Best Programming Language to Learn in 2020 (for Job & Future)

Posted in Software Development (https://hackr.io/blog/category/software-development), Programming (https://hackr.io/blog/category/programming)

Aman Goel
(https://hackr.io/blog/author/amangoel)
Last Updated 06 Aug, 2020

Share:

(https://twitter.com/intent/tweet?text=Best+Programming+Language+to+Learn+in+2020+%28for+Job+%26+Future%29+https%3A%2F%2Fhackr.io%2Fblog%2Fbest-programming-languages-to-learn-2020-jobs-future) (https://www.linkedin.com/shareArticle?mini=true&url=https://hackr.io/blog/best-programming-languages-to-learn-2020-jobs-future) (http://www.reddit.com/submit?url=https://hackr.io/blog/best-programming-languages-to-learn-2020-jobs-future) (https://news.ycombinator.com/submitlink?u=https://hackr.io/blog/best-programming-languages-to-learn-2020-jobs-future) (https://api.whatsapp.com/send?text=https%3A%2F%2Fhackr.io%2Fblog%2Fbest-programming-languages-to-learn-2020-jobs-future)

78 Comments

Table of Contents

The most important skill to learn in today's world is to know how to write a computer program. Today, computers have entered in almost every industry. Be it the autopilot in an aircraft or digital speedometer in your bike, computers in various forms surround us. Computers are extremely useful for an organization to scale up well. Gone are the days of pen and paper. Today, in order to store and access your information, you absolutely need computers.

The programming and developer communities are emerging at a rate faster than ever before. Various new programming languages are coming up that are suited for different categories of developers (beginners, intermediate, and experts) as well as for different use cases (web application, mobile applications, game development, distributed system, etc). Every beginner is puzzled with the question, "What programming language should I learn?"

Levels of Programming

There exist several programming languages with their own specific purpose and contains a unique set of keywords and syntax that are used to create instructions. The programming language varies in the level of abstraction and classified into two categories:

1. Low-level language
2. High-level language

Low-Level Language

Low-level languages provide abstraction from the hardware and are represented in the binary form i.e. 0 or 1 which are the machine instructions. Low-level languages are further classified as machine-level language & assembly level language.

Parameters	Machine Level Language	Assembly Level Language
Hierarchy Level	It is at the lowest level in the hierarchy and has zero abstraction level from the hardware.	It is above the machine level language in the hierarchy and so has less abstraction level from the hardware.

Learning Curve	It is hard to understand by Humans.	It is easy to learn and maintain.
Written as	It is written in binary that is 0 or 1.	It is written in simple English and is easy to understand.
Generation	It is a first-generation programming language.	It is the second-generation programming language.
Requirement for Translator/Assembler	The machine code is executed directly so no translator is required.	It requires an assembler to convert assembly language to machine code.

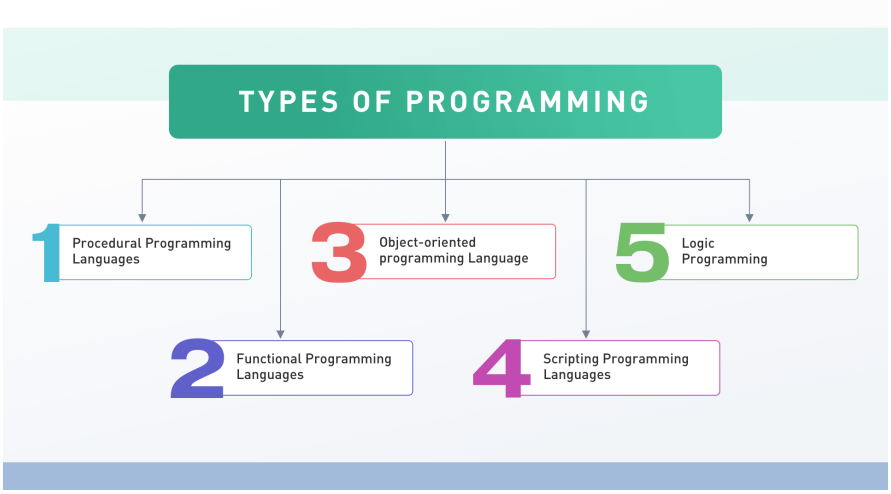
High-Level Language

High-level language allows us to write programs that are independent of the type of computer. The high-level languages are named as high-level because they are close to human languages and can be understood easily, however it requires attention to the logic of the problem. The language needs a compiler to translate a high-level language into a low-level language. Further, the high-level languages provide the following advantage.

1. The high-level language is easy to learn & maintain.
2. The high-level languages are portable i.e. they are machine-independent.

Parameters	**Low-Level Language**	**High-Level Language**
Level of Understanding	It is machine friendly i.e. easily understood by computers.	It is user friendly, as it is written in simple English.
Time of Execution	Takes time to execute.	Executes at a faster pace.
Tool Required	It requires the assembler to convert assembly code to machine code.	It requires the compiler to convert the high-level language to machine instructions.
Portability	It is not portable.	It is portable.
Memory Efficiency	It is memory efficient.	It is less memory efficient.
Debugging and Maintenance	Not easy	Easy

Types of Programming



Let us now take an overview of different types of programming languages:

1. Procedural Programming Languages

This programming paradigm, derived from structured programming specifies a series of well-structured procedures and steps to compose a program.

It provides a set of commands by segregating the program into variables, functions, statements & conditional operators. Various Programming editors or IDEs help users develop programming code using one or more programming languages. Some of them are Adobe Dreamweaver, Eclipse or Microsoft visual studio, BASIC, C, Java, PASCAL, FORTRAN are examples of Procedural Programming Language.

2. Functional Programming Languages

A functional programming language is a declarative programming paradigm where programs are constructed by applying and composing functions. The language emphasizes expressions and declarations than on the execution of statements. The foundation of functional programming is lambda calculus which uses conditional expressions and recursion to perform the calculations. It does not support iteration like loop statements & conditional statements like if-else. Some of the most prominent functional programming languages are Haskell, SML, Scala, F#, ML, Scheme, and More.

3. Object-oriented programming Language

This programming paradigm is based on the “objects” i.e. it contains data in the form of fields and the code in the form of procedures. OOPs, offer many features like abstraction, encapsulation, polymorphism, inheritance, classes, and Objects. Encapsulation is the main principle as it ensures secure code. It also emphasizes code reusability with the concept of inheritance and polymorphism allows the spreading of current implementations without changing much of the code. Most multi-paradigm languages are OOPs languages such as Java, C++, C#, Python, Javascript, and more.

You can read more in detail about OOPs Concept here.

4. Scripting Programming Languages

All scripting languages are programming languages that do not require a compilation step and are rather interpreted. The instructions are written for a run time environment. The languages are majorly used in web applications, System administration, games applications, and multimedia. It is used to create plugins and extensions for existing applications. Some of the popular scripting languages are:

- Server Side Scripting Languages: Javascript, PHP, and PERL.
- Client-Side Scripting Languages: Javascript, AJAX, Jquery
- System Administration: Shell, PERL, Python
- Linux Interface: BASH
- Web Development: Ruby

5. Logic Programming

The programming paradigm is largely based on formal logic. The language does not tell the machine how to do something but employs restrictions on what it must consider doing. PROLOG, ASAP(Answer Set programming), and Datalog are major logic programming languages, rules are written in the form of classes.

Let us take a look at best Programming Languages to learn in 2020 for a job and for future prospects:

Best Programming Languages

1. Python

Python (https://www.python.org/) undoubtedly tops the list. It is widely accepted as the best programming language to learn first. Python is fast, easy-to-use, and easy-to-deploy programming language that is being widely used to develop scalable web applications. YouTube, Instagram, Pinterest, SurveyMonkey are all built-in Python. Python provides excellent library support and has a large developer community. The programming language provides a great starting point for beginners. Talking about those who are looking for a better job, you should definitely learn Python ASAP! A lot of startups are using Python as their primary backend stack and so, this opens up a huge opportunity for full-stack Python developers. Here is a sample Python “Hello World!” program:

```
print “Hello World!"
```

Yes, Python is that simple! Anyone who wishes to join a startup should master Python programming (https://hackr.io/tutorials/learn-python?ref=blog).

Difficulty level: Easy to learn. Best language for beginners. 5 out of 5.
Job opportunity: Huge! 5 out of 5.

Pros:

- Creating and using classes and objects is easy thanks to OOP characteristics
- Extensive library support
- Focuses on code readability
- Has the ability to scale even the most complex applications
- Ideal for building prototypes and testing out ideas faster
- Open-source with an ever-growing community support
- Provides support for a multitude of platforms and systems
- Very easy to learn (https://hackr.io/tutorials/learn-python?ref=blog-post) and use

Cons:

- Not suitable for mobile computing
- Slower by virtue of being an interpreted programming language
- The database access layer is somewhat immature
- Threading isn't good because of GIL (Global Interpreter Lock)

2. Java

Java (https://www.java.com/en/) is another popular choice in large organizations and it has remained so for decades. Java is widely used for building enterprise-scale web applications. Java is known to be extremely stable and so, many large enterprises have adopted it. If you are looking for a development based job at a large organization, Java is the language that you should learn. Java is also widely used in Android App Development. Almost any business today needs an Android Application owing to the fact that there are billions of Android users today. This opens up a huge opportunity for Java developers given the fact that Google has created an excellent Java-based Android development framework - Android Studio.

Difficulty level: Easy to moderate to learn. 4 out of 5.
Job opportunity: Huge! 4.5 out of 5. [Best Java tutorials and courses (https://hackr.io/tutorials/learn-java?ref=blog)]

Pros:

- An abundance of open-source libraries
- Automatic memory allocation and garbage collection
- Follows the OOP paradigm
- Has the stack allocation system
- A high degree of platform independence thanks to the JVM feature

- Highly secure due to the exclusion of explicit pointer and inclusion of a security manager responsible for defining the access of classes
- Ideal for distributed computing
- Offers a galore of APIs for accomplishing different tasks, such as database connection, networking, utilities, and XML parsing
- Supports multithreading

Cons:

- Absence of templates limits creating high-quality data structures
- Expensive memory management
- Slower than natively compiled programming languages, like C and C++

3. C/C++

C/C++ is like the bread and butter of programming. Almost all low-level systems such as operating systems, file systems, etc are written in C/C++. If you wish to be a system-level programmer, C/C++ is the language you should learn. C++ is also widely used by competitive programmers owing to the fact that it is extremely fast and stable. C++ also provides something called STL - Standard Template Library. STL is a pool of ready-to-use libraries for various data structures, arithmetic operations, and algorithms. The library support and speed of the language make it a popular choice in the High-frequency trading (https://www.investopedia.com/terms/h/high-frequency-trading.asp) community as well.

Difficulty level: Easy to moderate to learn. 3 out of 5.
Job opportunity: Moderate! 3.5 out of 5.

Pros:

- A galore of compilers and libraries to work with [C++]
- Eases accessing blocked or hidden objects by other programming languages [C]
- Faster execution of programs than most programming languages [C/C++]
- Forms the basis for understanding more complex programming languages [C/C++]
- Language of choice for multi-device, multi-platform app development [C++]
- Offers a great degree of portability [C]
- Procedure-oriented language with a group of function modules and blocks. These make debugging, testing, and maintaining the programs easier [C]
- Programs are more efficient and easy to understand [C/C++]
- Rich function library [C++]
- Runs close to the system hardware and hence, offers a low level of abstraction [C/C++]
- Support exception handling and function overloading [C++]
- Wide variety of application domains, such as games, GUI applications, and real-time mathematical simulations [C++]

Cons:

- Complex syntax [C/C++]
- Doesn't support program namespace [C]
- Incapable of solving modern, real-world programming challenges [C]
- Less efficient object-oriented system compared to other OOP-based programming languages [C++]
- Need to manually create the high-level constructs [C]
- No garbage collection or dynamic memory allocation [C/C++]
- No run-time checking [C/C++]
- No strict type checking [C]
- Not an easy first-choice for learning programming [C/C++]
- Plagued by the issues of buffer overflow and memory corruption [C/C++]
- Smaller standard library [C]

4. JavaScript

JavaScript (https://www.javascript.com/) is the "frontend" programming language. JavaScript is widely used to design interactive frontend applications. For instance, when you click on a button which opens up a popup, the logic is implemented via JavaScript.

These days, many organizations, particularly startups, are using NodeJS which is a JavaScript-based run-time environment. Node.js lets developers use JavaScript for server-side scripting—running scripts server-side to produce dynamic web page content before the page is sent to the user's web browser. Hence now with JS, you can use a single programming language for server-side and client-side scripts. If you are looking for that cool tech job at your favorite startup, you should seriously consider learning JavaScript.

Difficulty level: Easy to learn. 4.5 out of 5.
Job opportunity: Huge! 5 out of 5. [Best JavaScript tutorials and courses (https://hackr.io/tutorials/learn-javascript?ref=blog)]

Pros:

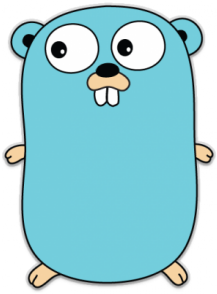
- Client-side JavaScript is very fast. It runs immediately within the web browser as there is no compilation requirement
- Gives a richer interface to a website
- Highly versatile
- It is the programming language of the web
- Reduced website server demand by virtue of being client-side
- Regular updates via the ECMA specification
- Several add-ons, such as Greasemonkey, for extending the functionality
- Simplistic implementation
- Plenty of resources and a mammoth community support
- Used for building a diverse range of applications

- Works exceptionally well with other programming languages

Cons:

- Absence of copy or equivalent method
- Allows only single inheritance
- As the code executes on the user machine, many people choose to disable JavaScript due to the fear of being exploited for a malicious intent
- Might be interpreted differently by different browsers

5. Go programming language



Go, also known as Golang, is a programming language built by Google. Go provides excellent support for multithreading and so, it is being used by a lot of companies that rely heavily on distributed systems. Go is widely used in startups in Silicon Valley. However, it is yet to be adopted by Indian companies/startups. Those who wish to join a Valley-based startup specializing in core systems should master Golang.

Difficulty level: Easy to moderate to learn. 3 out of 5.
Job opportunity: Moderate! 2.5 out of 5. [Best Golang tutorials (https://hackr.io/tutorials/learn-golang?ref=blog)]

Pros:

- Backed by Google
- Being a statically-typed language makes it more secure
- Cleaner syntax makes it easier to learn
- Comprehensive standard library offering a range of inbuilt functions for working with primitive types
- Ideal for building SPAs (single-page applications)
- Smart documentation
- Very fast as it is compiled to machine code

Cons:

- Absence of a virtual machine makes complex programs less efficient
- Implicit interfaces
- Lacks versatility
- No GUI library

- Underprivileged library support

6. R

R programming language (https://www.r-project.org/about.html) is one of the most commonly used programming languages for Data Analysis and Machine Learning. R provides an excellent framework and built-in libraries to develop powerful Machine Learning algorithms. R is also used for general statistical computing as well as graphics. R has been well adopted by enterprises. Those who wish to join “Analytics” team of a large organization should definitely learn R.

Difficulty level: Easy to moderate to learn. 3 out of 5.
Job opportunity: Huge! 4 out of 5. [Best R tutorials (https://hackr.io/tutorials/learn-r?ref=blog)]

Pros:

- Ability to run seamlessly on various operations systems
- Active, mushrooming community
- Being open-source and free grants the ability to make tweaks as per the requirements
- Comprehensive statistical analysis language
- Highly extensible
- Powerful package ecosystem

Cons:

- Lacks security features
- No strict programming guidelines
- Poor memory management
- Quality of some packages is subpar

7. Swift

Swift (https://developer.apple.com/swift/) is the programming language that is used to develop iOS applications. iOS-based devices are becoming increasingly popular. Apple iPhone, for instance, has captured a significant market share and is giving a tough competition to Android. Therefore, those who want to serve this community can learn Swift programming.

Difficulty level: Easy to moderate to learn. 3.5 out of 5.
Job opportunity: Huge! 4 out of 5. [Best Swift tutorials (https://hackr.io/tutorials/learn-ios-swift?ref=blog)]

Pros:

- Automatic memory management prevents memory leaks
- Backed by Apple
- Better scalability allows easily adding functionalities to the product and/or bringing in additional developers
- Easy to add new features
- Encourages developers to write clean and readable code

- English-like syntax makes it highly readable
- Interoperable with Objective-C
- It is possible to integrate Server-side Swift with any technology
- Makes code sharing better and development process faster when used for both frontend and backend development
- Very fast as compared to other popular programming languages, such as Objective-C and Python

Cons:

- Limited community support and resources
- Somewhat unstable due to being a relatively new arrival on the programming scene
- No support for legacy projects; can be used only for iOS7 or later apps

8. PHP

PHP is among the most popular backend programming language. Though PHP is facing tough competition from Python (https://hackr.io/blog/python-vs-php) and JavaScript, the market still needs a large number of PHP developers. Those who wish to join a reasonably well old organization as a backend developer should aim to learn PHP programming.

Difficulty level: Easy to learn. 4.5 out of 5.
Job opportunity: Huge! 4.5 out of 5. [Best PHP tutorials and courses (https://hackr.io/tutorials/learn-php?ref=blog)]

Pros:

- Abundance of powerful frameworks
- Easy to get started for making web pages
- First-class debugging with Xdebug
- Gigantic community support and a huge ecosystem
- Lots of automation tools for testing and deploying applications
- No scarcity of good automation tools for deployment and testing
- Supports object-oriented and functional programming paradigms (https://hackr.io/blog/programming-paradigms)

Cons:

- Developing websites completely in PHP is slower as compared to using other options
- Lacks in terms of security
- Poor error handling
- Requires extensions to a greater extent

9. C#

C# is a general-purpose programming language developed by Microsoft. C# is widely used for backend programming, building games (using Unity), building Window mobile phone apps and lots of other use cases.

Difficulty level: Easy to moderate to learn. 3.5 out of 5.
Job opportunity: 2.5 out of 5. [Best C# tutorials and courses (https://hackr.io/tutorials/learn-c-sharp?ref=blog)]

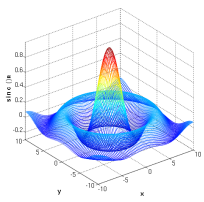
Pros:

- As pointer types aren't permitted, much safer than C and C++
- Ability to work with shared codebases
- Automatic scalable and updateable
- Component-oriented, object-oriented programming language
- Follows a syntax similar to the C programming language
- Fully integrated with the .NET library
- Ideal for all types of Windows development
- Rich sets of library functions and data types
- Supports type safety
- Quick compilation and execution times

Cons:

- Allows pointers in 'unsafe' blocks
- Almost all variables are references and memory deallocation is implicit using a garbage collector
- Offers less flexibility than C++
- Requires decent effort and time to learn
- Resolving errors requires serious expertise and knowledge

10. MATLAB



MATLAB (https://www.mathworks.com/products/matlab.html) is a statistical analysis tool that is used in various industries for Data Analysis. MATLAB is used widely in the Computer Vision and Image processing industry as well.

Difficulty level: Easy to moderate to learn. 3 out of 5.
Job opportunity: Huge! 4 out of 5. [Best MATLAB tutorials and courses (https://hackr.io/tutorials/learn-matlab?ref=blog)]

Pros:

- Eases developing scientific simulation thanks to a rich inbuilt library
- Functionality can be extended greatly by adding toolboxes
- High coding efficiency and productivity as it doesn't require a compiler for execution
- Ideal for developing scientific research applications
- Matlab Coder allows converting code for using in other programming languages, such as C++, Java, and Python
- Platform-independent

***Cons*:**

- Not free to use, requires purchasing a license
- Not much application beyond the scope of numerical computing
- Dealing with errors generated during cross-compilation requires extensive knowledge and experience
- Slower due to being an interpreted programming language

Summary

- Python and JavaScript (https://hackr.io/blog/python-vs-javascript) are hot in the startup world. Many startups use Django (Python), Flask (Python), and NodeJS (JavaScript) as their backend frameworks. Python and JavaScript are easy-to-learn and therefore considered the best programming languages (https://hackr.io/blog/what-is-programming-language) to learn for beginners. Moreover, both of them also provide a huge market opportunity. Therefore, those who are looking for a job change may also consider learning them.
- Java and PHP are hot in the corporate world. Many organizations use Spring (Java) and Codeigniter (PHP) as their web backend framework (https://hackr.io/blog/top-10-web-development-frameworks-in-2020).
- R and MATLAB (https://hackr.io/blog/r-vs-matlab) are hot in the Data Analytics world. If you wish to develop a career in Data Analytics or Data Science, these are the languages to learn.
- C/C++ and Golang are the top choices in building low-latency and scalable systems.

I hope you would have found your answer to "which programming language to learn first" by now. Feel free to ask your question(s) in the comment, and I'll be happy to respond. All the best!

People are also reading:

- Top Programming Interview Questions (https://hackr.io/blog/programming-interview-questions)
- Best Web Development IDE (https://hackr.io/blog/web-development-ide)
- What is Functional Programming? (https://hackr.io/blog/functional-programming)
- What is Procedural programming? (https://hackr.io/blog/procedural-programming)
- How to Learn Programming? (https://hackr.io/blog/how-to-learn-programming)

- Top SDLC Methodologies (https://hackr.io/blog/sdlc-methodologies)
- What is RAD? (https://hackr.io/blog/rapid-application-development-model)

Web Development (https://hackr.io/blog/tag/web-development) Javascript (https://hackr.io/blog/tag/javascript) Python (https://hackr.io/blog/tag/python) Java (https://hackr.io/blog/tag/java) Swift (https://hackr.io/blog/tag/swift) Programming (https://hackr.io/blog/tag/programming) Software Development (https://hackr.io/blog/tag/software-development) Golang (https://hackr.io/blog/tag/golang) R (https://hackr.io/blog/tag/r) C (https://hackr.io/blog/tag/c) PHP (https://hackr.io/blog/tag/php) MATLAB (https://hackr.io/blog/tag/matlab)

Share:

(https://twitter.com/intent/tweet?text=Best+Programming+Language+to+Learn+in+2020+%28for+Job+%26+Future%29+https%3A%2F%2Fhackr.io%2Fblog%2Fbest-programming-languages-to-learn-2020-jobs-future) (https://www.linkedin.com/shareArticle?mini=true&url=https://hackr.io/blog/best-programming-languages-to-learn-2020-jobs-future) (http://www.reddit.com/submit?url=https://hackr.io/blog/best-programming-languages-to-learn-2020-jobs-future) (https://news.ycombinator.com/submitlink?u=https://hackr.io/blog/best-programming-languages-to-learn-2020-jobs-future) (https://api.whatsapp.com/send?text=https%3A%2F%2Fhackr.io%2Fblog%2Fbest-programming-languages-to-learn-2020-jobs-future)

Aman Goel

(https://hackr.io/blog/author/amangoel)

Entrepreneur, Coder, Speed-cuber, Blogger, fan of Air crash investigation! Aman Goel is a Computer Science Graduate from IIT Bombay. Fascinated by the world of technology he went on to build his own start-up - AllinCall Research and Solutions to build the next generation of Artificial Intelligence, Machine Learning and Natural Language Processing based solutions to power businesses. View all posts by the Author (https://hackr.io/blog/author/amangoel)

Related Posts

(https://hackr.io/blog/free-coding-bootcamps)

11 Best Online Free Coding Bootcamps in 2020

(https://hackr.io/blog/free-coding-bootcamps) **Read More (https://hackr.io/blog/free-coding-bootcamps)**

(https://hackr.io/blog/best-programming-books)

10 Best Programming Books You Should Know

(https://hackr.io/blog/best-programming-books) **Read More (https://hackr.io/blog/best-programming-books)**

(https://hackr.io/blog/how-to-code-a-game)

How to Code a Game: Building a Game From Scratch

(https://hackr.io/blog/how-to-code-a-game) **Read More (https://hackr.io/blog/how-to-code-a-game)**

Leave a comment

Email address*

Enter email

Your email will not be published

Name*

Name

Comment*

SUBMIT

RANJANI
DOES NOT MEANTIONED SELENIUM ?

Reply

ranjani
hi am ranjani.....i want learn something about best scope computer language for my incresed my career ..so i choosed the selenium . but u say java is an best language ,but selenuym is an more than advatage of java,,and thats have an many scopes in computer language ,then how u do not speak about that,,,,

Reply

Abhishek Thakur
Hello respected person I'm new here. I didn't know about computer languages and coding. I want to learn how to code and make very useful applications for android and ios system. please teach me.

Reply

Tester
C# Windows mobile phone apps in 2020 really shows how much you know about c# and how updated you are.

Reply

Seth K.
Overall, I thought it was an interesting article and well done considering any article that addresses this topic will fan eternal flame wars even in the Mariana Trench. One thing, I would consider, however is if you are trying to write this article with regards to the present time would be to at least use Python3. The days of the print statement have been long gone even before Python2's end of life, which directly conflicts this article being an accurate portray of the then and now.

Reply

David
I'm sorry, but this list is misleading. A scan of the many top languages surveys will show that the result you get depends on the metric you use. What's most important to you?

To get a job: learn Python, Java or C# - those languages feature top of the "number of job vacancies listed" consistently and have done for years (I'm leaving javascript out of that list for reasons I'll mention in a moment).

Do you want to get a job in a particular field, say data science: learn a language that's key to that field - data scientists use Python or R, or Scala maybe if they are using Spark, they don't tend to use Swift or PHP. But if you want a job in web development, then it's Javascript or Ruby or PHP etc etc.

If you want a high paying job and you're already an established developer? Then maybe look at Go or Kotlin - there are far fewer jobs advertised, but also far fewer developers with those skills so developers are in high demand and can get better salaries.

Javascript often tops off a lot of lists, but that is because it is used as an adjunct to lots of other technologies - in particular web development. It's not so much used as a standalone language. It would be a mistake to just learn javcascript because in the web development world, javascript knowledge alone is pretty useless without knowledge of the tech that it works with - CSS, HTML, Angular, Express, React etc... all the millions of different frameworks in web dev client and server side.

Reply

Benoît
Not meaning to be rude.
But your two first points on JavaScript's cons don't even make sense.
I don't know what you mean by a "copy" method but if you mean cloning an object, there are certainly ways to do it. Also, not being able to do multiple inheritance is certainly not a drawback. Then again, I'm pretty sure you can find workarounds, due to JS's flexibility, which is both a strength and a weakness.

As for Go, there are lots of GUI libraries out there (bindings for GTK, Qt, etc.).

Overall it feels like you stuffed loads of things you read here and there about this and that language, without even verifying if they really made sense. Btw Node.js is not a framework but a JavaScript runtime. It does provide a standard library as any language would do. But if you want something similar to Django, you'd use NestJS, Sails, and if you want something similar to Flask, you'd choose Express, Koa, etc.

Reply

john
what about node js,angular js?

Reply

kidusw.com
Bro those are JavaScript based frameworks not languages.

Reply

loombago
My TOP 5 is:
#1 Python - best all-rounder, best for newbies and must know for DS and ML
#2 JS - must know for web, growing at back-end as well
#3 Kotlin - best JVM language, future (soon) king of android moble apps
#4 C# - must know for .Net ecosystem, very good all-rounder on pair with Java at big corporations
#5 Rust - C-challenger for system development, blazingly fast and secure

Reply

designer
thanks for sharing the web design coding related information

Reply

LOAD MORE COMMENTS

Related Tutorials

 Data Science

(https://hackr.io/tutorials/learn-data-science)

Intro to Programming

(https://hackr.io/tutorials/learn-intro-to-programming)

JavaScript

(https://hackr.io/tutorials/learn-javascript)

 Python

(https://hackr.io/tutorials/learn-python)