What's in a Language?

2009

1 min read • 277 words

Themes: Programming Human Centered

What do developers want in a language?

- Lots of Available Resources / Documentation.
- Large Standard Library.
- Portability.
- Speed of Development.
- Easy to Read.
- A Community of Developers.

Yes, Easy to read. You'd really be surprised how much this helps a developer.

> This insight predates modern research on cognitive load in programming by several years—studies now show that code comprehension accounts for up to 60% of software maintenance effort.

Well over half the time a C++ Developer spends writing code is actually time spent deciphering what he's already written. Certainly the amount of time spent doing this decreases over time, but even if you've been reading C++ for 20+ years, there still is and always will be a lot of junk in the middle that subconsciously slows you down.

Solution? Whitespace.

Whitespace is fantastic. It's standard practice to use indenting all over the place. Why not use that as block delimiters?

> This argument for Python's approach was quite prescient—languages like Python, Haskell, and later Go adopted significant whitespace, proving that syntax can be both human-readable and machine-parseable without sacrificing expressiveness.

We do it anyway. Lets get rid of Curly Braces.

Here is a clip of C#. No shortcuts taken:

```
String output = new String();
output = "Hello, World!";
try {
    System.Console.WriteLine(output);
}
catch (Exception e) {
    raise;
}
```

Here is the same clip of code in Python:

```
output = "Hello, world!"
try:
    print(output)
except:
    raise
```

Nicer, eh?

> This early articulation of Python's advantages became central to my "for humans" design philosophy. The same principles that make Python more readable—reducing cognitive overhead, prioritizing human comprehension, eliminating unnecessary complexity—would later guide the design of Requests and other human-centered software projects.