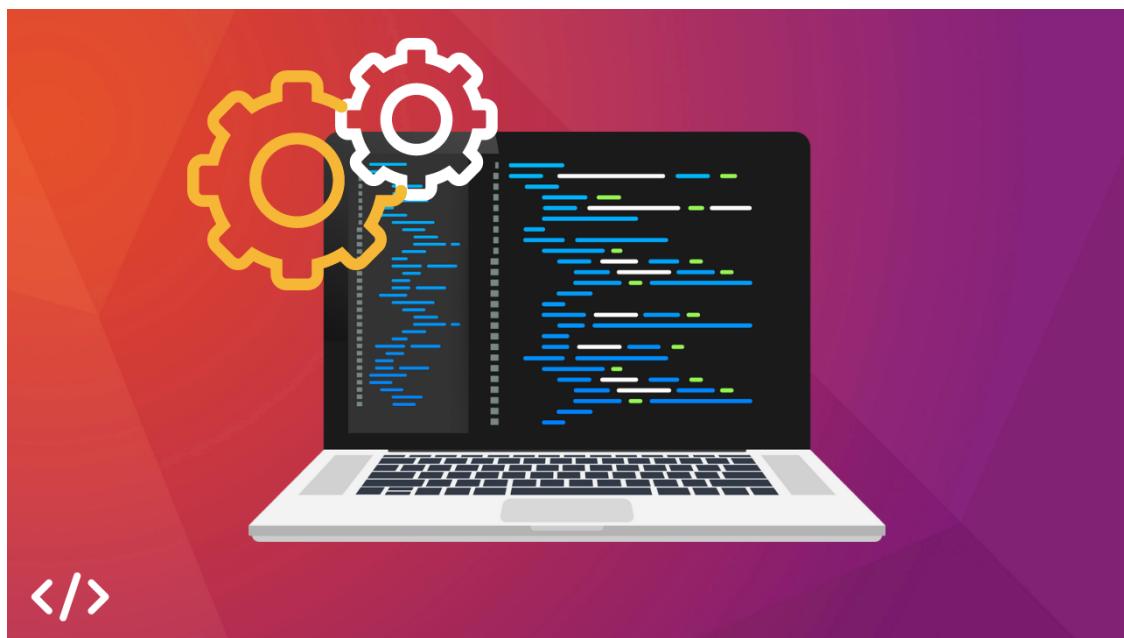
Some programming languages can be classified as **compiled** programming languages while others can be classified as **interpreted** programming languages based on how they are transformed into machine-language instructions.

Support our charity and our mission. Donate to freeCodeCamp.org.

Awesome. Now you know why binary code is so important for computer science. Without it, basically programming would not exist because computers would not be able to understand our instructions.

Now let's dive into the applications of programming and the different areas that you can explore.

◆ **Real-World Applications of Programming**



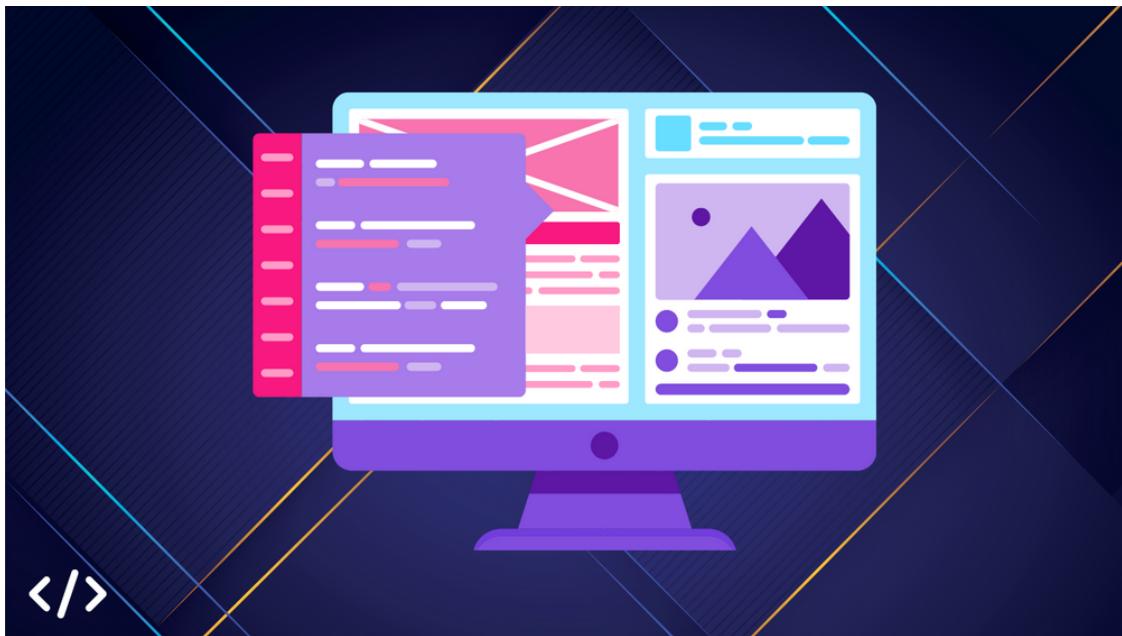
Programming has many real-world applications in basically every industry that you can imagine.

Programming has many different applications in many different industries. This is truly amazing because you can apply your knowledge in virtually any industry that you are interested in.

Support our charity and our mission. Donate to freeCodeCamp.org.

Let's see some of them. (I promise you. They are amazing! ⭐).

Front-End Web Development



Front-End Web Developers develop the parts of websites and web applications that users can see and interact with.

If you learn how to code, you can use your programming skills to design and develop websites and online platforms. Front-End Web Developers create the parts of the websites that users can see and interact with directly.

For example, right now you are reading an article on [freeCodeCamp's](https://freeCodeCamp.org) publication. The publication looks like this and it works like this thanks to code that front-end web developers wrote line by line.

 **Tip:** If you learn front-end web development, you can do this too.

Support our charity and our mission. Donate to freeCodeCamp.org.

The screenshot shows the freeCodeCamp.org homepage. On the left, there are two main sections: "Build projects." and "Earn certifications.". Below these are logos for Apple, Google, Microsoft, Spotify, and Amazon.com. A yellow button labeled "Get started (it's free)" is visible. On the right, the Chrome Developer Tools' Elements tab is open, displaying the HTML and CSS code for the page. The CSS code includes styles for Bootstrap and custom styles for the footer and page wrapper.

The HTML and CSS code for freeCodeCamp's Home Page (This is a preview of the code in Chrome Developer Tools).

Front-End Web Developers use HTML and CSS to create the structure of the website (these are markup languages, which are used to present information) and they write JavaScript code to add functionality and interactivity.

If you are interested in learning front-end web development, you can learn HTML and CSS with these free courses on freeCodeCamp's YouTube Channel:

- [Learn HTML5 and CSS3 From Scratch - Full Course](#)
- [Learn HTML & CSS - Full Course for Beginners](#)
- [Frontend Web Development Bootcamp Course \(JavaScript, HTML, CSS\)](#)
- [Introduction To Responsive Web Design - HTML & CSS Tutorial](#)

You can also learn JavaScript for free with these free online courses:

- [Learn JavaScript - Full Course for Beginners](#)

Support our charity and our mission. Donate to freeCodeCamp.org.

- [Learn JavaScript by Building 7 Games - Full Course](#)

 **Tip:** You can also earn a [Responsive Web Design Certification](#) while you learn with interactive exercises on freeCodeCamp.

Back-End Web Development



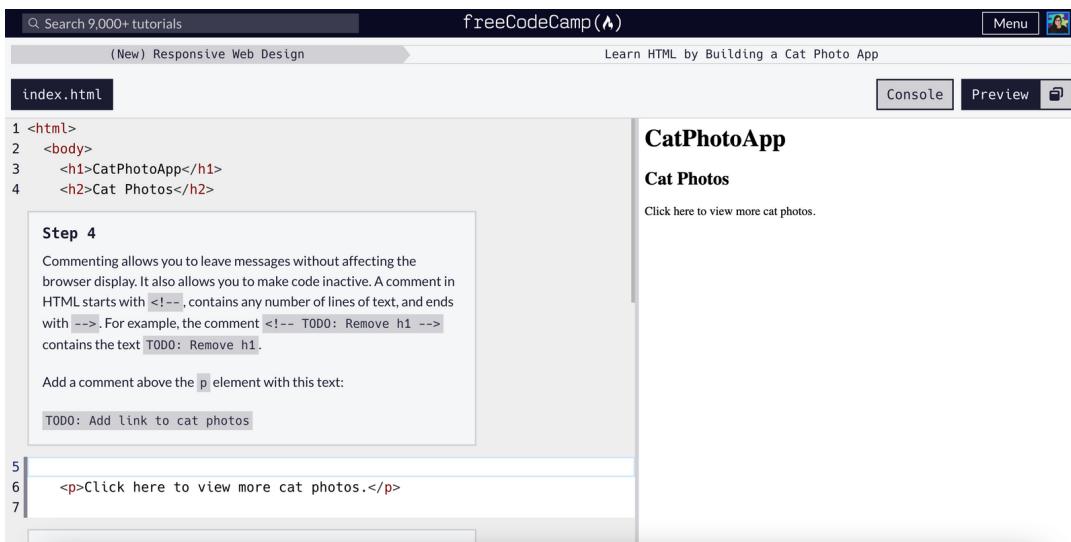
Back-End Web Developers develop servers and databases to handle everything that runs behind the scenes to make more complex web applications work correctly.

More complex and dynamic web applications that work with user data also require a **server**. This is a computer program that receives requests and sends appropriate responses. They also need a **database**, a collection of values stored in a structured way.

Back-End Web Developers are in charge of developing the code for these servers. They decide how to handle the different requests, how to send appropriate resources, how to store the information,

Support our charity and our mission. Donate to freeCodeCamp.org.

A real-world example of back-end web development is what happens when you create an account on freeCodeCamp and complete a challenge. Your information is stored on a database and you can access it later when you sign in with your email and password.



A freeCodeCamp interactive challenge.

This amazing interactive functionality was implemented by back-end web developers.

 **Tip:** Full-stack Web Developers are in charge of both Front-End and Back-End Web Development. They have specialized knowledge on both areas.

All the complex platforms that you use every day, like social media platforms, online shopping platforms, and educational platforms, use servers and back-end web development to power their amazing functionality.

Support our charity and our mission. Donate to [freeCodeCamp.org](https://www.freecodecamp.org/donate).

This is partly because it is simple and easy to learn and yet powerful and versatile enough to be used in real-world applications.

 **Tip:** if you are curious about the specific applications of Python, this is [an article I wrote on this topic](#).

JavaScript can also be used for back-end web development thanks to Node.js.

Other programming languages used to develop web servers are PHP, Ruby, C#, and Java.

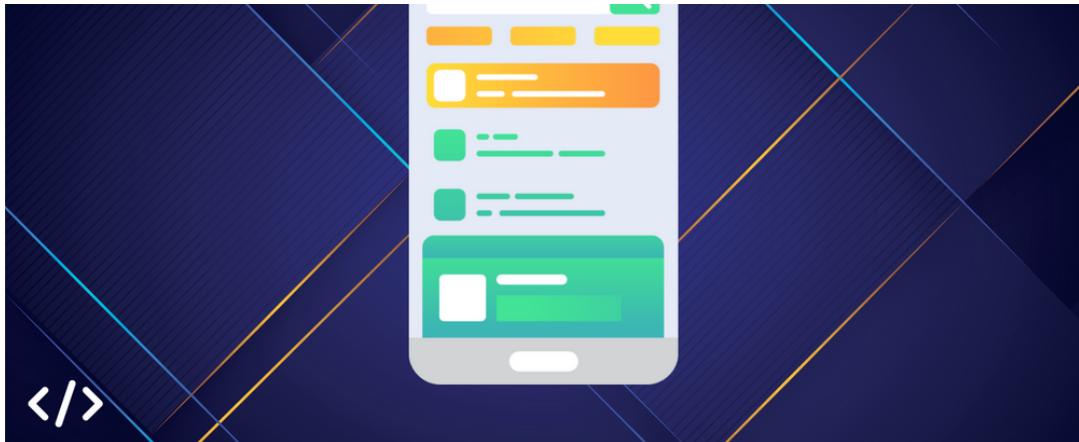
If you would like to learn Back-End Web Development, these are free courses on freeCodeCamp's YouTube channel:

- [Python Backend Web Development Course \(with Django\)](#)
- [Node.js and Express.js - Full Course](#)
- [Full Stack Web Development for Beginners \(Full Course on HTML, CSS, JavaScript, Node.js, MongoDB\)](#)
- [Node.js / Express Course - Build 4 Projects](#)

 **Tip:** freeCodeCamp also has a free [Back End Development and APIs certification](#).

Mobile App Development

Support our charity and our mission. Donate to freeCodeCamp.org.



Mobile app developers design and develop the mobile apps we use every day.

Mobile apps have become part of our everyday lives. I'm sure that you could not imagine life without them.

Think about your favorite mobile app. What do you love about it?

Our favorite apps help us with our daily tasks, they entertain us, they solve a problem, and they help us to achieve our goals. They are always there for us.

That is the power of mobile apps and you can be part of this amazing world too if you learn mobile app development.

Developers focused on mobile app development are in charge of planning, designing, and developing the user interface and functionality of these apps. They identify a gap in the existing apps and they try to create a working product to make people's lives better.

 **Tip:** regardless of the field you choose, your goal as a developer should always be making people's lives better. Apps are not just apps, they have the potential to change our lives. You should always

Support our charity and our mission. Donate to freeCodeCamp.org.

Mobile app developers use programming languages like JavaScript, Java, Swift, Kotlin, and Dart. Frameworks like [Flutter](#) and [React Native](#) are super helpful to build cross-platform mobile apps (that is, apps that run smoothly on multiple different operating systems like Android and iOS).

According to [Flutter's](#) official documentation:

Flutter is an open source framework by Google for building beautiful, natively compiled, multi-platform applications from a single codebase.

If you would like to learn mobile app development, these are free courses that you can take on freeCodeCamp's YouTube channel:

- [Flutter Course for Beginners – 37-hour Cross Platform App Development Tutorial](#)
- [Flutter Course - Full Tutorial for Beginners \(Build iOS and Android Apps\)](#)
- [React Native - Intro Course for Beginners](#)
- [Learn React Native Gestures and Animations - Tutorial](#)

Game Development

Support our charity and our mission. Donate to freeCodeCamp.org.



Games create long-lasting memories. I'm sure that you still remember your favorite games and why you love (or loved) them so much. Being a game developer means having the opportunity of bringing joy and entertainment to players around the world.

Game developers envision, design, plan, and implement the functionality of a game. They also need to find or create assets such as characters, obstacles, backgrounds, music, sound effects, and more.

 **Tip:** if you learn how to code, you can create your own games. Imagine creating an awesome and engaging game that users around the world will love. That is what I personally love about programming. You only need your computer, your knowledge, and some basic tools to create something amazing.

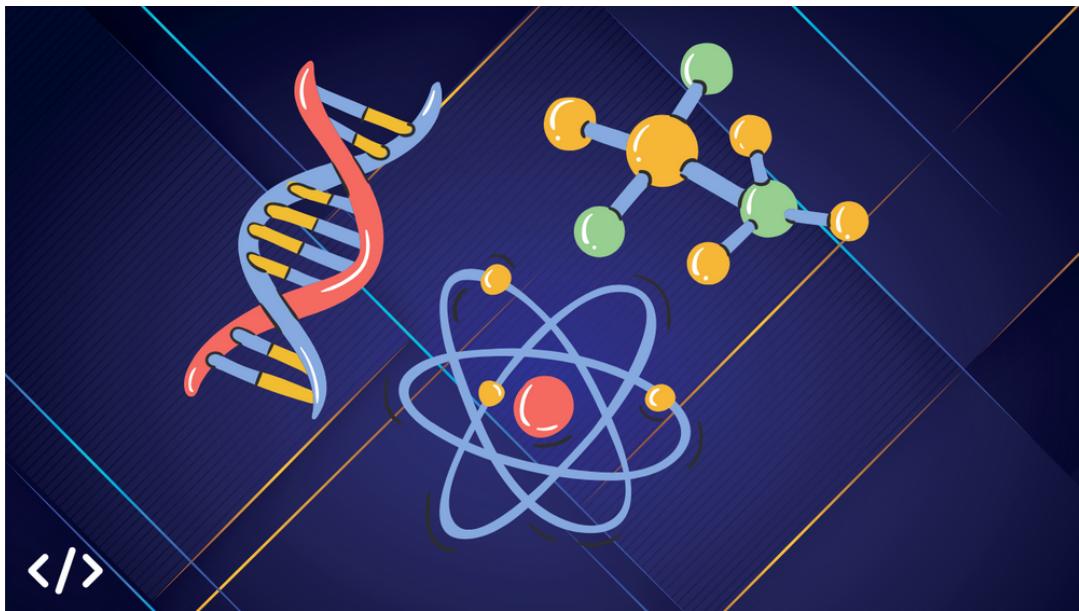
Popular programming languages used for game development include JavaScript, C++, Python, and C#.

If you are interested in learning game development, you can take these free courses on freeCodeCamp's YouTube channel:

Support our charity and our mission. Donate to freeCodeCamp.org.

- [Learn Unity - Beginner's Game Development Tutorial](#)
- [Learn Python by Building Five Games - Full Course](#)
- [Code a 2D Game Using JavaScript, HTML, and CSS \(w/ Free Game Assets\) – Tutorial](#)
- [2D Game Development with GDevelop - Crash Course](#)
- [Pokémon Coding Tutorial - CS50's Intro to Game Development](#)

Biology, Physics, and Chemistry



Programming can be applied in every scientific field that you can imagine, including biology, physics, chemistry, and even astronomy. Yes! Scientists use programming all the time to collect and analyze data. They can even run simulations to test hypotheses.

Biology

Support our charity and our mission. Donate to [freeCodeCamp.org](https://www.freecodecamp.org/donate).

According to this [article](#) "Bioinformatics" by Ardeshir Bayat, member of the Centre for Integrated Genomic Medical Research at the University of Manchester:

Bioinformatics is defined as the application of tools of computation and analysis to the capture and interpretation of biological data.

Dr. Bayat mentions that bioinformatics can be used for genome sequencing. He also mentions that its discoveries may lead to drug discoveries and individualized therapies.

Frequently used programming languages for bioinformatics include Python, R, PHP, PERL, and Java.

 **Tip:** R is a programming "language and environment for statistical computing and graphics" ([source](#)).

An example of a great tool that scientists can use for biology is [Biopython](#). This is a Python framework with "freely available tools for biological computation."

If you would like to learn more about how you can apply your programming skills in science, these are free courses that you can take on freeCodeCamp's YouTube channel:

- [Python for Bioinformatics - Drug Discovery Using Machine Learning and Data Analysis](#)
- [R Programming Tutorial - Learn the Basics of Statistical Computing](#)

Support our charity and our mission. Donate to freeCodeCamp.org.

Physics

Physics requires running many simulations and programming is perfect for doing exactly that. With programming, scientists can program and run simulations based on specific scenarios that would be hard to replicate in real life. This is much more efficient.

Programming languages that are commonly used for physics simulations include C, Java, Python, MATLAB, and JavaScript.

Chemistry

Chemistry also relies on simulations and data analysis, so it's a field where programming can be a very helpful tool.

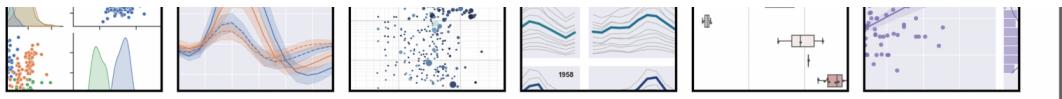
In [this scientific article](#) by Dr. Ivar Ugi and his colleagues from Organisch-chemisches Institut der Technischen Universität München, they mention that:

The design of entirely new syntheses, and the classification and documentation of structures, substructures, and reactions are examples of new applications of computers to chemistry.

Scientific experiments also generate detailed data and results that can be analyzed with computer programs developed by scientists.

Think about it: writing a program to generate a box plot or a scatter plot or any other type of plot to visualize trends in thousands of measurements can save researchers a lot of time and effort. This lets them focus on the most important part of their work: analyzing the results.

Support our charity and our mission. Donate to freeCodeCamp.org.



Example of data visualizations that you can create with [Seaborn](#), a Python data visualization library. This is very helpful to analyze data, right?

 **Tips:** if you are interested in diving deeper into this, this is a [list of chemistry simulations](#) by the American Chemical Society. These simulations were programmed by developers and they are helping thousands of students and teachers around the world.

Think about it... You could build the next great simulation. If you are interested in a scientific field, I totally recommend learning how to code. Your work will be much more productive and your results will be easier to analyze.

If you are interested in learning programming for scientific applications, these are free courses on freeCodeCamp's YouTube channel:

- [Python for Bioinformatics - Drug Discovery Using Machine Learning and Data Analysis](#)
- [Python for Data Science - Course for Beginners \(Learn Python, Pandas, NumPy, Matplotlib\)](#)

Data Science and Engineering

Support our charity and our mission. Donate to freeCodeCamp.org.



Talking about data...programming is also essential for a field called **Data Science**. If you are interested in answering questions through data and statistics, this field might be exactly what you are looking for and having programming skills will help you to achieve your goals.

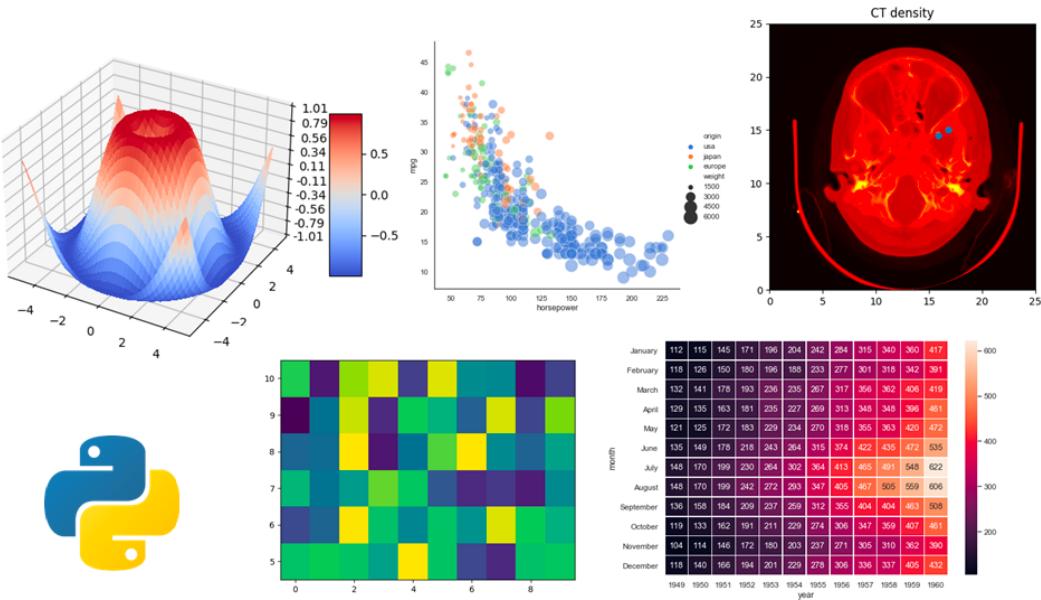
Data scientists collect and analyze data in order to answer questions in many different fields. According to UC Berkeley in the article "[What is Data Science?](#)":

Effective data scientists are able to identify relevant questions, collect data from a multitude of different data sources, organize the information, translate results into solutions, and communicate their findings in a way that positively affects business decisions.

There are many powerful programming languages for analyzing and visualizing data, but perhaps one of the most frequently used ones for this purpose is Python.

This is an example of the type of data visualizations that you can create with Python. They are very helpful to analyze data visually

Support our charity and our mission. Donate to freeCodeCamp.org.



Sample data visualizations from the [Matplotlib](#) and [Seaborn](#) galleries

If you are interested in learning programming for data science, these are free courses on freeCodeCamp's YouTube channel:

- [Learn Data Science Tutorial - Full Course for Beginners](#)
- [Intro to Data Science - Crash Course for Beginners](#)
- [Python for Data Science - Course for Beginners \(Learn Python, Pandas, NumPy, Matplotlib\)](#)
- [Build 12 Data Science Apps with Python and Streamlit - Full Course](#)
- [Data Analysis with Python - Full Course for Beginners \(Numpy, Pandas, Matplotlib, Seaborn\)](#)

Tip: you can also earn these free certifications on freeCodeCamp:

- [Data Visualization](#)

Support our charity and our mission. Donate to freeCodeCamp.org.

Engineering

Engineering is another field where programming can help you to succeed. Being able to write your own computer programs can make your work much more efficient.

There are many tools created specifically for engineers. For example, the R programming language is specialized in statistical applications and Python is very popular in this field too.

Another great tool for programming in engineering is MATLAB. According to its official website:

MATLAB is a programming and numeric computing platform used by millions of engineers and scientists to analyze data, develop algorithms, and create models.

Really, the possibilities are endless.

You can learn MATLAB with [this crash course on the freeCodeCamp YouTube channel](#).

If you are interested in learning engineering tools related to programming, this is a free course on freeCodeCamp's YouTube channel that covers AutoCAD, a 2D and 3D computer-aided design software used by engineers:

- [AutoCAD for Beginners - Full University Course](#)

Medicine and Pharmacology

Support our charity and our mission. Donate to freeCodeCamp.org.



Programming has helped scientists to develop new medical techniques and devices.

Medicine and pharmacology are constantly evolving by finding new treatments and procedures. Let's see how you can apply your programming skills in these fields.

Medicine

Programming is really everywhere. If you are interested in the field of medicine, learning how to code can be very helpful for you too. Even if you would like to focus on computer science and software development, you can apply your knowledge in both fields.

Specialized developers are in charge of developing and writing the code that powers and controls the devices and machines that are used by modern medicine.

Think about it...all these machines and devices are controlled by software and someone has to write that software. Medical records are also stored and tracked by specialized systems created by developers. That could be you if you decide to follow this path. Sounds exciting, right?

Support our charity and our mission. Donate to freeCodeCamp.org.

Major uses of computers in medicine include hospital information system, data analysis in medicine, medical imaging laboratory computing, computer assisted medical decision making, care of critically ill patients, computer assisted therapy and so on.

Pharmacology

Programming and computer science can also be applied to develop new drugs in the field of pharmacology.

A remarkable example of what you can achieve in this field by learning how to code is presented in [this article](#) by MIT News. It describes how an MIT senior, Kristy Carpenter, was using computer science in 2019 to develop "new, more affordable drugs." Kristy mentions that:

Artificial intelligence, which can help compute the combinations of compounds that would be better for a particular drug, can reduce trial-and-error time and ideally quicken the process of designing new medicines.

Another example of a real-world application of programming in pharmacology is related to Python (yes, Python has many applications!). Among its [success stories](#), we find that Python was selected by AstraZeneca to develop techniques and programs that can help scientists to discover new drugs faster and more efficiently.

The [documentation](#) explains that:

To save time and money on laboratory work, experimental chemists use computational models to narrow the field of

Support our charity and our mission. Donate to [freeCodeCamp.org](https://www.freecodecamp.org/donate).

If you are interested in learning programming for medicine or health-related fields, this is a free course on freeCodeCamp's YouTube channel on programming for healthcare imaging:

- [PyTorch and Monai for AI Healthcare Imaging - Python Machine Learning Course](#)

Education



Programming can be used to create tools that help teachers and students to have a more productive and engaging learning experience. Teaching students to code also develops their problem-solving skills.

Have you ever thought that programming could be helpful for education? Well, let me tell you that it is and it is very important. Why? Because the digital learning tools that students and teachers use nowadays are programmed by developers.

Support our charity and our mission. Donate to freeCodeCamp.org.

programming is making that possible.

As a programmer or as a teacher who knows how to code, you can create the next great app that will enhance the learning experience of students around the world.

Perhaps it will be a note-taking app, an online learning platform, a presentation app, an educational game, or any other app that could be helpful for students.

The important thing is to create it with students in mind if your goal is to make something amazing that will create long-lasting memories.

If you envision it, then you can create it with code.

Teachers can also teach their students how to code to develop their problem-solving skills and to teach them important skills for their future.

 **Tip:** if you are teaching students how to code, **Scratch** is a great programming language to teach the basics of programming. It is particularly focused on teaching children how to code in an interactive way.

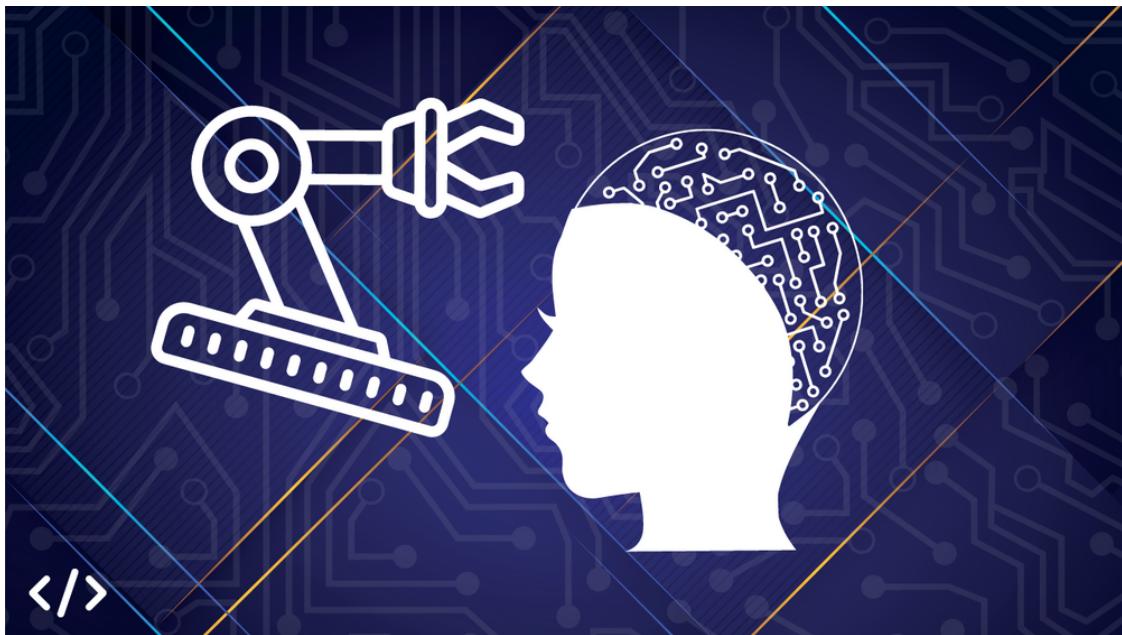
According to the official [Scratch](#) website:

Scratch is the world's largest coding community for children and a coding language with a simple visual interface that allows young people to create digital stories, games, and animations.

Support our charity and our mission. Donate to freeCodeCamp.org.

- [Scratch Tutorial for Beginners - Make a Flappy Bird Game](#)
- [Computational Thinking & Scratch - Intro to Computer Science - Harvard's CS50 \(2018\)](#)
- [Android Development for Beginners - Full Course](#)
- [Flutter Course for Beginners – 37-hour Cross Platform App Development Tutorial](#)
- [Learn Unity - Beginner's Game Development Tutorial](#)

Machine Learning, Artificial Intelligence, and Robotics



Machine Learning and Artificial Intelligence are very popular nowadays because platforms can learn how users engage with their content in order to suggest them relevant information and products.

Some of the most amazing fields that are directly related to programming are Machine Learning, Artificial Intelligence, and

Support our charity and our mission. Donate to freeCodeCamp.org.

The project of developing systems endowed with the intellectual processes characteristic of humans, such as the ability to reason, discover meaning, generalize, or learn from past experience.

Machine learning is a branch or a subset of the field of Artificial Intelligence in which systems can learn on their own based on data. The goal of this learning process is to predict the expected output. These models continuously learn how to "think" and how to analyze situations based on their previous training.

The most commonly used programming languages in these fields are Python, C, C#, C++, and MATLAB.

Artificial intelligence and Machine Learning have amazing applications in various industries, such as:

- Image and object detection.
- Making predictions based on patterns.
- Text recognition.
- Recommendation engines (like when an online shopping platform shows you products that you may like or when YouTube shows you videos that you may like).
- Spam detection for emails.
- Fraud detection.
- Social media features like personalized feeds.
- Many more... there are literally millions of applications in virtually every industry.

Support our charity and our mission. Donate to [freeCodeCamp.org](https://www.freecodecamp.org/donate).

- [Machine Learning for Everybody – Full Course](#)
- [Machine Learning Course for Beginners](#)
- [PyTorch for Deep Learning & Machine Learning – Full Course](#)
- [TensorFlow 2.0 Complete Course - Python Neural Networks for Beginners Tutorial](#)
- [Self-Driving Car with JavaScript Course – Neural Networks and Machine Learning](#)
- [Python TensorFlow for Machine Learning – Neural Network Text Classification Tutorial](#)
- [Practical Deep Learning for Coders - Full Course from fast.ai and Jeremy Howard](#)
- [Deep Learning Crash Course for Beginners](#)
- [Advanced Computer Vision with Python - Full Course](#)

 **Tip:** you can also earn a [Machine Learning with Python](#) Certification on freeCodeCamp.

Robotics

Programming is also very important for robotics. Yes, robots are programmed too!

Robotics is defined by [Britannica](#) as the:

Design, construction, and use of machines (robots) to perform tasks done traditionally by human beings.

Support our charity and our mission. Donate to [freeCodeCamp.org](https://www.freecodecamp.org/donate).
, -----, -----
machinery found in manufacturing facilities.

If you are interested in learning how to code for robotics, electronics, and related fields, this is a free course on Arduino on freeCodeCamp's YouTube channel:

- [Arduino Course for Beginners - Open-Source Electronics Platform](#)

Other Applications

There are many other fascinating applications of programming in almost every field. These are some highlights:

- **Agriculture:** in [this article](#) by MIT News, a farmer developed an autonomous tractor app after learning how to code.
- **Self-driving cars:** autonomous cars rely on software to analyze their surroundings and to make quick and accurate decisions on the road. If you are interested in this area, [this is a course on this topic](#) on freeCodeCamp's YouTube channel.
- **Finance:** programming can also be helpful to develop programs and models that predict financial indicators and trends. For example, this is a [course on algorithmic trading](#) on freeCodeCamp's YouTube channel.

The possibilities are endless. I hope that this section will give you a notion of why learning how to code is so important for your present and for your future. It will be a valuable skill to have in any field you choose.

Support our charity and our mission. Donate to freeCodeCamp.org.

◆ Skills of a Successful Programmer



After going through the diverse range of applications of programming, you must be curious to know what skills are needed to succeed in this field.

Curiosity

A programmer should be curious. Whether you are just starting to learn how to code or you already have 20 years of experience, coding projects will always present you with new challenges and learning opportunities. If you take these opportunities, you will continuously improve your skills and succeed.

Enthusiasm

Support our charity and our mission. Donate to freeCodeCamp.org.

 **Tip:** If you ever feel like you are not as enthusiastic as you used to be, it's time to find or learn something new that can light the spark in you again and fill you with hope and dreams.

Patience

A programmer must be patient because transforming an initial idea into a working product can take time, effort, and many different steps. Patience will keep you focused on your final goal.

Resilience

Programming can be challenging. That is true. But what defines you is not how many challenges you face, it's how you face them. If you thrive despite these challenges, you will become a better programmer and you could create something that could change the world.

Creativity

Programmers must be creative because even though every programming language has a particular set of rules for writing the code, coding is like using LEGOs. You have the building-blocks but you need to decide what to create and how to create it. The process of writing the code requires creativity while following the established best practices.

Problem-solving and Analysis

Programming is basically analyzing and solving problems with code. Depending on your field of choice, those problems will be simpler or

Support our charity and our mission. Donate to freeCodeCamp.org.

Questions like:

- What should I build?
- How can I build it?
- What is the best way to build this?

Are part of the everyday routine of a programmer.

Ability to Focus for Long Periods of Time

When you are working on a coding project, you will need to focus on a task for long periods of time. From creating the design, to planning and writing the code, to testing the result, and to fixing bugs (issues with the code), you will dedicate many hours to a particular task. This is why it's essential to be able to focus and to keep your final goal in mind.

Taking Detailed Notes

This skill is very important for programmers, particularly when you are learning how to code. Taking detailed notes can help you to understand and remember the concepts and tools you learn. This also applies for experienced programmers, since being a programmer involves life-long learning.

Communication

Initially, you might think that programming is a solitary activity and imagine that a programmer spends hundreds of hours alone sitting on a desk.

Support our charity and our mission. Donate to freeCodeCamp.org.

Open to Feedback

In programming, there is usually more than one way to implement the same functionality. Different alternatives may work similarly, but some may be easier to read or more efficient in terms of time or resource consumption.

When you are learning how to code, you should always take constructive feedback as a tool for learning. Similarly, when you are working on a team, take your colleagues' feedback positively and always try to improve.

Life-long Learning

Programming equals life-long learning. If you are interested in learning how to code, you must know that you will always need to be learning new things as new technologies emerge and existing technologies are updated. Think about it... that is great because there is always something interesting and new to learn!

Open to Trying New Things

Finally, an essential skill to be a successful programmer is to be open to trying new things. Step out of your comfort zone and be open to new technologies and products. In the technology industry, things evolve very quickly and adapting to change is essential.

◆ Tips for Learning How to Code

Support our charity and our mission. Donate to freeCodeCamp.org.



Now that you know more about programming, programming languages, and the skills you need to be a successful programmer, let's see some tips for learning how to code.

 **Tip:** these tips are based on my personal experience and opinions.

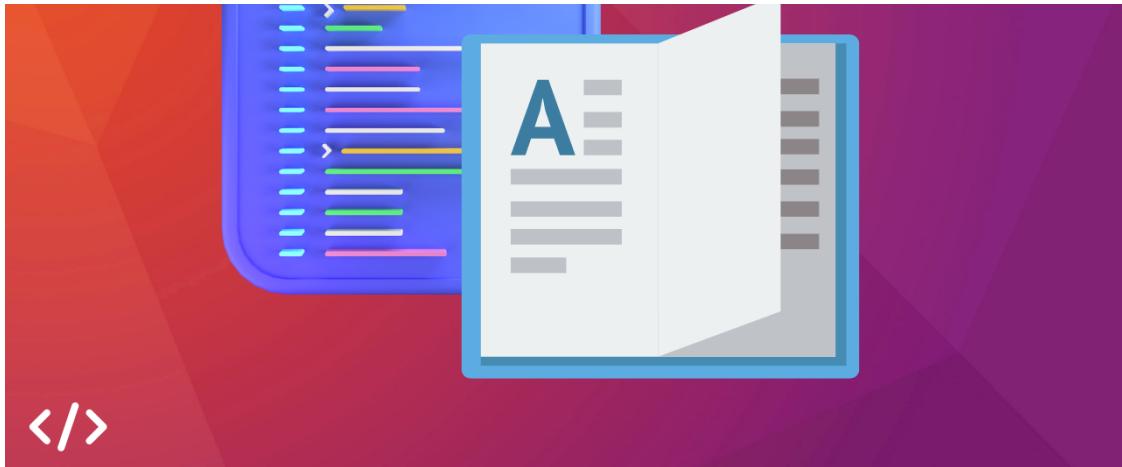
- **Choose one programming language to learn first.** When you are learning how to code, it's easy to feel overwhelmed with the number of options and entry paths. My advice would be to focus on understanding the essential computer science concepts and one programming language first. Python and JavaScript are great options to start learning the fundamentals.
- **Take detailed notes.** Note-taking skills are essential to record and to analyze the topics you are learning. You can add custom comments and annotations to explain what you are learning.
- **Practice constantly.** You can only improve your problem-solving skills by practicing and by learning new techniques and tools. Try to practice every day.

Support our charity and our mission. Donate to freeCodeCamp.org.

- **Always try again.** If you can't solve a problem on your first try, take a break and come back again and again until you solve it. That is the only way to learn. Learn from your mistakes and learn new approaches.
- **Learn how to research and how to find answers.** Programming languages, libraries, and frameworks usually have official documentations that explain their built-in elements and tools and how you can use them. This is a precious resource that you should definitely refer to.
- **Browse Stack Overflow.** This is an amazing platform. It is like an online encyclopedia of answers to common programming questions. You can find answers to existing questions and ask new questions to get help from the community.
- **Set goals.** Motivation is one of the most important factors for success. Setting goals is very important to keep you focused, motivated, and enthusiastic. Once you reach your goals, set new ones that you find challenging and exciting.
- **Create projects.** When you are learning how to code, applying your skills will help you to expand your knowledge and remember things better. Creating projects is the perfect way to practice and to create a portfolio that you can show to potential employers.

◆ Basic Programming Concepts

Support our charity and our mission. Donate to freeCodeCamp.org.



Great. If reading this article has helped you confirm that you want to learn programming, let's take your first steps.

These are some basic programming concepts that you should know:

- **Variable:** a variable is a name that we assign to a value in a computer program. When we define a variable, we assign a value to a name and we allocate a space in memory to store that value. The value of a variable can be updated during the program.
- **Constant:** a constant is similar to a variable. It stores a value but it cannot be modified. Once you assign a value to a constant, you cannot change it during the entire program.
- **Conditional:** a conditional is a programming structure that lets developers choose what the computer should do based on a condition. If the condition is True, something will happen but if the condition is False, something different can happen.
- **Loop:** a loop is a programming structure that let us run a code block (a sequence of instructions) multiple times. They

Support our charity and our mission. Donate to freeCodeCamp.org.

- **Function:** a function helps us to avoid code repetition and to reuse our code. It is like a code block to which we assign a name but it also has some special characteristics. We can write the name of the function to run that sequence of instructions without writing them again.

 **Tip:** Functions can communicate with main programs and main programs can communicate with functions through **parameters**, **arguments**, and **return statements**.

- **Class:** a class is used as a blueprint to define the characteristics and functionality of a type of object. Just like we have objects in our real world, we can represent objects in our programs.
- **Bug:** a bug is an error in the logic or implementation of a program that results in an unexpected or incorrect output.
- **Debugging:** debugging is the process of finding and fixing bugs in a program.
- **IDE:** this acronym stands for Integrated Development Environment. It is a software development environment that has the most helpful tools that you will need to write computer programs such as a file editor, an explorer, a terminal, and helpful menu options.

 **Tip:** a commonly used and free IDE is [Visual Studio Code](#), created by Microsoft.

Awesome! Now you know some of the fundamental concepts in programming. Like you learned, each programming language has a

Support our charity and our mission. Donate to freeCodeCamp.org.

◆ **Types of Programming Languages**



Programming languages can be classified based on different criteria. If you want to learn how to code, it's important for you to learn these basic classifications:

Complexity

- **High-level programming languages:** they are designed to be understood by humans and they have to be converted into machine code before the computer can understand them. They are the programming languages that we commonly use. For example: JavaScript, Python, Java, C#, C++, and Kotlin.
- **Low-level programming languages:** they are more difficult to understand because they are not designed for humans.

Support our charity and our mission. Donate to freeCodeCamp.org.

Conversion into Machine Code

- **Compiled programming languages:** programs written with this type of programming language are converted directly into machine code by a compiler. Examples include C, C++, Haskell, and Go.
- **Interpreted programming languages:** programs written with this type of programming language rely on another program called the **interpreter**, which is in charge of running the code line by line. Examples include Python, JavaScript, PHP, and Ruby.



Tip: according to [this article](#) on freeCodeCamp's publication:

Most programming languages can have both compiled and interpreted implementations – the language itself is not necessarily compiled or interpreted. However, for simplicity's sake, they're typically referred to as such.

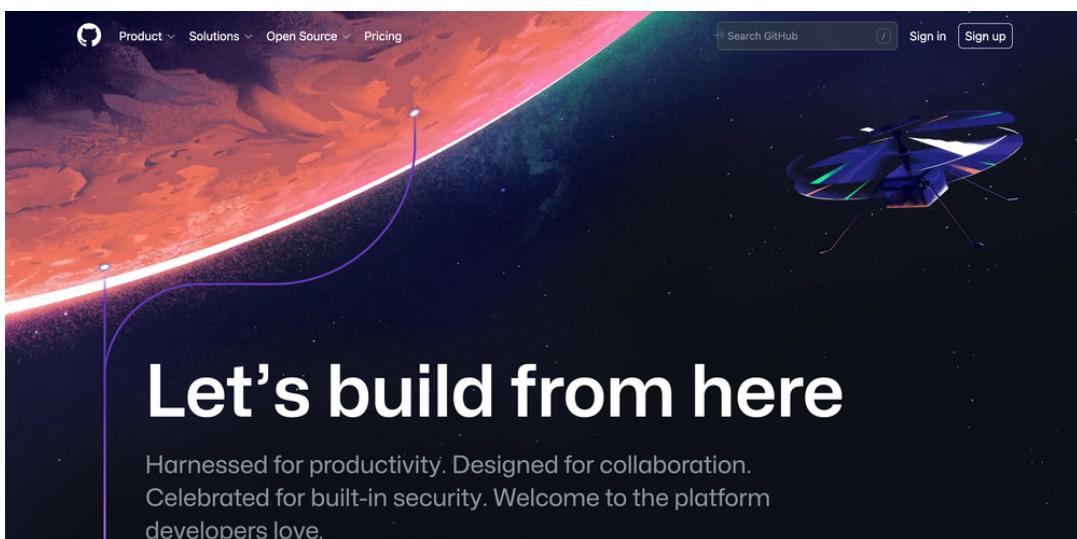
There are other types of programming languages based on different criteria, such as:

- Procedural programming languages
- Functional programming languages
- Object-oriented programming languages
- Scripting languages
- Logic programming languages

Support our charity and our mission. Donate to freeCodeCamp.org.

project.

◆ How to Contribute to Open Source Projects



GitHub's Home Page.

Finally, you might think that coding implies sitting at a desk for many hours looking at your code without any human interaction. But let me tell you that this does not have to be true at all. You can be part of a learning community or a developer community.

Initially, when you are learning how to code, you can participate in a learning community like freeCodeCamp. This way, you will share your journey with others who are learning how to code, just like you.

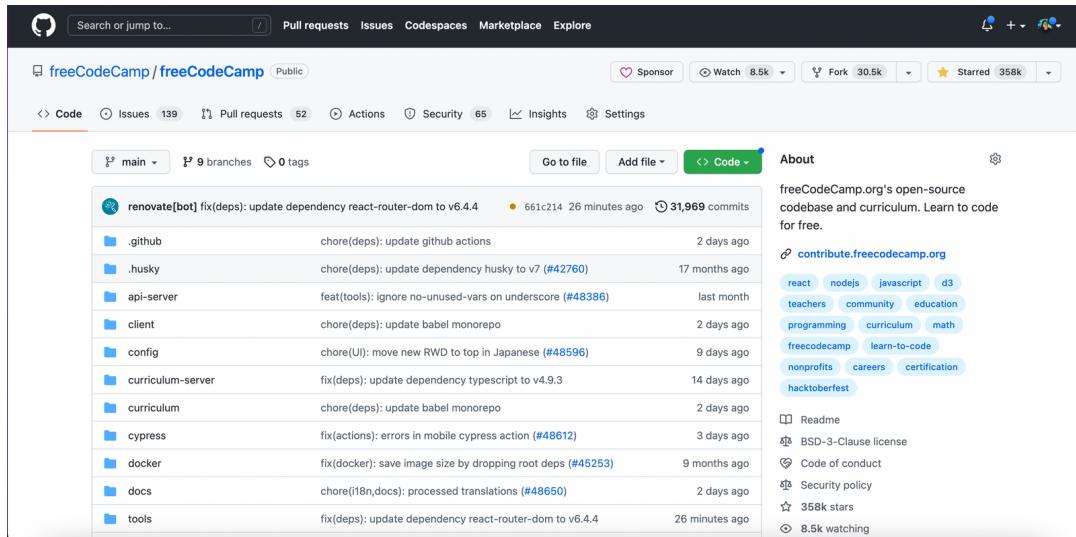
Then, when you have enough skills and confidence in your knowledge, you can practice by contributing to open source projects and join developer communities.

Support our charity and our mission. Donate to freeCodeCamp.org.

and enhance.

GitHub is an online platform for hosting projects with version control. There, you can find many open source projects (like [freeCodeCamp](https://freeCodeCamp.org)) that you can contribute to and practice your skills.

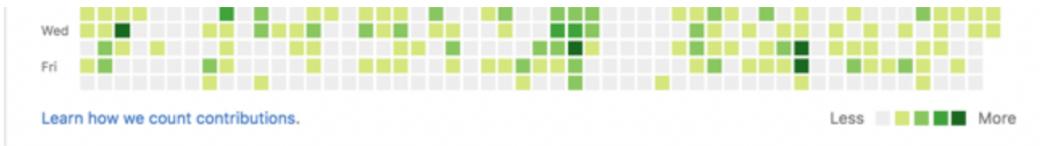
 **Tip:** many open source projects welcome first-time contributions and contributions from all skill levels. These are great opportunities to practice your skills and to contribute to real-world projects.



freeCodeCamp's [GitHub repository](#).

Contributing to open source projects on GitHub is great to acquire new experience working and communicating with other developers. This is another important skill for finding a job in this field.

Support our charity and our mission. Donate to freeCodeCamp.org.



GitHub tracks your [contributions](#) and shows them on your profile. These are shown as interactive gray and green squares that represent days of the current year. A darker shade of green means that more contributions were made on that day. Image taken from [this GitHub article](#).

Working on a team is a great experience. I totally recommend it once you feel comfortable enough with your skills and knowledge.

You did it! You reached the end of this article. Great work. Now you know what programming is all about. Let's see a brief summary.

◆ In Summary

- Programming is a very powerful skill. If you learn how to code, you can make your vision come true.
- Programming has many different applications in many different fields. You can find an application for programming in basically any field you choose.
- Programming languages can be classified based on different criteria and they share basic concepts such as variables, conditionals, loops, and functions.
- Always set goals and take detailed notes. To succeed as a programmer, you need to be enthusiastic and consistent.

Thank you very much for reading my article. I hope you liked it and found it helpful. Now you know why you should learn how to code.

Support our charity and our mission. Donate to freeCodeCamp.org.



Estefania Cassingena Navone

Developer, technical writer, and content creator @freeCodeCamp. I run the freeCodeCamp.org Español YouTube channel.

If you read this far, tweet to the author to show them you care.

[Tweet a thanks](#)

Learn to code for free. freeCodeCamp's open source curriculum has helped more than 40,000 people get jobs as developers.

[Get started](#)

freeCodeCamp is a donor-supported tax-exempt 501(c)(3) charity organization (United States Federal Tax Identification Number: 82-0779546)

Our mission: to help people learn to code for free. We accomplish this by creating thousands of videos, articles, and interactive coding lessons - all freely available to the public. We also have thousands of freeCodeCamp study groups around the world.

Donations to freeCodeCamp go toward our education initiatives, and help pay for servers, services, and staff.

You can [make a tax-deductible donation here](#).

Trending Guides

[Best Python IDEs](#)

[Python Index](#)

[Git List Branches](#)

[What is HTML?](#)

[Java Enum Example](#)

[What is UTF-8?](#)

Support our charity and our mission. Donate to freeCodeCamp.org.

[Python List Length](#)[HTML Dot Symbol](#)[Python String to Int](#)[indexOf in Java](#)[Int to String in C++](#)[Image File Types](#)[CSS Margin vs Padding](#)[SOLID Principles](#)[HTML Starter Template](#)[How to Flush DNS](#)[Rubber Duck Debugging](#)[DNS Server Not Responding](#)[SQL vs NoSQL Databases](#)[Module Exports in Node.js](#)[Python For Loop Example](#)[Computer Skills for Résumé](#)[Int Object Not Iterable](#)[Force Delete a Windows File](#)[How to Type Vertical Bar](#)[Inspect an Element in Chrome](#)

Our Charity

[About](#) [Alumni Network](#) [Open Source](#) [Shop](#) [Support](#) [Sponsors](#) [Academic Honesty](#)[Code of Conduct](#) [Privacy Policy](#) [Terms of Service](#) [Copyright Policy](#)