**Pony for C# developers**

We all love C# as a high-level language, with good performance watched over by the .NET virtual machine in which it executes. It is highly applicable in a whole range of domains from desktop to web and mobile applications, and Visual Studio offers us almost the nirvana of Integrated Development Environments.

So why would you ever leave this safe heaven, and start doing projects with a new and still quite unknown language called Pony?

Pony is an open-source whose development started around 2012 at Imperial College in London. It fully recognizes that the future world of software development will have to focus on multicore and distributed applications, and that these applications will have to guarantee safety and correct execution (meaning: no data-races).

These are the two main focuses:

1) A highly-performant language with concurrency and parallel execution built-in as a first class goal. To achieve this, it implements the well-known actor model also used by Erlang and Akka, but with more guarantees and delivering better performance.

2) Safety, so that no Pony program will ever crash. It does this:

- by applying strong type guarantees

- implementing a concurrent, per-actor garbage collection

- not allowing any null or non-initialized values

- by supplementing the type system with additional qualifiers (so called *reference capabilities*) to make it safe to work on data with multiple actors, so as to avoid data-races.

These outstanding performance and guarantees are made possible by an ahead of time compiler, which generates native code. This eliminates the need for a heavy, resource consuming virtual machine on your target platform.

As we will see, a lot of care has been taken to make sure that the language has a clear, very readable and easy-to-use and learn syntax, avoiding unnecessary cruft of braces and semicolons.

And as a bonus for us OOP programmers, objects, classes and interfaces still play a part in Pony.

A first program.

To get a first feel for the syntax, let us compare the inevitable Hello-World program. Here is the C# version:

using System;

namespace HelloWorld

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Hello C# World!"); // output

Console.Read();

}

}

}

And here is the Pony version:

actor Main

new create(env: Env) =>

env.out.print("Hello, Pony world!")

We see a superficial resemblance: in both languages the action starts in something called Main. In C#, this is a static method from a class Program. In Pony however, Main is an actor which starts executing automatically in its create constructor. For safety reasons, Pony has no global variables, so the Main actor has to have a reference Env to its startup environment for program arguments and output, which is used here to print a string with env.out.print

What is immediately clear from this trivial example is how much easier and shorter the Pony code is, by using less protocol and avoiding curly braces as begin- and end-markers.

What can C# use for concurrency and how does Pony improve upon that?

Asynchronous programming: async / await

https://www.safaribooksonline.com/library/view/concurrency-in-c/9781491906675/ch01.html

https://msdn.microsoft.com/en-us/library/hh873175(v=vs.110).aspx

https://msdn.microsoft.com/en-us/library/dd460693(v=vs.110).aspx

https://code.msdn.microsoft.com/samples-for-parallel-b4b76364

For which projects would you apply Pony development?

It will excell especially for aplications where concurrency and distribution play a crucial role, such as:

- high-performance financial systems

- data analytics

- video games

- physical simulations

- cryptography

Where would you prefer Pony over C#?

C# is largely targeted and bound to Windows environments, with some possibilities of running your program on a Linux platform. It still needs the .NET framework to be present on the target machine, which restricts the portability in the development of C# programs by a large amount, and makes it more Microsoft Windows dependent. This makes it a no-go for real-time systems.

But Pony is written in C an so can run on any platform where C can run. Moreover it can interface very easily with C, as well as being called by C. Pony is optimized towards running million actors in a distributed real-time system.

* Version on Windows? http://tutorial.ponylang.org/getting-started/installation/
* Elementary types, strong typing
* Some control structures, exception handling
* Example with class and actor
* More about actors
* Arrays, Lists, Maps, enumeration types, union, pattern matching
* Ref capabilities
* Packages: use
* C FFI

Use commands are a similar concept to Python and Java "import", C/C++ "#include" and C# "using" commands, but not exactly the same.

**Can I just specify the full package path and forget about the use command, like I do in Java and C#?** No, you can't do that in Pony

i) example of a data-race + solution in C#

🡪 ii) solution in Pony, benchmark ?