

KEY ELEMENTS FOR FUTURE PROGRAMMING LANGUAGES

 Read the text and choose the correct heading for each paragraph.

Abstract • Automation • Easy API calls • Easy learning curve • Potential for expansion • Special purpose languages

Key Elements for Future Programming Languages

What are future programmers going to need from their programming languages? That's a big and complicated question, especially considering we don't know what humanity's technological needs are going to be in the near or distant future. However, we can speculate about some important elements that all future programming languages are going to need to have.

- 1: Today's apps and tools need to make frequent exchanges with each other, with the cloud, and with other databases. That's why modern programming languages need to have easy and convenient API (Application Programming Interface, ie. libraries of compiled code) calls available.
- 2: It's easy to understand why developers favour automation: it makes everything easier. With the help of automation in a programming language, you can simplify your most menial tasks, test for bugs, and deploy code faster and more effectively.
- 3: To date, some of the most successful programming languages have been ones that can be used for just about anything. But as our needs become more focused and more refined, it's going to be more important for "special purpose"

languages to emerge – highly concentrated programming languages that are exclusively good for one or two types of applications.

- 4: Everyone is going to be a programmer in the future – at least to an extent. Simple apps and scripts, like ones that make spreadsheet management easier, are going to be in demand by professionals and amateurs alike. That's why new programming languages need to have the simplest, most accessible learning curve; they need to be easily learnable if they're going to survive.
- 5: The original purpose of programming languages was to convert the highly technical realm of coding to a higher layer of abstraction. In the future, programming languages may take this abstraction to an even higher level – grouping functions and simplifying commands to be even more intuitive.
- 6: It's hard to imagine what kinds of devices we'll invent (and grow to need) in the future. That's why tomorrow's programming languages need to be flexible and allow plenty of room for further growth and development. You can't just make a language for today; you have to make a language for the decades to come.

Adapted from: <https://readwrite.com/what-the-future-of-programming-languages-looks-like/>

Mark the sentences as “TRUE” or “FALSE”

- 1 We know exactly what humanity's technological needs are going to be in the future.
- 2 Modern programming languages don't need easy and convenient API calls.
- 3 Automation in programming languages makes everything more difficult.
- 4 Special purpose languages are not important for future programming languages.
- 5 Simple and accessible learning curves are not important for future programming languages.
- 6 The original purpose of programming languages was not to convert the highly technical realm of coding to a higher layer of abstraction.
- 7 Programming languages in the future may group functions and simplify commands to be even more intuitive.
- 8 Tomorrow's programming languages don't need to be flexible and allow plenty of room for further growth and development.
- 9 You only need to make a language for today, not for the decades to come.
- 10 Everyone will be a programmer in the future.