

Forum:[Hack Forums](#)

Thread: [This is THE BEST programming language to learn FIRST](#)

Kalvaxas - March 2, 2024, 7:50 pm

First off, thanks for giving into my clickbait title. The internet gods appreciate your service ?

Now let me tell you why I'm creating this thread. As of late, I'm seeing quite a lot of posts asking "Which programming language should I learn first?" Or, "What's the first language you started with, and why?" - Well, the latter may be relevant to this thread simply because the OP usually explains WHAT said language is used for. Okay Sasuke, so what are you getting at here? Let me explain.

Say you own a car, a bike, a plane, and an ATV. All of them are drivable on the road. However, you won't be driving all of them on the road. Programming is very much the same. You can theoretically use every major programming language to do the same task, but a lot of them wouldn't make sense to perform certain tasks. For example, while you could use Python to program an OS, you won't. Nobody will. My point here is that you absolutely NEED to research languages best suited for what kind of projects you think you'll be making the most.

A prime example of why you don't want to take someone's advice on what language to learn first is: Most people will tell you that the Python syntax is very easy to understand. It's very user friendly and clear with what's being done in the code. This is very true. However, if you're wanting to become a web developer, then learning Python shouldn't be your first choice as you'll be using JavaScript primarily. You'd also need to understand and learn how HTML/CSS work. Likewise, if you wanted to program apps for iOS, it wouldn't make sense to learn JS or Python first. You'd want to learn Swift. See what I'm getting at here?

Now, before I get into some of the most popular languages, I'd like to mention that there's a plethora of languages out there. As I mentioned a bit earlier in the thread, most languages can and will be able to do all of the same tasks. This is where research comes in. You need to do your own research to figure out which language is optimal for the tasks you'll be performing. The amazing thing here is that even though the syntaxes are different in every language - Once you fully master a language, learning another language will go much quicker than you may think since you already know how code works. That said, I don't recommend learning a second language until you've become proficient in your primary language.

Functions of some popular languages

C

What C Does Best ([Click to View](#))System-Level Programming:

C is powerful for low-level system programming and tasks like developing operating systems.

Efficient Memory Management:

Offers manual memory management, giving precise control over memory usage.

Low-Level Hardware Interaction:

Ideal for tasks that involve direct interaction with hardware, like device drivers.

Procedural Programming:

Follows a procedural paradigm, making it suitable for clear step-by-step execution.

C#

What C# Does Best ([Click to View](#))Windows Applications:

Widely used for developing Windows desktop applications with graphical user interfaces.

Web Development:

Part of the .NET framework, making it a strong choice for server-side web development.

Game Development with Unity:

Primary language for game development using the Unity engine.

Object-Oriented Programming:

Supports object-oriented programming for modular and reusable code.

C++

What C++ Does Best ([Click to View](#))System-Level Programming:

Used for system-level programming and resource-intensive applications.

Game Development:

Major player in the gaming industry, employed for high-performance games and engines.

I'd like to also add here that C++ is the language that primarily developed Unity itself to give you an idea of how much more powerful C++ is when compared to C# in game development.

Embedded Systems:

Suitable for embedded systems programming, offering a blend of abstraction and low-level control.

Object-Oriented Programming:

Extends C with object-oriented features, combining structured and object-oriented programming.

Python

What Python Does Best ([Click to View](#))Automation:

Excellent for automating repetitive tasks, making it a go-to language for various automation scenarios.

Data Analysis:

Widely used in data science for handling data, crunching numbers, and running statistical analyses.

Web Development:

Used for web development, both on the server side and in frameworks like Django.

Artificial Intelligence and Machine Learning:

Popular in AI and ML for its simplicity and vast library support.

JavaScript

What JS Does Best ([Click to View](#))Web Development:

The language of the web, used for adding interactivity and dynamic features to websites.

Server-Side Development:

Can be used on the server side with platforms like Node.js.

Game Development:

Utilized for browser-based and simple game development.

Asynchronous Programming:

Known for its asynchronous capabilities, handling multiple operations concurrently.

Java

What Java Does Best [\(Click to View\)](#)Platform Independence:

Java is big on essentially every platform. Probably the most versatile language.

Web Development:

Used for web applications and server-side development.

Android App Development:

Primary language for developing Android applications.

Enterprise-Level Applications:

Widely used in large-scale enterprise applications due to its scalability and reliability.

Swift - Not much to say ?

What Swift Does Best [\(Click to View\)](#)iOS App Development:

Default language for everything Apple. For what it's worth, Swift is open-source, so collabs are very much a thing. That's really all I have on Swift.

Rust

What Rust Does Best [\(Click to View\)](#)System-Level Programming:

This is one of the top 'memory-safe' programming languages out there at the system level.

Concurrency

One of the top concurrent languages. I know most of you that are reading this won't understand what this means, so I'll break it down a bit. Concurrent programming is where you're running multiple sequences at the same time. Think of it as multitasking, because it's not much different than that. It's just a whole lot more effective.

Memory Safety:

Emphasizes zero-cost abstractions while preventing common programming errors like null pointer dereferences. Essentially what this means is that your program won't be randomly crashing or exiting whatever sequence is currently running.

WebAssembly Support:

Remember how I said that Java is extremely versatile? So is Rust. WebAssembly is meant for high performance, and is a low level code. You'd use it for anything from media apps, to tools such as Libre Office. To CAD software, to emulators. I believe Rust is becoming more and more prolific for AI as well, so keep that in mind. Rust can even create IDEs (What you write your code in). You can use it for cryptography and networking as well. See what I mean by versatile? Shout out to [MavenGradle](#) for mentioning Rust to be added.

I'm going to use ChatGPT here to generate coin flip snippets to show you some code in all of the programming languages listed above. I'm doing this for a few reasons. One, I want you to see how simple some syntaxes are vs others. Two, I want you to be able to grasp how simple tasks work. Three, I want you to see that depending on the language - libraries can seriously shorten time required to write out code.

C Code for Coin Flip:

```
Code \(Click to View\)Code #include <stdio.h>
#include <stdlib.h>
#include <time.h>
```

```
int main() {
    srand(time(0));
    int result = rand() % 2;
    printf("It's %s!\n", result == 0 ? "Heads" : "Tails");
    return 0;
}
```

C# Code for Coin Flip:

Code [\(Click to View\)](#)Code using System;

```
class Program {
    static void Main() {
        Random random = new Random();
        int result = random.Next(2);
        Console.WriteLine($"It's {(result == 0 ? "Heads" : "Tails")}!");
    }
}
```

C++ Code for Coin Flip:

```
Code \(Click to View\)Code #include <iostream>
#include <cstdlib>
#include <ctime>
```

```
int main() {
    std::srand(std::time(0));
    int result = std::rand() % 2;
    std::cout << "It's " << (result == 0 ? "Heads" : "Tails") << "!" << std::endl;
    return 0;
}
```

Python Code for Coin Flip:

Code [\(Click to View\)](#)Code import random

```
def flip_coin():
    result = random.choice(["Heads", "Tails"])
    print(f'It's {result}!')
```

```
# Call the function to flip the coin
flip_coin()
```

JavaScript Code for Coin Flip:

Code [\(Click to View\)](#)

```
Code function flipCoin() {  
  // Generate a random number (0 or 1)  
  const result = Math.floor(Math.random() * 2);  
  
  // Display the result  
  console.log(`It's ${result === 0 ? 'Heads' : 'Tails'}!`);  
}  
  
// Call the function to flip the coin  
flipCoin();
```

Java Code for Coin Flip:

Code [\(Click to View\)](#)

```
Code import java.util.Random;  
public class CoinFlip {  
  public static void main(String[] args) {  
    // Create a random number generator  
    Random random = new Random();  
  
    // Generate a random number (0 or 1)  
    int result = random.nextInt(2);  
  
    // Display the result  
    System.out.println("It's " + (result == 0 ? "Heads" : "Tails") + "!");  
  }  
}
```

Swift Code for Coin Flip:

Code [\(Click to View\)](#)

```
Code import Foundation  
  
func flipCoin() {  
  // Generate a random number (0 or 1)  
  let result = Int.random(in: 0...1)  
  
  // Display the result  
  print("It's \(result == 0 ? "Heads" : "Tails")!")  
}  
  
// Call the function to flip the coin  
flipCoin()
```

I hope that this helps everyone who's asking which language they should learn first. If you see someone asking, or even if you're asking this question for yourself. Feel free to guide them here or keep coming back as needed. Feel free to quick love this thread if I was of any help to you.
Thanks!

-Sasuke

MavenGradle - March 3, 2024, 7:39 am

Good thread except for the clickbait title that benefits no one. Would be even better if you add a section for Rust, perhaps under or next to C.

It deserves a special place on your list for what it can do ?.

Kalvaxas - March 3, 2024, 8:11 am

(Mar 3, 2024 08:39 AM)MavenGradle Wrote: [\[CITE\]](#) Good thread except for the clickbait title that benefits no one. Would be even better if you add a section for Rust, perhaps under or next to C.

It deserves a special place on your list for what it can do ?.

Thanks!

I do plan to add more languages to this thread over time. Ruby is one I considered adding yesterday, but didn't get around to it because of time constraints. I'll definitely add Rust to the list later today when I get some more time. ?

Dirty - March 3, 2024, 7:00 pm

The message you are trying to convey is a good one, but does not work for someone with 0,0 programming experience. For them there is no concept of what a good language for their problem would be. Which is why I usually suggest that it does not matter, and that getting coding is actually simply the priority. They will not release their code to any meaningful audience, no one will pay them, it will be a simple hobby project. Their concern is learning how to program then they can worry wether C, C++, Rust or Golang is their answer for their solution. (Hint it will always be Rust and/or Golang)

Kalvaxas - March 3, 2024, 7:17 pm

(Mar 3, 2024 08:00 PM)Dirtybox Wrote: [\[CITE\]](#) The message you are trying to convey is a good one, but does not work for someone with 0,0 programming experience. For them there is no concept of what a good language for their problem would be. Which is why I usually suggest that it does not matter, and that getting coding is actually simply the priority. They will not release their code to any meaningful audience, no one will pay them, it will be a simple hobby project. Their concern is learning how to program then they can worry wether C, C++, Rust or Golang is their answer for their solution. (Hint it will always be Rust and/or Golang)

Yeah, I totally understand that. I don't know if my logic applies to everyone, but for me as someone who previously had no experience, I never realized that certain languages were used for certain things. I always just assumed the programming language best suited for anyone would be whichever syntax they grasp first. Which, honestly, that may indeed be true. My goal here though is to help people to understand what they should be searching for when they decide to dive into programming. I guess you could compare it to buying your first car. For the most part, you just need a car that can get from point A to point B. You also would need to take into account the MPG that you get on said car though. Maintenance costs. Aesthetics, etc. That was my intention for creating this thread.

Also, [MavenGradle](#) - I added Rust to the OP. I actually ran out of characters for the OP, so unfortunately, Rust is the last language I can add to it. Hopefully people have a much better understanding though. ?