Roadmap for Python developers learning C#

Article • 04/09/2024

C# and Python share similar concepts. These familiar constructs help you learn C# when you already know Python.

1. ***Object oriented***: Both Python and C# are object-oriented languages. All the concepts around classes in Python apply in C#, even if the syntax is different.
2. ***Cross-platform***: Both Python and C# are cross-platform languages. Apps written in either language can run on many platforms.
3. ***Garbage collection***: Both languages employ automatic memory management through garbage collection. The runtime reclaims the memory from objects that aren't referenced.
4. ***Strongly typed***: Both Python and C# are strongly typed languages. Type coercion doesn't occur implicitly. There are differences described later, as C# is statically typed whereas Python is dynamically typed.
5. ***Async / Await***: Python's `async` and `await` feature was directly inspired by C#'s `async` and `await` support.
6. ***Pattern matching***: Python's `match` expression and pattern matching is similar to C#'s pattern matching `switch` expression. You use them to inspect a complex data expression to determine if it matches a pattern.
7. ***Statement keywords***: Python and C# share many keywords, such as `if`, `else`, `while`, `for`, and many others. While not all syntax is the same, there's enough similarity that you can read C# if you know Python.

As you start learning C#, you'll learn these important concepts where C# is different than Python:

1. ***Indentation vs. tokens***: In Python, newlines and indentation are first-class syntactic elements. In C#, whitespace isn't significant. Tokens, like `;` separate statements, and other tokens `{` and `}` control block scope for `if` and other block statements. However, for readability, most coding styles (including the style used in these docs) use indentation to reinforce the block scopes declared by `{` and `}`.
2. ***Static typing***: In C#, a variable declaration includes its type. Reassigning a variable to an object of a different type generates a compiler error. In Python, the type can change when reassigned.
3. ***Nullable types***: C# variables can be *nullable* or *non-nullable*. A non-nullable type is one that can't be null (or nothing). It always refers to a valid object. By contrast, a

nullable type might either refer to a valid object, or null.

4. ***LINQ***: The query expression keywords that make up Language Integrated Query (LINQ) aren't keywords in Python. However, Python libraries like `itertools`, `more-itertools`, and `py-linq` provide similar functionality.
5. ***Generics***: C# generics use C# static typing to make assertions about the arguments supplied for type parameters. A generic algorithm might need to specify constraints that an argument type must satisfy.

Finally, there are some features of Python that aren't available in C#:

1. ***Structural (duck) typing***: In C#, types have names and declarations. Except for tuples, types with the same structure aren't interchangeable.
2. ***REPL***: C# doesn't have a Read-Eval-Print Loop (REPL) to quickly prototype solutions.
3. ***Significant whitespace***: You need to correctly use braces `{` and `}` to note block scope.

Learning C# if you know Python is a smooth journey. The languages have similar concepts and similar idioms to use.