

## Programming Languages

One of the most important things to remember about programming languages is that many of them are very similar. So it's more important to learn basic concepts than to learn the detailed specifics of a specific language at first. There are always reference books and web sites for the exact syntax of a language. What you need to learn first are concepts that are common to all programming languages. Things like data structures ( functions, objects ), control structures ( commonly known as loops ), and some common APIs and libraries.

A programming language is designed to instruct a computer how to perform tasks. There are [high-level languages](#) and [low-level languages](#). High-level programming languages have been around since the beginning of digital computers and mimic human languages.

Low-level languages are normally specific to a given processor and very hard for a human to read. Some early popular favorite high-level languages were [COBOL](#) for business use, [FORTRAN](#) for science and engineering use, [Pascal](#), for academic use, and [Ada](#) for military use. These are still in use today but are gradually being replaced by a new wave of languages such as [Java](#), and C++.

Artificial languages, much like [natural languages](#), normally evolve until they become obsolete and die out. With every new innovation in computing, must come new functionality in a given language or programmers will migrate to something else. A typical credo for programmers is: always use the right tool (or programming language) for the job.

Now let's take a look at some of the most popular computer languages throughout time:

Perhaps the most venerable language of all time is [C](#). C was designed by Dennis Ritchey for the Unix operating system. It was, not surprisingly, based on a previous language called B. C is now an [ANSI](#) standard, and has been updated many times over the years and still enjoys much success as ANSI C and also with the derivatives C++, and more recently, Microsoft's C#.

Very often programming languages completely die. A once-popular language was [BASIC](#). Many students, this author included, was taught BASIC in school. BASIC was created to give non-technical users a simple way to interact with a computer in order to get them excited and make them productive. At one time in the 1980's, a version of it was included with nearly every computer sold. As time went on, however, the philosophy of teaching changed and now no one uses BASIC anymore. Most professionals agree that students should start out with the same languages that professionals use in their own daily work such as [JavaScript](#), [Perl](#), [PHP](#), Python, [Java](#), [C](#), or even [Visual Basic](#) (not really related to the original BASIC).

With the rise of the [Internet](#) and [distributed computing](#), future programming languages will most likely focus on security, [portability](#), and clear [syntax](#). Perhaps the best and most widely used example of this in action is the Java language developed by Sun Microsystems. Java is an [object-oriented language](#) which allows programmers to design programs using real-world concepts such as objects, classes, and [inheritance](#) from parents. What's more, Java code runs on a [virtual machine](#), which can be run on many different types of operating systems without modification. These features help make Java an ideal programming language for the future.