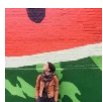


medium.com

Human languages vs. Programming languages

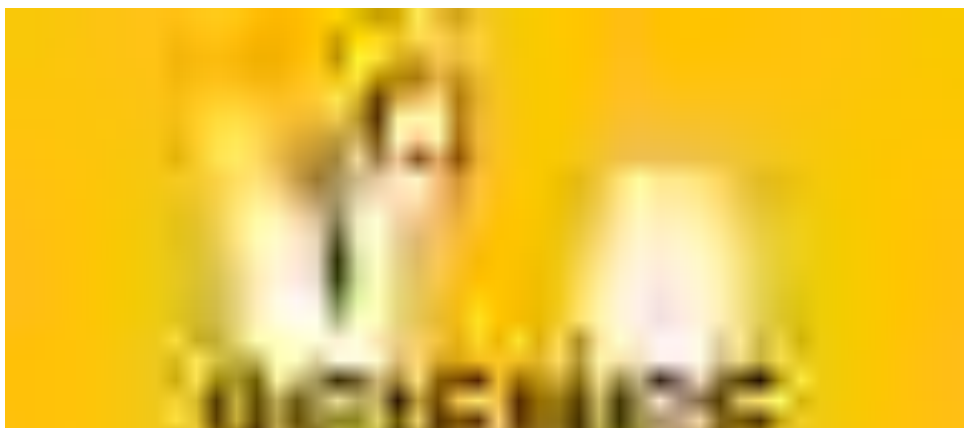
Ana V Harris

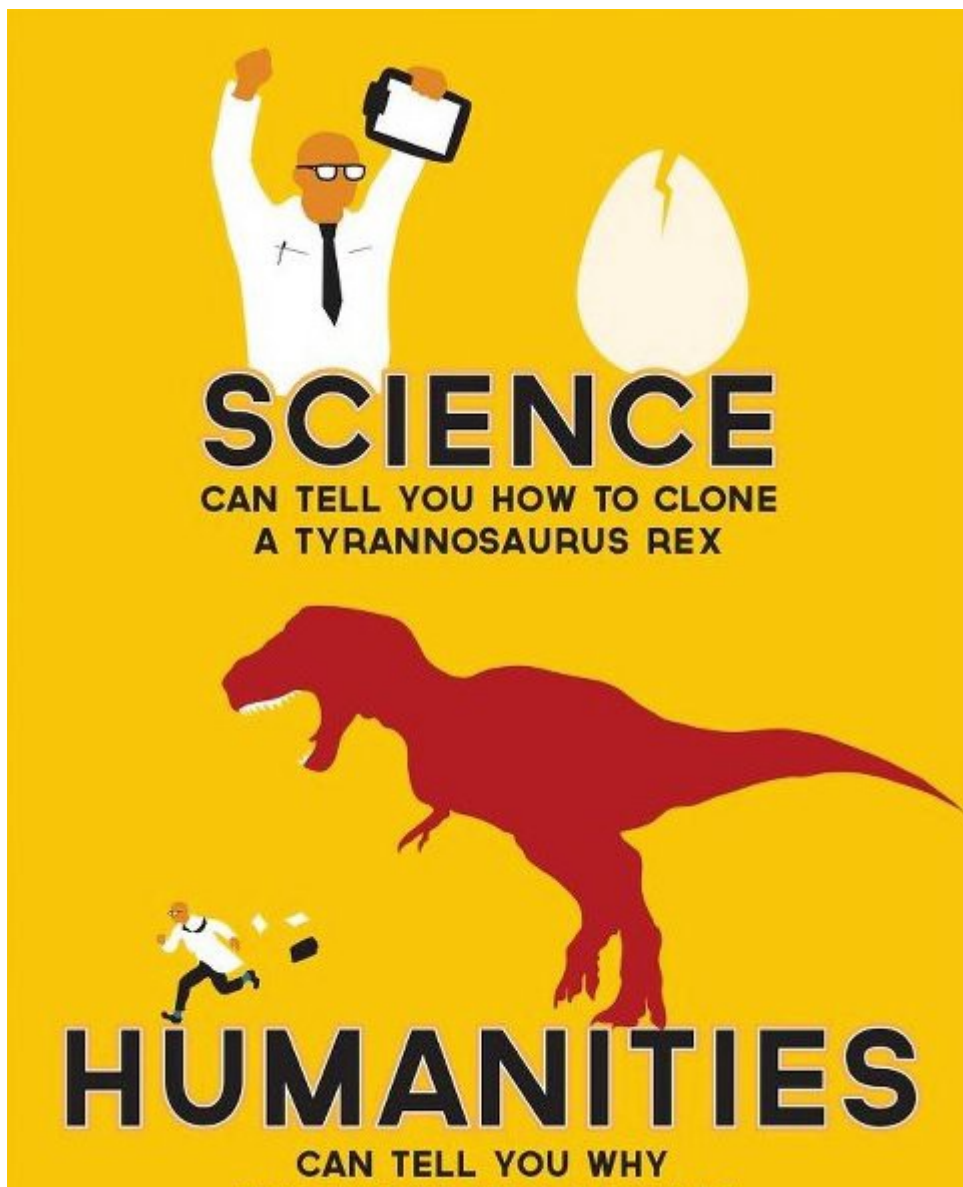
7-9 minutes



A brief comparison from a perspective of a linguist (soon to be) turned programmer

I still remember the day I first learned about computational linguistics. I had just graduated from science specialized high school, and faced the tough decision on what to study. For whatever reason I decided to go with languages, and it was on one of the first lectures that I learned about applied linguistics and its numerous subdivisions. Out of all those fields, computational linguistics caught my attention. I remember thinking it must be a perfect balance of two very different disciplines, ideal for the indecisive types like me who refuse to choose between humanities and science.





THIS MIGHT BE A BAD IDEA

Source: 3:AM Magazine

Nine years later I am learning again, but this time I'm on the other side of the coin — learning about programming. The memories of student days have returned, but this time I can do more than dream about this *perfectly balanced* field. I'm using this opportunity to learn more about linguistics, programming, and everything they have in common. So let's start from the very basics.

What is a language?

Defining a language might seem easy, but pages and pages could be (and have been) written about it. The reason behind this is the complex nature of a language as a phenomenon, as well as different ways to approach it and explain it.

Language can be defined as a system of spoken, manual, or written symbols that human beings use to express themselves, their identity, imagination, and emotions. Over time languages evolved and developed, and we found a way to describe and systematize those changes. However, their main purpose — communication — didn't change.

Programming languages revolve around the same principle of communication. They were created by humans as a system of symbols and rules used to communicate a set of instructions to a machine/computer. Although a lot simpler in their nature, programming languages have also developed their own classification and history.

So what are the similarities and differences between programming languages and human languages?

Similarities

We already got this from the previous paragraph, but let's repeat it again: the main function of languages, be it Python or Chinese, is communication. This is the most important similarity between them, and one of the main reasons we refer to both of them as languages.

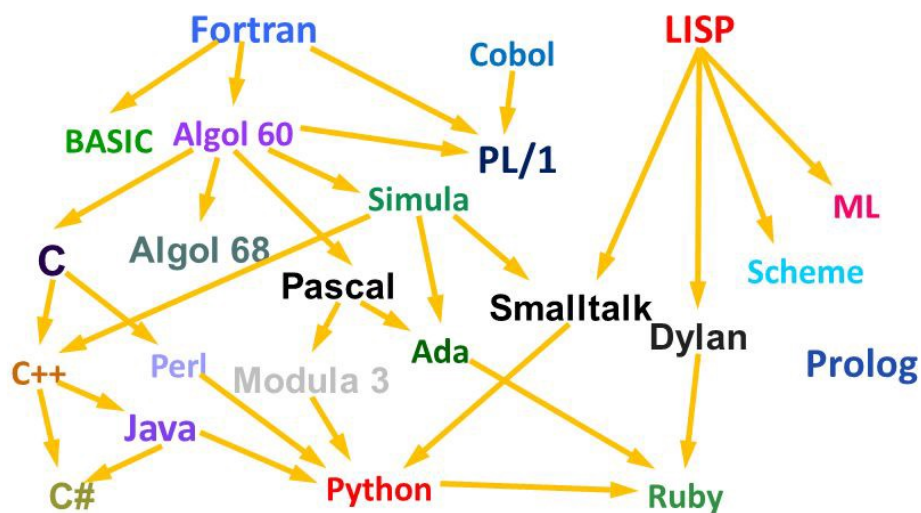
Another important feature that they have in common is structure. Two of the main concepts in linguistics are semantics and syntax. Semantics refers to the meaning of a certain word, or rather an information connected to a certain concept. For example, a word *eat* applies to a specific action living organisms can do. Syntax, on the other hand, is a set of rules that tell us how to arrange and combine words and phrases. The two are closely intertwined and only together can create a fully functioning expression. Take a sentence *Bed eats* as an example: while it's correct from the point of view of a syntax, semantically it's wrong since a bed is not alive and therefore cannot eat. Similarly to natural languages, programming languages distinguish between syntax and semantics. Every programming language is written with a certain idea or intention in mind (semantics) while following the set of rules around the use of variables, functions, different kinds of parenthesis, colons, etc. (syntax).

It is also important to point out that both form language families, or groups of related languages that branch one from another. We all have heard of the Indo-European language family, that includes English, among many other languages. Programming languages have their own families of language with relatively similar syntax and/or semantics. In the picture below you can see some of the more familiar programming languages (for more detailed genealogical tree, click [here](#)).



A family tree of languages

Some of the 2400 + programming languages



Source: History of Programming Languages by Ursula Lewis

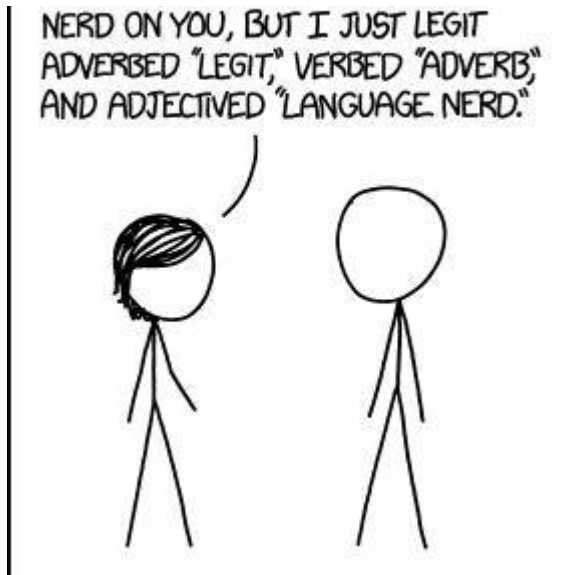
Differences

While speaking about the structure of languages, let's not forget to mention morphology. Morphology is the study of

words, their formation, their relationship with other words in the same language, as well as the ways context can change a word's pronunciation and meaning. While morphology plays a very important role in the analysis of human languages, we can't really say that programming languages have anything similar to it.

First of all, programming languages are artificial creations. This means that all of their rules and definitions were designed beforehand, which allows for them to be fully described and studied in their entirety. Their grammar is self-defining, and it doesn't change depending on the context. Think about it this way: every line of code has either zero or one meaning (in other words, it either contains an error or it's a valid program you can run). There are no synonyms (although Ruby's `.map` and `.collect` could be seen as synonyms, for example), allegories, analogies, historical or cultural references. Because of this, programming languages don't really have morphology, at least not the same way human languages do.





Source: Steve the vagabond and silly linguist

Second, due to the fact that they follow very strict set of rules, programming languages can't evolve and develop the same way human languages do (although we could say that programming languages evolve through various libraries). There's no room for errors or improvisation. On the other hand, human languages are full of imperfections. Just think of dialects, slang, jargon, argot (secret language used by a certain group that's incomprehensible to outsiders), namesake, accents, mispronounced words, typos, irregular punctuation and many other aspects of human languages that don't disrupt the message we're trying to communicate.





Source: Steve the vagabond and silly linguist

And the message human languages communicate is both logical and emotional. If spoken, it involves body language, intonation, volume, and many other nonverbal clues. In fact, languages are much defined by the physical attributes of human bodies (eyes, tongue, hands), and are for that reason unique to humans. None of these applies to programming languages. Although highly skilled programmers can develop their own styles of writing code, the nature of that code remains the same: logical, precise, perfectly unambiguous. Which shouldn't come as a surprise, since the foundation of all programming languages are billions and billions of 1s and 0s.

Why is this important?

Understanding the difference between programming languages and human languages is of great value for many fields that work on creating tools for important practical tasks such as machine translation, speech recognition, speech

synthesis, information extraction from text, grammar checking, text mining and more. Think of Google Translate or Siri, and try to understand everything it takes to turn a simple sentence in English into a message that can be processed by a computer. Now think of the first conversation in Spanish you had outside of class. It made you feel dizzy, didn't it?



when a native speaker switches the words around in a phrase you've memorized in a specific order



Source: Steve the vagabond and silly linguist