HW14 – PL feature survey

This HW is designed to help you search about different popular PLs, getting more familiar with them, and getting more comfortable with this activity – getting an overview of a PL before learning the details. Being able to quickly identify characters or features of a new PL will help you approach a new PL easier. For example, if you find out this PL has garbage collector by a quick 1 min search, you would know that you do not need to learn how to do memory release. You could also guess that this PL might have dynamically sized array or list that you could use easily. With a few searches, blogs reading, and Youtube video watching, you would be able to get a “feel” of a new PL and decide whether it might be a good candidate for your job at hand.

Please google a list of 10 most popular PL in this year, do some investigations about them, and fill in the following table. You do not need to find out an exact “yes/no” answer for questions like: “is this PL good for concurrency”. Just google and get a feel of what others judge this PL. You could write answers like “This PL supports concurrency via package XXX. But it seems people do not recommend it for serious concurrent programming because…”

**This HW is a group HW. Only 1 in 1 group (1-3 people) should submit. Make sure you write down your group members’ name in submission comment.**

**Not required but would be helpful for your own learning: Feel free to add more columns/topics (like year created, comprehensive tutorial available or not, naming rules, code samples, etc.) to the table (or create a new table below to have additional items in it) and make a more complete PL feature reference table for yourself to keep while getting to know the popular PLs, especially those that you never had the chance to learn about.**

**You could also expand your search to include a few more popular PLs. Examine the table you have. Do you see any trends? Is there any thing in common that you could find? What makes them to become popular PLs? Will those insights help you choosing a good PL to use in your future job?**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Name of PL | Supported platform  (Windows, Linux, etc.) | Paradigm(s) | Suitable tasks  (Web, database, general, etc.) | Complier or? | Popular IDE name | Type system | Good for concurrency | Garbage collector |
| Python | Windows, macOS | * OOP * Procedural * Functional | * Web dev * Scripting * Prototyping * Data science | interpreter | * Pycharm * IDLE * Visual Studio | Dynamic | Support concurrency but not recommended because multithreading is impeded by GIL | yes |
| JavaScript | Linux, Windows, macOS | * Functional * oop | * Mobile app dev * Front end dev * Backend dev * Game dev * Mobile apps * Database | interpreter | * VSCODE * Netbeans * Webstorm * eclipse | dynamic | Yes using eventloop | yes |
| java | Windows, macOS, Linux | Oop  imperative | * Desktop apps * Mobile dev * Web apps * Big data | Uses both compiler and interpreter | * Eclipse * IntelliJ | static | Yes but better to use Jython, Scala or Clojure | yes |
| C++ | MacOS, windows, linux | oop | * Operating systems * Game development * Device drivers * Embedded systems * Web browsers * Computer graphics | Compiler | * cLion * eclipse * VSCODE | static | yes | no |
| PHP | Linux  windows | * Imperative * Functional * Reflective * Oop * Procedural | * Cookie management * E commerce * Desktop * Content management systems | interpreter | * Netbeans * VSCODE * Rapid php | dynamic | No but can use third party libraries | yes |
| C# | MacOS, windows, linux | oop | * Desktop apps * Enterprise software * Game development * Cloud-based services * Mobile applications | Compiler | * Rider * VScode * Eclipse * atom | static | yes | No but can do with structs |
| Rust | Windows, linux, macos but needs Xcode for compiler | Multi-paradigm: concurrent, functional, generic, imperative, structured | * Operating systems * Game engines * Web browsers * Embedded systems | Compiler | * VScode * IntelliJ * Clion * Emacs | static | yes | no |
| Swift | Mac os, ios  TvOS, watchOS | Multi-paradigm: protocol-oriented, object-oriented, functional, imperative, block structured, declarative, concurrent | * Software development * MacOS and iOS applications | Compiler | * Xcode * Atom * SCADE * Apple developer tools | Strong type inference | yes | No but has ARC which for the most part deallocates memory |
| Ruby | Unix-like os such as Linux, macOS, windows | procedural, object-oriented, and functional programming | * Web development * Data Science * Automation * Command-line tools | interpreter | * Rubymine * Komodo * Cloud9 | dynamic | yes | yes |
| Go | Windows, android, linux, macOs, iOS | Multi-paradigm: concurrent imperative, object-oriented | * Web applications * Systems programming * Big data * Cloud computing * Machine Learning | Compiler | * GoLand * VSC * LiteIDE * Komodo | Inferred static strong | yes | yes |