Twitter Thread by Oleg Kyrylchuk



Oleg Kyrylchuk
@okyrylchuk

All my tweets about C# 10 features in one thread. ■

Retweets are appreciated ■

#dotnet #csharp #coding #devcommunity

Constant interpolated strings
https://t.co/JLB2IJXKBq

> C# 10 allows initializing 'const' strings using string interpolation, but the placeholder must also be a 'const' string.
>
> The placeholder can't be a numeric constant cause it's converted to string at runtime.
>
> Will you use the feature in your code?#dotnet #coding #devcommunity pic.twitter.com/H52ithdZC4
>
> — Oleg Kyrylchuk (@okyrylchuk) September 9, 2021

Extended property patterns
https://t.co/VTZqLNhPDr

> Starting from C# 10, you can reference nested properties or fields within a proper pattern.
>
> The property pattern becomes more readable and requires fewer curly brackets \U0001f642
>
> What do you think about the feature? #dotnet #coding #devcommunity pic.twitter.com/iHmbdYUAOW
>
> — Oleg Kyrylchuk (@okyrylchuk) September 10, 2021

File scoped namespaces
https://t.co/ApSQljwE6k

> C# 10 introduces a new way of namespace declarations - file scoped namespaces.
>
> However, you cannot declare a nested namespace or a second file-scoped namespace in the same file.
>
> C# continue to rid of curly brackets {\U0001f600}#dotnet #programming #devcommunity pic.twitter.com/docHOfZQYf
>
> — Oleg Kyrylchuk (@okyrylchuk) September 11, 2021

Global usings
https://t.co/F1HJlc3jZJ

> C# 10 adds a new modifier to a 'using' directive - 'global'.
>
> It means that 'using' is applied to all files in the compilation.
>
> All 'global using' directives must be before non-global 'using' directives.
>
> It can be combined with a 'static' modifier#dotnet #coding #devcommunity pic.twitter.com/osr3ZPh5AI
>
> — Oleg Kyrylchuk (@okyrylchuk) September 12, 2021

Assignment and declaration in the same deconstruction
https://t.co/9l3POMGfIb

> In previous versions of #csharp, a deconstruction could initialize newly declared variables or assign all values to existing variables.
>
> C# 10 can do both assignment and declaration in the same deconstruction \U0001f4aa#dotnet #coding #devcommunity pic.twitter.com/hrBM095jzF
>
> — Oleg Kyrylchuk (@okyrylchuk) September 13, 2021

Sealed when overriding ToString
https://t.co/NbuVBsR6Gp

> You couldn't add a 'sealed' modifier when you override 'ToString' method in the record type in C# 9.
>
> In #csharp 10, you can do that.
>
> As in classes, it forbids to override 'ToString' method in the derived record types.#dotnet #coding #devcommunity pic.twitter.com/h4UB7BlmDS
>
> — Oleg Kyrylchuk (@okyrylchuk) September 14, 2021

Record struct and class declarations
https://t.co/bAqMYoiXep

#csharp 10 introduces a record struct. You can declare it with 'record struct' keywords.

It also adds new syntax for a record class with 'record class' keywords.

A record struct has all struct properties. A record class has all class properties.#dotnet #coding #devcommunity pic.twitter.com/fRR01507ej

— Oleg Kyrylchuk (@okyrylchuk) September 15, 2021

Struct vs Record struct
https://t.co/MKOoFPqsmT

Yesterday I tweeted about the new 'record struct' type in C# 10. I showed the way of declaration.

Let's see today the difference between the new 'record struct' and old fashioned C# 'struct' \U0001f642

Which do you like the most? Reply with the number \U0001f609#dotnet #coding #devcommunity pic.twitter.com/RaVRxN3NLE

— Oleg Kyrylchuk (@okyrylchuk) September 16, 2021

Parameterless struct constructors
https://t.co/iYdgMH1hZQ

Previous C# versions don't support field initializers in the structs.

C# 10 fixes it and closes the gap between struct and class declarations.

If the struct has field initializers, the compiler will synthesize a public parameterless constructor.#dotnet #coding #devcommunity pic.twitter.com/5XVfU0xQTS

— Oleg Kyrylchuk (@okyrylchuk) September 17, 2021

Lambdas with attributes
https://t.co/rAGZJ9II0n

C# 9 has allowed attributes on local functions.

C# 10 allows attributes on lambda expressions and lambda parameters.

To distinguish expression attribute from parameter attribute, you must use a parenthesized parameter list for lambda expression.#dotnet #coding #devcommunity pic.twitter.com/ITYL1ZBL6c

— Oleg Kyrylchuk (@okyrylchuk) September 19, 2021

Explicit return type in lambdas
https://t.co/tSzXUnxZ3P

In C# 10, you can specify an explicit return type for lambdas.

Explicit return types are not supported for anonymous methods declared with 'delegate { }' syntax.#dotnet #coding #devcommunity pic.twitter.com/jcmeoHgwGj

— Oleg Kyrylchuk (@okyrylchuk) September 20, 2021

AsyncMethodBuilder attribute applying to the method
https://t.co/BhndpaKVsu

Since C# 7, you can apply the AsyncMethodBuilder attribute to a type only.

In C# 10, you can also apply the attribute to a single method.

Disclaimer: the example just shows the applying of attribute \U0001f642#dotnet #coding #devcommunity pic.twitter.com/4QLrz3Tedm

— Oleg Kyrylchuk (@okyrylchuk) September 22, 2021

Expression 'with' in structs
https://t.co/8DfXHPOxKo

C# 9 has introduced 'with' expression for record types.

It produces a copy of its operand with the specified properties and fields modified.

In C# 10 you can also use 'with' expression with struct types \U0001f60e#dotnet #coding #devcommunity pic.twitter.com/709NcFVclq

— Oleg Kyrylchuk (@okyrylchuk) September 23, 2021