# Quelle: https://de.quora.com/Welche-ist-die-beste-Programmiersprache-um-das-Programmieren-zu-lernen Welche ist die beste Programmiersprache um das Programmieren zu lernen?

12 Antworten

Diese Frage ist dazu geeignet, einen Religionskrieg auf dieser Plattform auszulösen! Die Sprengkraft dieser Frage wird wahrscheinlich nur noch von jener der Frage “Warum ist Windows das Beste aller denkbaren Betriebssysteme?” überboten.

Und natürlich ist es, wie bei allen “Das Beste”-Frage, zuerst einmal wichtig, die Kriterien für gut und schlecht festzulegen. Schon da wird es schwierig …

Aber ich kann es ja mal versuchen!

Zuerst einmal soll es ja primär darum gehen, das *Programmieren* zu erlernen, und die Programmiersprache ist dafür nur ein Vehikel. Einigkeit dürfte schon mal darüber bestehen, dass man Programmieren nicht nur durch darüber lesen erlernt, sondern – wie z.B. auch Autofahren – durch *machen*. Das parallele Lernen einer Programmiersprache ist also Pflicht.

Da aber diese Programmiersprache nicht im Fokus steht, sollte sie einfach zugänglich sein, entweder weil sie simpel ist, oder weil komplexe Features am Anfang nicht benötigt werden und folglich erste einmal weggelassen werden können.

Dann sollte für diese Sprache gute Dokumentation leicht zu bekommen sein, und wenn die Werkzeuge für den Umgang mit der Sprache auch noch frei erhältlich sind, dann hat diese Programmiersprache schon mal eine ganze Reihe von Pluspunkten verbucht.

Dann geht es beim Programmieren-Lernen darum, Konzepte zu verstehen. Einfach sind Dinge wie Sequenz, Fallunterscheidung oder Schleife, interessanter sind dann schon Datentypen, Geltungsbereich, Namensraum, Sichtbarkeit, Rekursion (vs. Iteration), Datenstrukturen, Zeiger und Referenzen, Fehler- bzw. Ausnahmebehandlung und noch so einige mehr, die mir jetzt nicht alle einfallen.

Eine Programmiersprache, die eines dieser Konzepte nicht oder nicht “sauber” umsetzt, ist für das Verständnis dieses Konzepts nicht unbedingt hilfreich (weshalb z.B. viele reine Java-Programmierer Probleme haben mit dem Verständnis von Call-by-Value vs. Call-by-Reference), wobei manche der genannten (und der nicht genannten) Konzepte auch wichtiger sind als andere, zumindest am Anfang. Die Bedeutung mancher Konzepte (wie z.B. der Objektorientierung) erschließt sich manchmal auch erst, wenn man mit einem Projekt auf die Nase gefallen ist.

Dies haben auch verschiedene Lehrende erkannt und Sprachen mit dem Fokus auf die Ausbildung entwickelt; ein Beispiel wäre da Logo, aber auch das von Niklas Wirth entwickelte PASCAL und Oberon gehören in diese Klasse – siehe dazu auch [Bildungsorientierte Programmiersprachen – Wikipedia](https://de.wikipedia.org/wiki/Bildungsorientierte_Programmiersprachen). . Manche dieser Programmiersprachen sind für “echte” Projekte ungeeignet oder haben sich nicht durchsetzen können, andere (wie PASCAL) waren für eine Zeitlang sehr erfolgreich (siehe Delphi). Status heute ist jedoch keine dieser Lehrsprachen kommerziell wirklich relevant.

Was aber auch nicht wirklich wichtig ist, denn wenn man das Programmieren *richtig* erlernt hat – fast egal, mit welcher Programmiersprache – dann braucht man verhältnismäßig wenig Zeit für das Erlernen einer neuen Programmiersprache – eine Faustregel besagt, dass ein Programmierer eine neue Programmiersprache in ihren Grundzügen innerhalb von sechs Wochen drauf hat. Wobei der Löwenanteil beim Lernen einer neuen Programmiersprache darin liegt, die verfügbaren Bibliotheksfunktionen der Sprache kennenzulernen – was mit dem *Programmieren-Lernen* nichts zu tun hat! Eine Bibliothek mit Funktionen ist im Prinzip nur die Umsetzung eines Sprachkonzepts (Funktionen und Prozeduren) und als Lernender ist es viel besser (für den Lerneffekt), einen Algorithmus selbst zu implementieren als eine Bibliotheksfunktion aufzurufen.

Was ist nun die beste Programmiersprache, um das Programmieren zu erlernen?

Für das Selbststudium würde *ich persönlich* blutigen Anfängern immer noch PASCAL empfehlen, für jemanden mit gewissen Vorkenntnissen käme auch Java in Frage; zurzeit ist Java die bevorzugte Lehrsprache an den Hochschulen, was zur Folge hat, dass es hier sehr viel verfügbares Material mit einem didaktischen Schwerpunkt gibt.

Für einen Präsenzkurs würde ich noch immer einen einfachen Assembler (z.B. 6502) empfehlen, zumindest für den Anfang: Assembler zwingt dazu, schon einfachsten Code zu strukturieren und vor allem, zu kommentieren, anderenfalls weiß man in der Regel schon am nächsten Tag nicht mehr, was man da eigentlich verbrochen hat. Dies ist eine Lektion, die man für jede andere Programmiersprache auch gut nutzen kann, obwohl das natürlich streng genommen nichts mit dem “Programmieren lernen” zu tun hat.

Für weitere Schritte wäre meine Wahl dann in jedem Fall Java, auch – aber nicht nur –, weil ich ein Fan dieser Sprache bin.

BASIC und alle Derivate davon halte ich für absolut ungeeignet für Anfänger, weil es jegliche Struktur vermissen läßt. Mit VisualBasic hat sich da viel geändert, aber noch längst nicht alles.

C++ halte ich ebenfalls ungeeignet für Einsteiger, einfach weil es einen erschlägt. Die überall durchscheinenden Relikte von C machen es für Anfänger auch nicht einfacher, die verschiedenen Konzepte zu erkennen und zu verstehen.

C selbst würde meiner Meinung nach in einem Präsenzkurs ganz gut funktionieren, aber für ein Selbststudium vermutlich eher nicht so gut.

Für C# gilt fast dasselbe wie für Java, nur die Verfügbarkeit von Lehrmaterial ist bei weitem nicht so hoch. Zudem ist es auf anderen Betriebssystem als Windows nicht so ganz einfach, die Sprache an den Start zu bringen – was aber für jemanden ohne Programmierkenntnisse aber mit grundsätzlichen IT-Erfahrung ein lösbares Problem darstellen sollte.

Auch Python sollte man nicht vergessen: im Zeichen von Big Data hat die Sprache wieder (oder einfach mehr) an Bedeutung gewonnen, was sich auch auf die Verfügbarkeit von Tutorials und anderer Sekundärdokumentation ausgewirkt hat und ein Selbststudium vereinfacht.

Ein Kandidat, der als Lehrsprache auch immer gern genannt wird, ist Lisp. Wie gut oder schlecht diese Sprache für das Programmieren-Lernen geeignet ist, kann ich nicht beurteilen – ich habe das Programmieren mit anderen Sprachen gelernt und als ich das erste Mal mit Lisp konfrontiert wurde, hatte ich echte Probleme damit. Allerdings kenne ich einige Leute, die mit Lisp das Programmieren angefangen haben und mir sagen, dass es ihnen mit Sprachen wie C, C++ und Java so ging wie mir mit Lisp …

[Am Sonntag beantwortet](https://de.quora.com/Welche-ist-die-beste-Programmiersprache-um-das-Programmieren-zu-lernen/answer/Thorsten-Reitz)

Ich habe einige Jahre als Lehrbeauftragter Studierenden Software Engineering und Programmierung vermittelt, und bin nun dabei, es meinem älteren Sohn beizubringen.

Aus meiner Sicht hängt die “beste” Programmiersprache davon ab, wem und warum ich programmieren beibringen soll. Ich bin nicht so der Freund eher akademischer Sprachen mit geringer praktischer Relevanz, vor allem bei Programmieranfängern oder sogar Kindern. Ich denke, zu Beginn ist es wichtig, generelle Problemlösungsstrategien zu entwickeln und zügig erste sichtbare Erfolge zu erzielen. Dazu gehört die Analyse der Aufgabe, aber auch die Nutzung eines leistungsfähigen Debuggers. Das Setup der Entwicklungsumgebung sollte einfach sein.

Konkret habe ich die folgenden Sprachen verwendet, um Konzepte und deren Umsetzung zu vermitteln:

* Python für prozedurale Programmierung
* Java für klassische Objektorientierte Programmierung
* Kotlin für funktionale Programmierung
* JavaScript/TypeScript, im Client oder serverseitig

Mit meinem Sohn habe ich zuletzt [Programmieren mit der Maus](https://programmieren.wdrmaus.de/welcome%20(https:/programmieren.wdrmaus.de/welcome)) ausprobiert, das ist auch ganz witzig :).

[Vor 2 Std. beantwortet](https://de.quora.com/Welche-ist-die-beste-Programmiersprache-um-das-Programmieren-zu-lernen/answer/Albrecht-Scheidig)

Jede gängige Programmiersprache taugt meines Erachtens für das Erlernen des Programmierens.

Einen Tipp habe ich dennoch: Die Programmierumgebung sollte mit möglichst wenig Ablenkung daherkommen, damit man “einfach” Programme schreiben und debuggen kann. Eine gute IDE oder eine Art Sandbox (wie <https://jsfiddle.net/>) ist also ganz am Anfang eine empfehlenswerte Hilfe.

Etwas später sollte man dann lernen, wie alles “zu Fuß” geht, um besser zu verstehen, was da eigentlich passiert.

[Am Samstag beantwortet](https://de.quora.com/Welche-ist-die-beste-Programmiersprache-um-das-Programmieren-zu-lernen/answer/Thorsten-Edler)

Sehen Sie, darauf zu antworten ist für mich sehr problematisch.  
Nicht weil ich um eine Antwort verlegen bin.  
Nein, sondern weil ich keinen religiösen Kleinkrieg auf Quora starten will und mein persönlicher Stack meinem aktuellen Profil entspricht.  
Der ist neben Brocken proprietärer Syntax diverser BI/Reporting Frameworks Powershell, Python, R.

Jetzt weis ich ja auch nicht, was Sie vorhaben mit ihren zukünftigen Programmierfähigkeiten.  
Deshalb wage ich mal einen generischen Ansatz, bei dem es nicht all zu viel Nörgelei geben sollte.

Python  
Java  
Powershell (Oh, oh…)

Python um Programmieren zu lernen.  
Und fürs Prototyping.  
Einfacher geht es nun wirklich nicht.  
Und da lernen Sie dann auch Einrückung.

Java als breit angelegte Basis für alles.

Powershell, weil es das Schweizer Taschenmesser für alles in der Microsoft Welt ist.  
Andererseit, wenn Sie Python beherrschen, ist Powershell redundant.  
Das eine wird auf meiner Arbeitsstelle aber nicht unterstützt und Powershell lag da halt so rum.  
Da nehm ich doch dann lieber das, bevor ich ewig rumnöle beim Sysadmin, wegen der Installation irgendwelcher Libraries für Python.  
Sie sehen, Flexibilität ist alles.  
Hält den Geist jung.

Sie können aber auch gefühlte zehn andere Sprachen nehmen, die besser zu Ihnen passen könnten.  
Sie möchten sich ein klein wenig als elitärer Hacker fühlen, den schnöden restlichen Library Skriptern weit überlegen.  
Nehmen Sie C.  
Sie erhalten dann automatisch ein Dennis-Ritchie Gedenk T-Shirt zum handgeklöppelten Compiler dazu.  
Wenn Sie es auf Linux per gcc maken, signiert Ihnen Torvald den Kernel mit seinem persönlichen Schlüssel.  
You got the picture.

Für die drei da oben finden Sie aber für so ziemlich jeden Bereich, den Sie erkunden wollen Tutorials, Kurse, Stackoverflow-Antworten usw.

Außerdem ist die Jobsuche auf Jobrobot mit den Suchbegriffen ‘Java’, ‘Python’ oder ‘Powershell’ einfacher, als mit ‘ C ’.

p.s   
Ja, ich weis.  
Java ist eine Katastrophe.  
Powershell ist kein richtiges Programmieren.  
Und Python ein Emporkömmling.

Aber ich habe wichtigeres zu tun, als mich darüber zu zanken.  
Nehmen Sie einfach C#.  
Oder C++.  
Oder…

Formularende

[Am Sonntag beantwortet](https://de.quora.com/Welche-ist-die-beste-Programmiersprache-um-das-Programmieren-zu-lernen/answer/Florian-Woelki)

Wenn ich neu anfangen würde, würde ich mich für JavaScript, Java oder Python entscheiden.

Die Gründe dafür sind klar und einfach.

Python ist eine sehr klare und “Anfängerfreundliche” Programmiersprache. Du kannst sie lernen und danach sofort unglaubliches damit schaffen, weil es so viele Libs für Python gibt.

Java ist eine starke Enterprise Programmiersprache und wird in jedem Unternehmen verwendet, weil es sehr vielfältig ist und skalierbar ist. Java kann dir schnell Kopfschmerzen bereiten, aber die Regeln hier sind auch klar definiert.

JavaScript. Ich glaube dazu muss ich nicht viel sagen, weil du heutzutage für alles JavaScript benutzen kannst. Es gibt geniale Frameworks, wie React oder VueJS, und die Sprache ist zugleich noch einfach zu lernen und die Websprache.

[Vor 23 Std. beantwortet](https://de.quora.com/Welche-ist-die-beste-Programmiersprache-um-das-Programmieren-zu-lernen/answer/Christian-Schüler)

Programmieren lernt man nicht durch eine Programmiersprache, sondern es ist viel .ehr als nur der Teil des Codierens (erst hier kommt die Sprache zum Einsatz).

Programmierung beginnt immer mit Verstehen. Es ist notwendig das Problem zu verstehen, welches man mittels Software lösen möchte. Erst nach dem Verstehen, kann man entscheiden, ob man es mit Software lösen kann.

Ist das Problem mit Software lösbar, dann muss man eine Lösung erarbeiten. Dazu zerlegt man das Problem in möglichst kleine Teilprobleme und beginnt diese logisch zu lösen.

Bis hierhin ist noch keine Programmiersprache zum Einsatz gekommen. Selbst beim Erarbeiten der Lösungen für die evtl. entstandenen Teilprobleme kommt noch immer keine Programmiersprache zum Einsatz. Es kommt hier eine Abstraktionssprache zum Einsatz, die den logischen Ablauf des Programmes darstellt.

Ist das alles geschafft, dann kann man an die Umsetzung gehen und das ist der Teil des Codierens. Man muss hier allerdings die am besten geeignete Programmiersprache wählen. Welche Sprache für welche Aufgaben geeignet ist, kann man recherchieren. Es gibt hierzu viele Meinungen, Beispiele und selbst die Erschaffer der Sprachen geben hier Auskunft.

Leider wird Programmierung sehr häufig nur auf den Teil des Codierens heruntergebrochen. Bei sehr kleinen Projekten, ist das vielleicht auch möglich und akzeptabel. Spätestens bei größeren Projekten, sollte die Vorarbeit vor dem Codieren, nicht gescheut werden.

Ich erlebe es leider auch immer wieder, dass ich an Projekten arbeiten muss, bei denen die Vorarbeit nicht geleistet wurde und ich viel Zeit in die Einarbeitung in den Code und daraus das Verständnis des Problems des Autors des Codes ableiten muss.

Es gibt keine Beste Programmiersprache, um das Programmieren zu erlernen. Wie bereits andere Antwortgeber schrieben, ist die Sprache nur das Umsetzungswerkzeug, die eigenen Gedanken und Strategien, maschinenverständlich darzulegen.

Es gibt allerdings Programmiersprachen, die mit sehr guten IDEs daherkommen und einem viel Arbeit abnehmen. Ich denke, dass eventuell dies zumindest bei der Codierung das Erlernen der Syntax erleichtert. Da wäre zum Beispiel Visual Studio, welches mehrere Sprachen zur Entwicklung abdeckt. Netbeans oder Eclipse sind ursprünglich für die Java Entwicklung konzipiert worden, mittlerweile aber auch für andere Sprachen sehr gut geeignet.

Neben den großen IDEs gibt es dann auch noch schlankere Editoren, die auch beim Erlernen der Syntax unterstützen. Hier zu nennen ist der Neuling Visualisierung Studio Code oder Atom, alte Profiwerkzeuge, wie vi usw.

Das Erlernen der Syntax ist aber nicht gleichzusetzen mit dem Erlernen der Programmierung. Nur weil ich eine Sprache syntaktisch richtig anwenden kann, heißt das nicht, dass ich logisch, effizient und zielorientiert programmieren kann.

[Am Sonntag beantwortet](https://de.quora.com/Welche-ist-die-beste-Programmiersprache-um-das-Programmieren-zu-lernen/answer/Wolfgang-Geithner)

Wie die anderen Autoren auch sagen – die Frage lässt sich nur schwer mit einem Satz beantworten, und „es kommt darauf an“.

Ich denke Python eignet sich sehr gut um das Programmieren „an sich“ (Kontrollstrukturen, etc.) zu lernen. Es lässt sich vom einfachen Skripe bis zu paketierten, objektorientierten Softwareprojekten ausweiten und bietet gerade für wissenschaftliche und technische Anwendungsfälle viele Bibliotheken, die man nutzen kann (Datendarstellung, Raspberry–Pi, KI, …), denn man will ja nicht in den luftleeren Raum programmieren, sondern es ist gut, verfolgt man von Anfang an ein bestimmtes Ziel/Projekt. Allerdings abstrahiert Python die darunterlegende Hardware komplett, so dass man über Speichermanagement, etc nichts oder nur sehr wenig lernt. Ebenfalls ungewöhnlich ist die Tatsache, dass Python untypisierte Variablen erlaubt. Will man später umsteigen, ist die Typisierung erst einmal gewöhnungsbedürftig.

Java/C# sind sehr gut geeignet, um das objektorietierte Programmieren von der Pike auf zu lernen, da man dort ausschließlich objektorientiert programmieren kann. Ich würde allerdings C# als Sprache vorziehen, da ich dort die Entwicklungsumgebung mit dem Microsoft Visual Studio angenehmer finde als bei Java mit Eclipse/NetBeans. In Java/C# lassen sich sehr komplexe Anwendungen bauen, die jeden Anwendungsfall abdecken. Allerdings ist die Hardware wie bei Phython komplett abstrahiert.

C (hier habe ich keine Erfahrungen) ist die Sprache, wo man am meisten über das Zusammenspiel zwischen Software und Hardware lernt. Man muss sich um sehr viele hardwarenahe Aspekte selbst kümmern, was zwar manchmal schwierig ist, aber einen großen Lerneffekt mit sich bringt, den die anderen Sprachen nicht bieten.

Also: je nachdem was Dein Ziel ist und welcher Programmierertyp Du bist, solltest Du überlegen was Du am Ende bauen möchtest.

[Am Samstag beantwortet](https://de.quora.com/Welche-ist-die-beste-Programmiersprache-um-das-Programmieren-zu-lernen/answer/Lukas-Schmidinger-1)

Ich würde Java oder Python nehmen.

Motivation Java:

* Hohe Praxisrelevanz
* Sehr aktive Kommunity
* Viele andere Sprachen haben ähnliche Syntax (C-Familie)
* Viele Programmierpradigmen sind direkt in die Sprache eingebaut die lernt man gleich mit.

Motivation Python:

* Man muss sich um den Datentyp keinen Kopf machen und kann einfach loscoden.
* Ein schöner Programmierstyle ist erzwungen.
* Man kann unheimlich viel impotieren und findet auch viel Unterstützung.
* Python ist konsolengeeignet, also kann man Miniprogramme nebenher schreiben um sich das Arbeiten zu erleichtern.
* Python handelt von Hausaus große Zahlen.

[Am Samstag beantwortet](https://de.quora.com/Welche-ist-die-beste-Programmiersprache-um-das-Programmieren-zu-lernen/answer/Rafael-Bachmann)

Das kommt darauf an, was du willst.

* Willst du Algorithmen programmieren und Daten verarbeiten? Python ist eine gute Wahl. Wenn unsicher, dann auch Python.
* Wenn du interessiert an Arduino, Linux, high performance o.ä. bist, dann C.
* Wenn du bald einen Job haben willst, dann Java…
* Wenn du mathematisch veranlagt bist, probiere Haskell.
* Wenn du ins Web willst, kommst du um javascript nicht herum.

Auch einfache und bequeme Sprachen haben dann auf einmal Aspekte, die überhaupt nicht einfach sind (Javascript var vs. let, pass by reference/value, …) und das Fazit ist, dass keine Sprache einfach einfach ist.

Beispiel Rust: wirklich keine einfache Sprache, und doch ist es in Rust einfacher als in C, korrekte Programme zu schreiben…

[Am Samstag beantwortet](https://de.quora.com/Welche-ist-die-beste-Programmiersprache-um-das-Programmieren-zu-lernen/answer/Michael-Franke)

Meiner Meinung nach das reine C. Das von K&R. Da gibt es keine “Services” die das Leben erleichtern und man muss alles verstehen was passiert.

In dieser Sprache wurde UNIX geschrieben. Es ist m.E. die beste der Welt.

[Am Samstag beantwortet](https://de.quora.com/Welche-ist-die-beste-Programmiersprache-um-das-Programmieren-zu-lernen/answer/Johann-Joss)

Da gibt es die unterschiedlichsten Meinungen. Meine Empfehlung: Java.

Viel wichtiger als die Programmiersprache sind allerdings die Übungen.

[Am Samstag beantwortet](https://de.quora.com/Welche-ist-die-beste-Programmiersprache-um-das-Programmieren-zu-lernen/answer/Rainer-Schuster)

Python legal kostenfrei!