Developer Tips: C# Tuples

By Andrew Hinkle

Tuples are simple data structures that store up to eight fields. Those fields can consist of any types from strings and ints, to objects, and even other Tuples. When you find yourself adding "out" parameters to your methods or creating a simple POCO or model to return multiple fields, you should consider using a Tuple. Use Tuples to simplify code and reduce the number of classes that need created. As of C# 7 named parameters and better performance have made them worthwhile using.

There are some limitations. Readable and maintainable code is important and many people prefer a well-defined model. As a rule of thumb if you are going to return more than three fields, review your method to determine if Tuples still make since or if a model makes the code easier to read. Using Tuples between MVC ViewModels and Views have a weird quirk. Read about all the Pros and Cons in the Resources links below.

The following are examples of how to use Tuples. Let's start by reviewing the older styles to the latest and greatest. These examples including some basic unit tests are stored in my GitHub Tips repository (Tips.Tuples) <https://github.com/penblade/Tips>.

# Tuple Class

If you prefer newing up the Tuple object directly, here's the code for you.



# Unnamed Tuples

Using parenthesis was the more common syntax for Tuples. Cleaner, except we still have that pesky Item1, Item2 usage. Notice that the example usage is identical to the Tuple class.





# Named Tuples

Named Tuples are AWESOME! As of C# 7, you could now give names to the fields in Tuples making them SO much more useful and self-documenting.



# Nested Tuples

Nested Tuples follows the same syntax. Just don't go crazy deep or you might confuse your C# code for LISP.



# Resources

* <https://docs.microsoft.com/en-us/dotnet/csharp/tuples>
  + Microsoft technical documentation on Tuples.
* <https://msdn.microsoft.com/en-us/library/dd383325(v=vs.110).aspx>
  + Overview on the older more explicit way of doing tuples which explains the limitations on the number of fields supported in a Tuple.
* <https://josephwoodward.co.uk/2017/04/csharp-7-valuetuple-types-and-their-limitations>
  + Great overview on how Tuples work especially in the IL. Joseph provides a great example with MVC ViewModels and the View. At compile time the named tuples are converted to the old style as Item1, Item2, etc., so while available they leave a horrible code smell. Read his article for more details on the Pros and Cons of Tuples.

# Summary

In summary, if you have been holding out on using Tuples like me due to the lack of naming fields or performance issues before C# 7, it's time to give them another chance.