

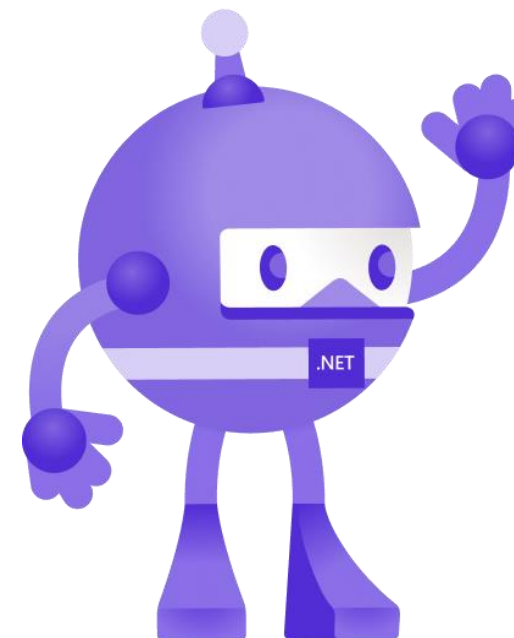


Gábor Szabó
Assistant Lecturer
Microsoft Evangelist
Disruptive Software Architect
[linkedin.com/in/gabor-szabo/](https://www.linkedin.com/in/gabor-szabo/)

.NET 6, C# 10 és ezeken túl

Mit hoz a jövő?

2021. november 2.



Visual Studio 2022 Launch Event

Monday November 8, 2021 | 8:30am PT

6 days until the Visual Studio 2022 launch

[Sign up for email updates](#)

[Add event to calendar >](#)

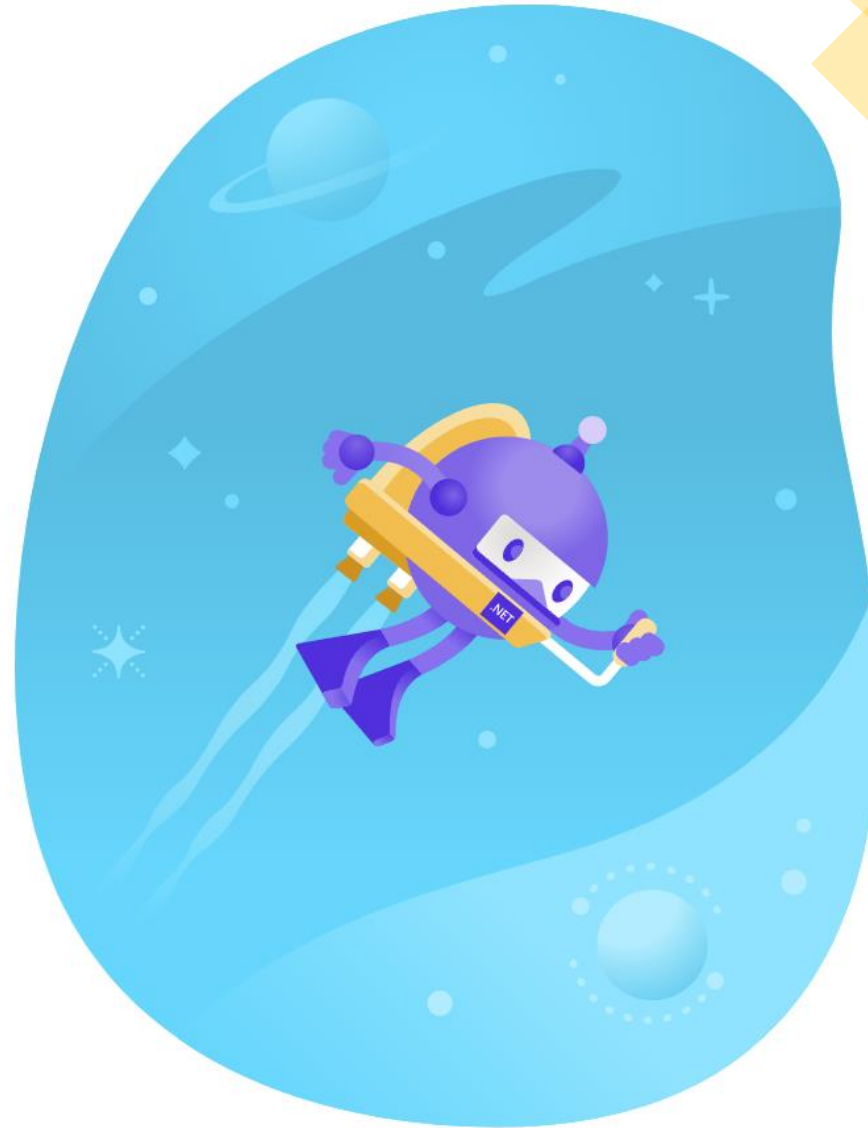
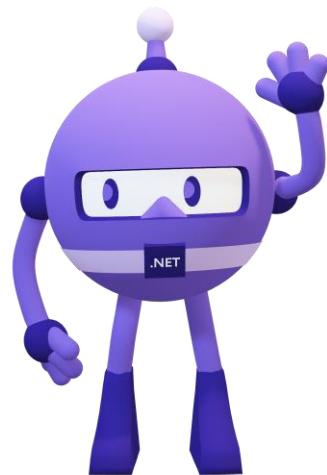


.NET Conf 2021

November 9-11

6 days 23 hours 0 minutes

Save the Date!



StackOverflow fejlesztői kérdőívek 2021

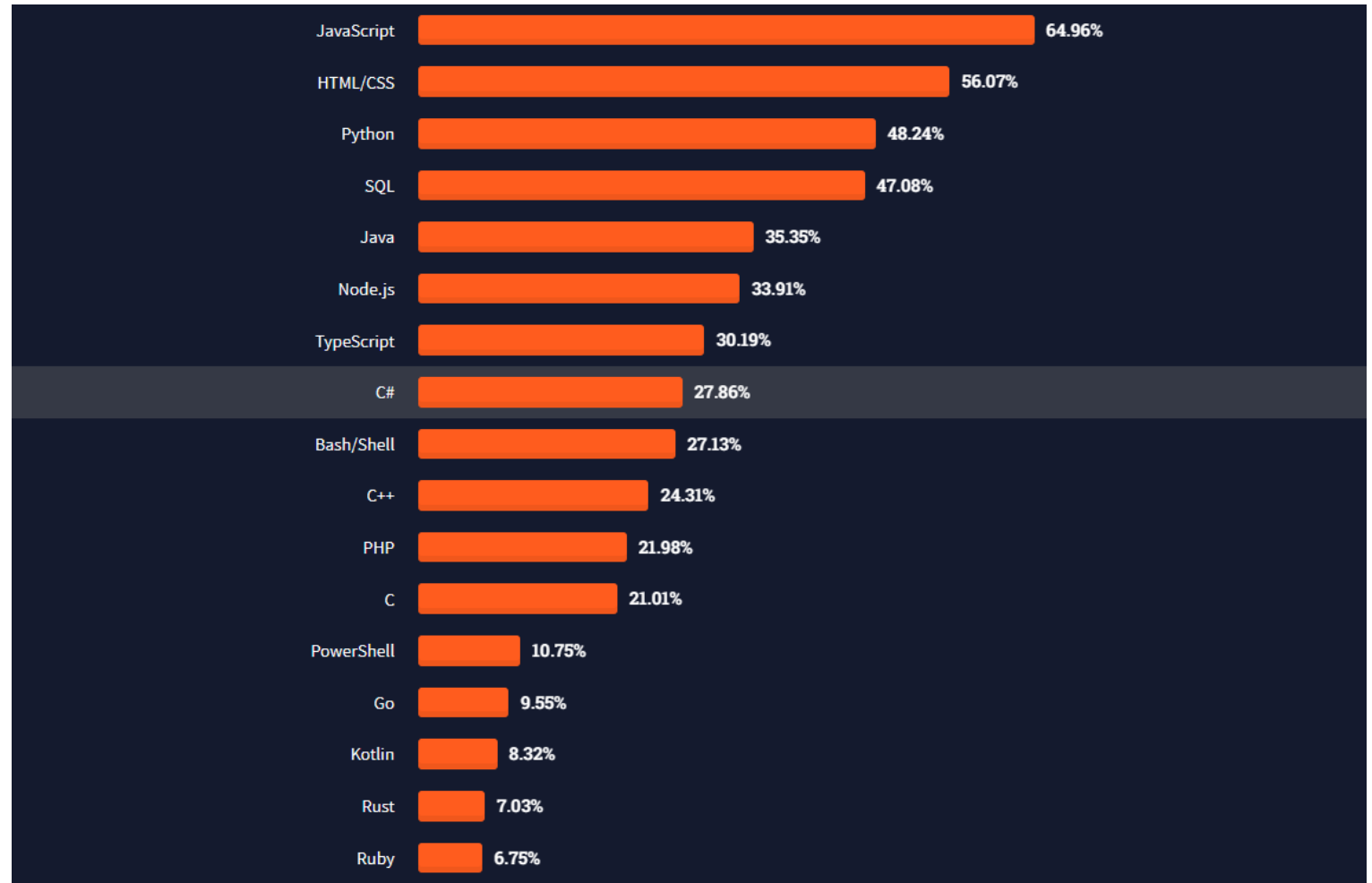
Hol van a .NET és a C#
kb. 80 000 fejlesztő válasza alapján?

<https://insights.stackoverflow.com/survey/2021>



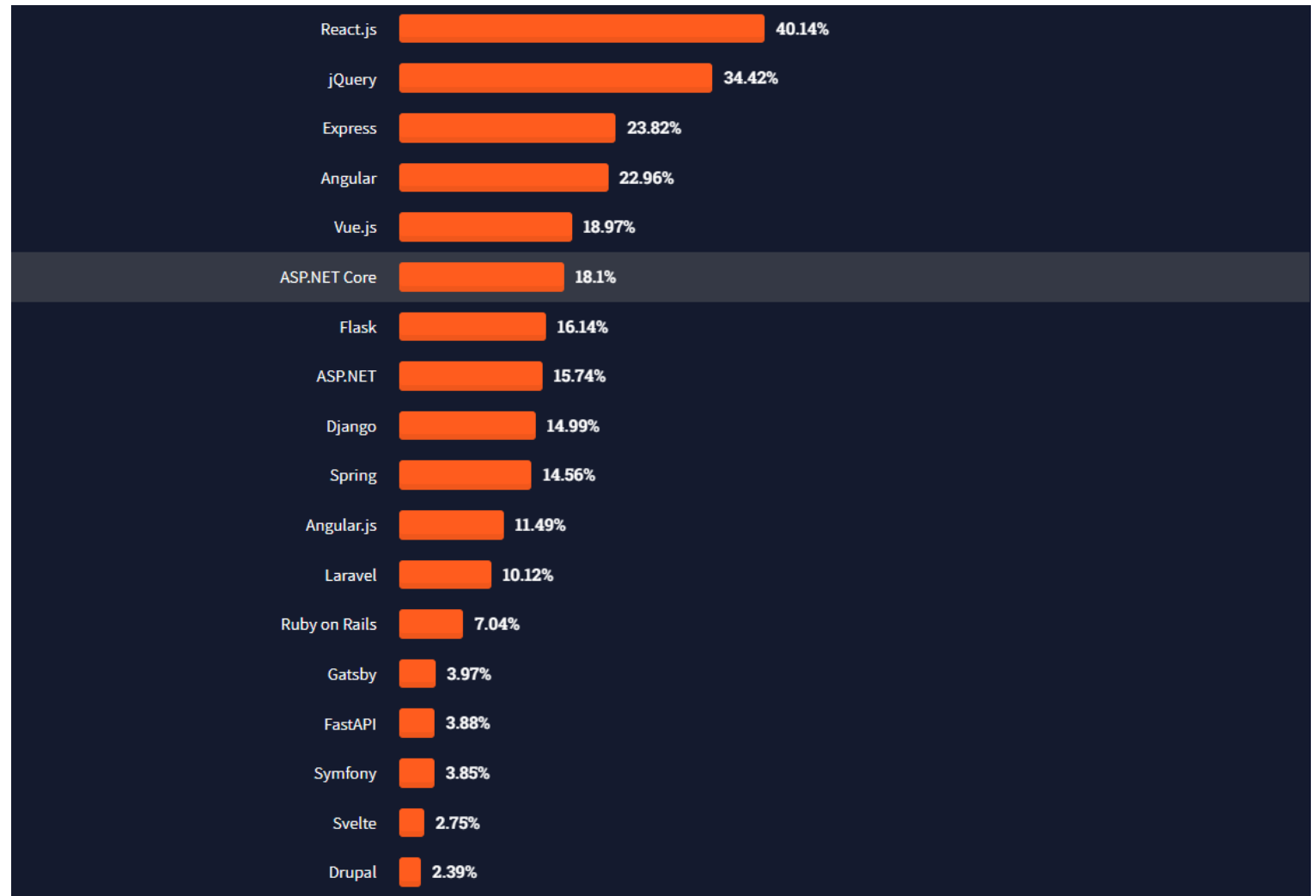
Which programming, scripting, and markup languages

have you done extensive development work in over the past year, and which do you want to work in over the next year? (If you both worked with the language and want to continue to do so, please check both boxes in that row.)



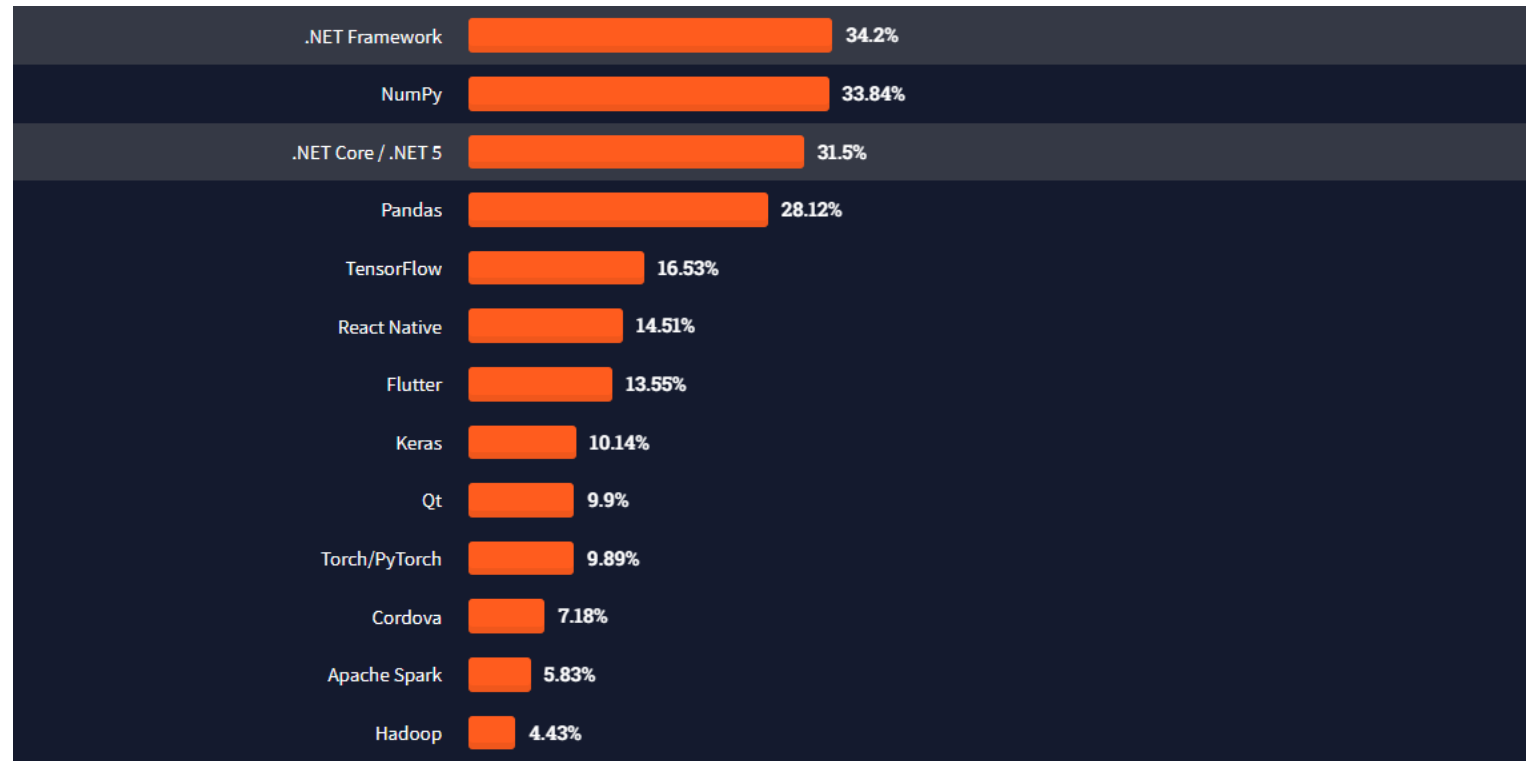
Which web frameworks and libraries

have you done extensive development work in over the past year, and which do you want to work in over the next year? (If you both worked with the framework and want to continue to do so, please check both boxes in that row.)



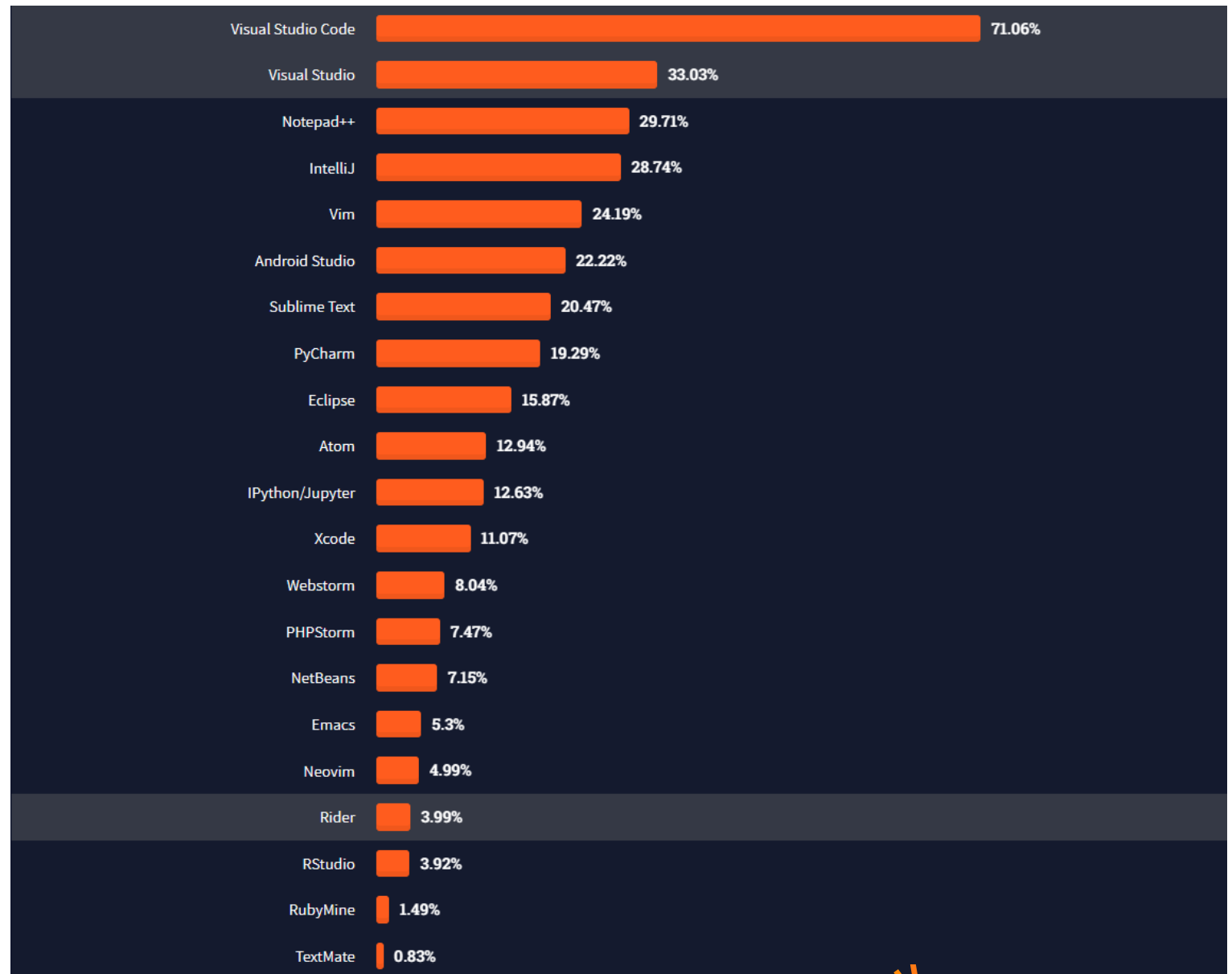
Which other frameworks and libraries

have you done extensive development work in over the past year, and which do you want to work in over the next year? (If you both worked with the framework and want to continue to do so, please check both boxes in that row.)



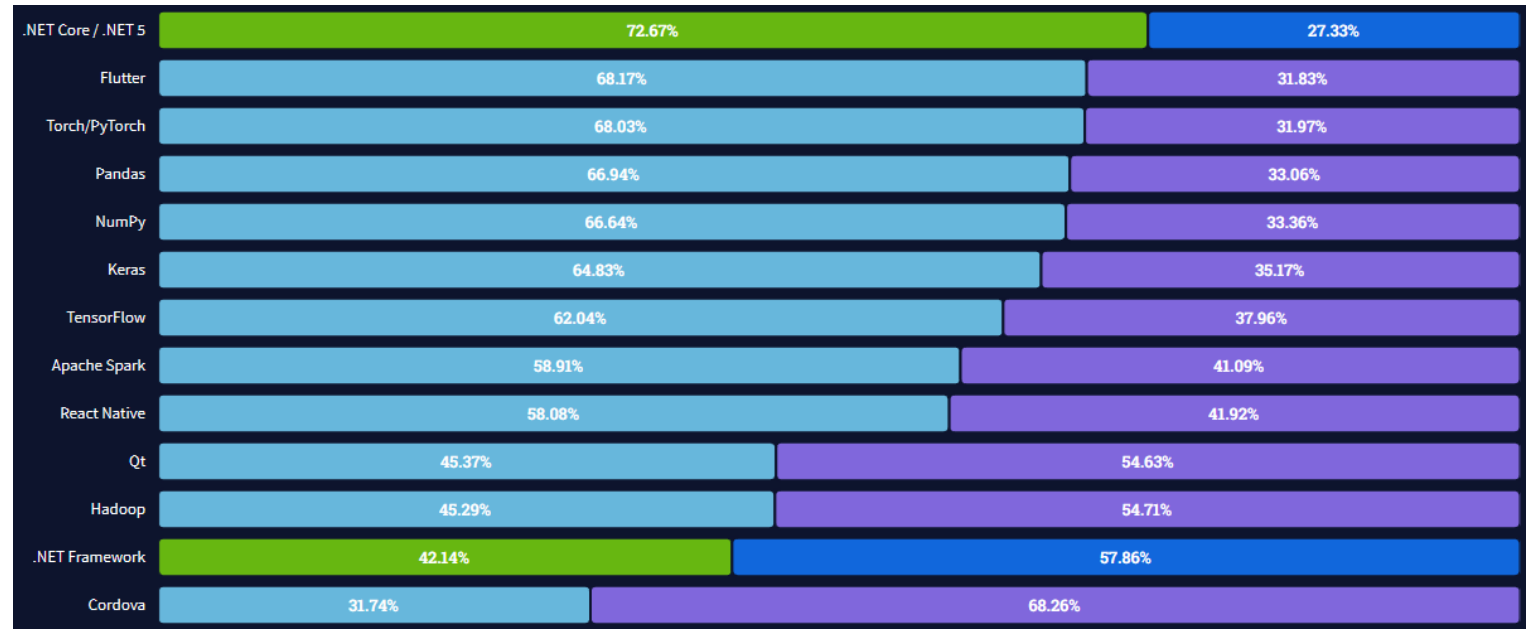
Which development environments

did you use regularly over
the past year, and which do
you want to work with over
the next year? Please check
all that apply.



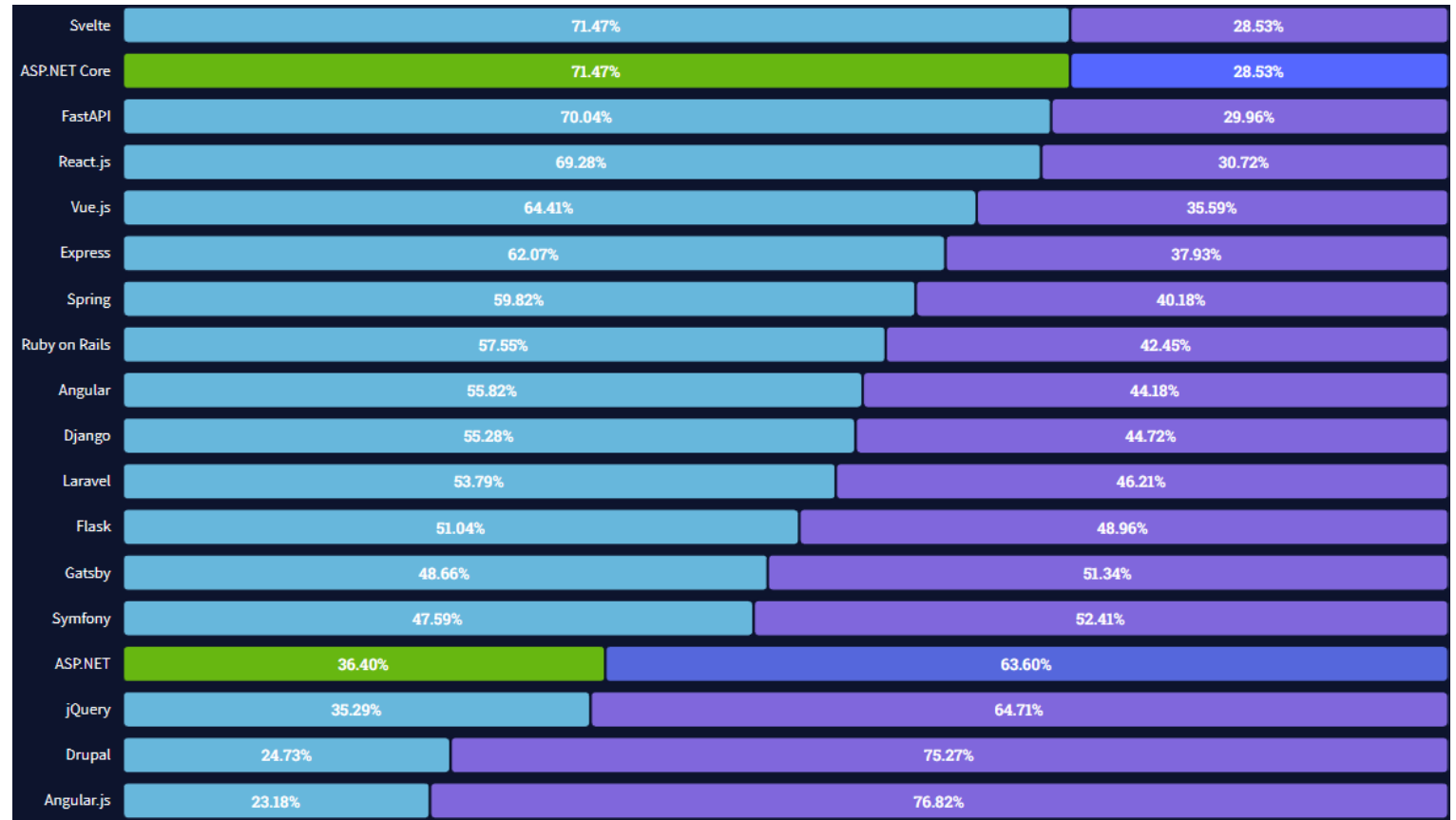
Which programming, scripting, and markup languages

have you done extensive
development work in over
the past year, and which do
you want to work in over
the next year? (If you both
worked with the language
and want to continue to do
so, please check both boxes
in that row.)



Which web frameworks and libraries

have you done extensive development work in over the past year, and which do you want to work in over the next year? (If you both worked with the framework and want to continue to do so, please check both boxes in that row.)



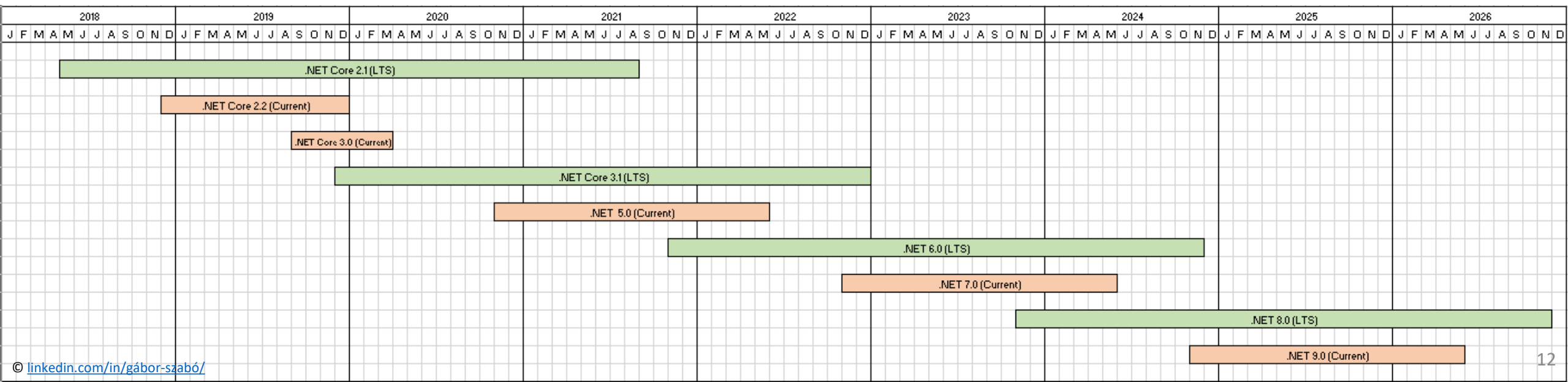
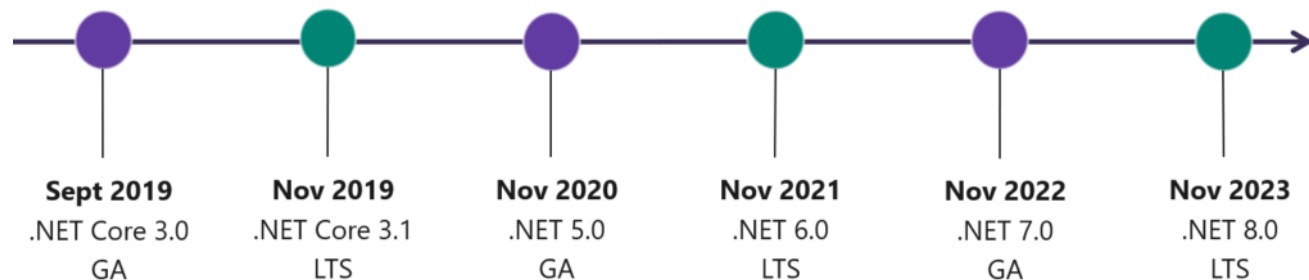
Microsoft .NET

2002-_____

1997	Visual Studio
2002	.NET Framework 1.0, WinForms, ADO.NET, ASP.NET
2006	.NET 2.0, C# 2.0, VS2005: generikus típusok, anonim metódusok, <code>foreach</code> , <code>Nullable<T></code>
2006	.NET 3.0, C# 3.0, VS2008, WPF, WCF, WF, lambdák/kifejezésfák, <code>var</code> , anonim típusok, bővítő metódusok, inicializálók
2007	.NET 3.5: LINQ
2008	.NET 3.5 SP1: ADO.NET Entity Framework
2009	ASP.NET MVC
2010	.NET 4.0, C# 4.0, VS2010, PLINQ, TPL, dynamic
2012	.NET 4.5, C# 5.0, VS2012, <code>async/await</code>
2014	.NET Core bejelentése, Roslyn, .NET Foundation
2015	C# 6.0, <code>? .</code> , auto-property inicializálók, kifejezés törzsű metódusok, string interpoláció, <code>using static</code>
2016	.NET Core 1.0, ASP.NET Core, EF Core, CoreCLR, RyuJIT
2017	.NET Core 2.0, C# 7.0, <code>Span<T></code> és társai, <code>out var</code> , <code>_</code> , ennesek, dekonstrukció, mintaillesztés, helyi függvények, kifejezés törzsek++
2019	.NET Core 3.0, .NET Framework 4.8, C# 8.0, Blazor Server, <code>System.Text.Json</code> , default interfész metódusok, mintaillesztés++, nullable referencia típusok, <code>[1..^1]</code>
2020	.NET 5, C# 9.0, rekord osztályok, legfelsőbb szintű utasítások, <code>new()</code>
2021+	<code>var isLTS = DateTime.Now.AddMonths(-10).Year - 2015 is var dotnet && dotnet % 2 == 0;</code>

Kiadások

- **Minden novemberben** új főverzió
 - Alverzió nem tervezett
- **LTS**: páros kiadások, 3 év support
- **Current**: páratlan kiadások, 18 hónap support
- Felzárkóztatás:
 - „Seekers”: 1 év
 - „Prospectors”: 2 év
 - „Laggers”: 3 év



C# 10 újdonságok

Avagy a kevesebb sokszor több





File-scoped **namespace** declaration

```
namespace MyProject  
{  
    public class MyClass { }  
}
```

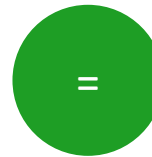


```
namespace MyProject;  
public class MyClass { }
```

C# global using

```
// Pl. "GlobalUsings.cs":
```

```
global using System;  
global using static System.Console;  
global using MyNamespace;
```



```
<!-- Pl. MyProject.csproj -->
```

```
<Project Sdk="Microsoft.NET.Sdk">
```

```
  <PropertyGroup>
```

```
    <OutputType>Exe</OutputType>
```

```
    <TargetFramework>net6.0</TargetFramework>
```

```
  </PropertyGroup>
```

```
  <ItemGroup>
```

```
    <Using Include="System" />
```

```
    <Using Include="System.Console" Static="True" />
```

```
    <Using Include="MyNamespace" />
```

```
  </ItemGroup>
```

```
</Project>
```

```
// Program.cs:
```

```
WriteLine("Hello, C# global usings!");
```

C# implicit global using

```
<!-- Pl. MyProject.csproj -->

<Project Sdk="Microsoft.NET.Sdk">

  <PropertyGroup>
    <OutputType>Exe</OutputType>
    <TargetFramework>net6.0</TargetFramework>
    <ImplicitUsings>enable</ImplicitUsings>
    <Nullable>enable</Nullable> <!-- .NET 6-tól alapértelmezett! -->
  </PropertyGroup>

</Project>

// See https://aka.ms/new-console-template for more information
Console.WriteLine("Hello, World!");

// ^^ Az alapértelmezett konzol app sablon Program.cs-je.
// Nincs "using System;" és a "Console" nem FQN!
// Használható await és args is
```

SDK	Default namespaces
Microsoft.NET.Sdk	System
	System.Collections.Generic
	System.IO
	System.Linq
	System.Net.Http
	System.Threading
	System.Threading.Tasks
Microsoft.NET.Sdk.Web	System.Net.Http.Json
	Microsoft.AspNetCore.Builder
	Microsoft.AspNetCore.Hosting
	Microsoft.AspNetCore.Http
	Microsoft.AspNetCore.Routing
	Microsoft.Extensions.Configuration
	Microsoft.Extensions.DependencyInjection
	Microsoft.Extensions.Hosting
Microsoft.NET.Sdk.Worker	Microsoft.Extensions.Logging
	Microsoft.Extensions.Configuration
	Microsoft.Extensions.DependencyInjection
	Microsoft.Extensions.Hosting



Demó app: konzol szintetizátor

```
var tempo = 100;
while (true)
    if (char.ToLower(Console.ReadKey().KeyChar) is var keyChar && keyChar is >= 'a' and <= 'z')
        Console.Beep((int)(440 * Math.Pow(2, ((double)(keyChar - 'n')) / 12.0)), tempo);
    else if (keyChar is '-' or '+')
        tempo = keyChar is '-' ? Math.Max(25, tempo - 25) : tempo + 25;
    else
        await Task.Delay(tempo);

/*
 8 sor; 357 karakter!
 Smoke on the Water dallam:
 l+ -o+ ++++q----- -l+ -o+ --r- ++++++++q----- -l+ -o+ ++++q----- -o+ ++++++++l----- ---
*/
```





const && \$" {string} "

```
const string Major = "6";
```

```
const string Minor = "0";
```

```
const string Patch = "0";
```

```
const string Version = $" {Major} . {Minor} . {Patch} ";
```



record struct

```
Battery battery = new("CR2032", 0.235, 100);

WriteLine(battery);
// Battery { Model = CR2032, TotalCapacityAmpHours = 0.235, RemainingCapacityPercentage = 100 }

while (battery.RemainingCapacityPercentage > 0)
    battery.RemainingCapacityPercentage--; // Alapértelmezetten mutábilis

WriteLine(battery);
// Battery { Model = CR2032, TotalCapacityAmpHours = 0.235, RemainingCapacityPercentage = 0 }

public record struct Battery(string Model, double TotalCapacityAmpHours, int RemainingCapacityPercentage);
```

- Lehet **readonly**
 - Alapértelmezetten módosítható, mert a **struct** egyébként is másolódik a stracken
- A normál **record** most már **record class**
- **ToString()**, ekvivalenciavizsgálat megegyezik (értékek szerint hasonlít)
- A **with** itt is használható új példány létrehozáshoz másolással



extended property patterns

```
Battery battery = new("CR2032", 0.235, 100);

var isBatteryNameSixCharsLong = battery switch
{
    { Model.Length: 6 } => true,
    // korábban: { Model: { Length: 6 } } => true,
    _ => false
};

public record struct Battery(string Model,
                             double TotalCapacityAmpHours,
                             int RemainingCapacityPercentage);
```



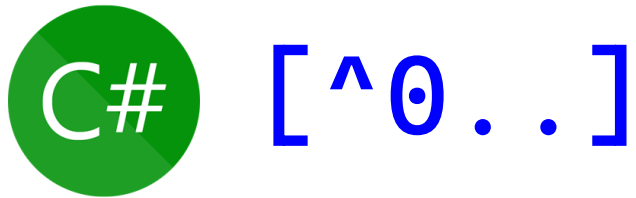
[0..⁻¹]

```
var s1 = new MyStruct(1);  
var s2 = s1 with { X = 2 }; // "with" structokkal is ✓
```

```
[MyGeneric<int>]  
struct MyStruct  
    : IMyInterface  
{  
    public int X;  
    public MyStruct(int x) => X = x;  
    public MyStruct() => X = 1; // Paraméter nélküli konstruktor ✓  
    public static void DoSomething(int x!) { } // Null check, tervben  
}
```

```
class MyGenericAttribute<T> : Attribute { } // preview
```

```
interface IMyInterface  
{  
    abstract static void DoSomething(int x); // preview  
}
```



- Egyedi `InterpolatedStringHandler`
- Forráskód-generátorok v2+
- Tuple kevert értékadás/deklaráció dekonstrukcióban
- Lambdák:
 - Explicit visszatérés
 - Attribútumok lambdákon
 - `Func/Action` típus kikövetkeztetése (`var` és metóduscsoport esetén)
- Majd:
 - „*Extension everything*”
 - Kötelező `init` tulajdonságok
 - Property `field` kulcsszó
 - Uniótípusok
 - Generikus matek (`abstract static` interfészen)
 - Elsődleges konstruktorok (nem csak `record` típusok esetében)

ASP.NET Core 6 újdonságok

Mindig van új a Nap alatt...



- Webes kliensalkalmazások részére
- HTML, CSS és C#
 - JavaScript nélkül (!)*
 - .NET-ben fut (!)
- WebAssembly-vel böngészőben futtatható: **Blazor WebAssembly**
- Vagy a kliensek futtathatók *a szerveren*: **Blazor Server**
 - Nem, nem typo :)
 - .NET 5 óta RTM
- Lehetséges app részenként, funkciónként is áttérni
- Könnyen integrálódik Razor Pages-be

Core Blazor @

```
@page "/counter"
```

```
<h1>Counter</h1>
```

```
<h3>Current count: @currentCount</h3>
```

```
<button class="btn btn-primary"
```

```
  @onclick="IncrementCount">
```

```
Click me
```

```
</button>
```

```
@code {
```

```
    private int currentCount = 0;
```

```
    private void IncrementCount()
```

```
    {
```

```
        currentCount++;
```

```
    }
```

```
}
```

MyBlazorApp

Counter

Current count: 0

Click me

ASP.NET Core Új projektsablonok

```
// Web App: Empty
var builder = WebApplication.CreateBuilder(args);
var app = builder.Build();

app.MapGet("/", () => "Hello World!");

app.Run();
```

```
// Web App (Razor Pages)
var builder = WebApplication.CreateBuilder(args);

// Add services to the container.
builder.Services.AddRazorPages();

var app = builder.Build();

// Configure the HTTP request pipeline.
if (!app.Environment.IsDevelopment())
{
    app.UseExceptionHandler("/Error");
    // The default HSTS value is 30 days. You may want to change this for production scenarios, see
    // https://aka.ms/aspnetcore-hsts.
    app.UseHsts();
}

app.UseHttpsRedirection();
app.UseStaticFiles();

app.UseRouting();

app.UseAuthorization();

app.MapRazorPages();

app.Run();
```

```
// Web App (MVC)
var builder = WebApplication.CreateBuilder(args);

// Add services to the container.
builder.Services.AddControllersWithViews();

var app = builder.Build();

// Configure the HTTP request pipeline.
if (!app.Environment.IsDevelopment())
{
    app.UseExceptionHandler("/Error");
    // The default HSTS value is 30 days. You may want to change this for production scenarios, see
    // https://aka.ms/aspnetcore-hsts.
    app.UseHsts();
}

app.UseHttpsRedirection();
app.UseStaticFiles();

app.UseRouting();

app.UseAuthorization();

app.MapControllerRoute(
    name: "default",
    pattern: "{controller=Home}/{action=Index}/{id?}");

app.Run();
```



Új projektsablonok

```
// Web API (Minimal API)
var builder = WebApplication.CreateBuilder(args);

// Add services to the container.
// Learn more about configuring Swagger/OpenAPI at
// https://aka.ms/aspnetcore/swashbuckle
builder.Services.AddEndpointsApiExplorer();
builder.Services.AddSwaggerGen();

var app = builder.Build();

// Configure the HTTP request pipeline.
if (app.Environment.IsDevelopment())
{
    app.UseSwagger();
    app.UseSwaggerUI();
}

var summaries = new[]
{
    "Freezing", "Bracing", "Chilly", "Cool", "Mild",
    "Warm", "Balmy", "Hot", "Sweltering", "Scorching"
};
```

```
app.MapGet("/weatherforecast", () =>
{
    var forecast = Enumerable.Range(1, 5).Select(index =>
        new WeatherForecast
        (
            DateTime.Now.AddDays(index),
            Random.Shared.Next(-20, 55),
            summaries[Random.Shared.Next(summaries.Length)]
        ))
        .ToArray();
    return forecast;
})
.WithName("GetWeatherForecast");

app.Run();

internal record WeatherForecast(DateTime Date,
                                int TemperatureC, string? Summary)
{
    public int TemperatureF => 32 + (int)(TemperatureC / 0.5556);
}
```

NET 6 újdonságok

MÉG több == MÉG több





LINQ újdonságok

```
var source = Enumerable.Range(1, 10);
```

```
source.ElementAt(^2); // 9
```

```
source.Take(2..7); // 3, 4, 5, 6, 7 | source.Take(7).Skip(2)
```

```
source.Take(^7..^3); // 4, 5, 6, 7 | source.TakeLast(7).SkipLast(3)
```

```
WriteLine($"{source.TryGetNonEnumeratedCount(out int count)} - {count}"); // True - 10
```

```
source.DistinctBy(x => x % 3); // 1, 2, 3
```

```
source.MinBy(x => DateTime.DaysInMonth(DateTime.Today.Year, x)); // 2
```

```
source.Chunk(3).Select(ch => string.Join(",", ch)); // 1,2,3 4,5,6 7,8,9 10
```

```
source.SingleOrDefault(5); // 5
```



Egyéb újdonságok kulcsszavakban

- Új kiinduló projektsablonok, kevesebb boilerplate
- Hot reload
- Single-file fejlesztések (tömörítés, statikus kódelemzők)
- Apple Silicon, ARM64 támogatás
- Tömerdek low- és mid-level runtime optimalizáció (ld. string interpoláció optimalizálás 5000 szóban)
- Profilvezérelt optimalizáció (PGO), build idő csökkentés, AOT
- Mono runtime is bevonva a lehetséges runtime targetek közé (pl. Blazor WebAssembly, MAUI, Xamarin)
- Nagyteljesítményű naplózás kódgenerálással (`Microsoft.Extensions.Logging`)
- Nagyteljesítményű JSON sorosítás kódgenerálással (`System.Text.Json`)
- Új kollekciótípus: `PriorityQueue<TElement, TPriority>`
- Új hardveresen gyorsított `System.Math` API-k
- `System.Text.Json` írható DOM
- Új `System.DateOnly` és `System.TimeOnly` típusok
- Új biztonsági (W^X, CET) és teljesítmény funkciók támogatása (HTTP/3)
- Új DI szolgáltatás: `Microsoft.Extensions.DependencyInjection.IServiceProviderIsService`



ASP.NET Core újdonságok kulcsszavakban

- Új Razor compiler az új forráskód-generátorok alapján
- Generikus típusmegkötések Razorben
- CSS izoláció klasszikus Razor nézetekben is
- View component tag helperek opcionális paraméterekkel
- `Microsoft.AspNetCore.Http.Results` utility Minimal API-khoz
- Nullable annotációk kezelése
- HTTP/3 és QUIC támogatás a Kestrelben (preview)
- Hot reload
- Biztonsági fejlesztések és teljesítmény optimalizációk
- Új HTTP logging middleware
- Kestrel az alapértelmezett futtatási mód IIS Express helyett
- Új általános SPA template
- WebSocket tömörítés támogatás
- Kestrel egyedi HTTP header encoding támogatás
- Új (**IdentityServer4** > **Duende**) IdentityServer licensz, továbbra is template-ben

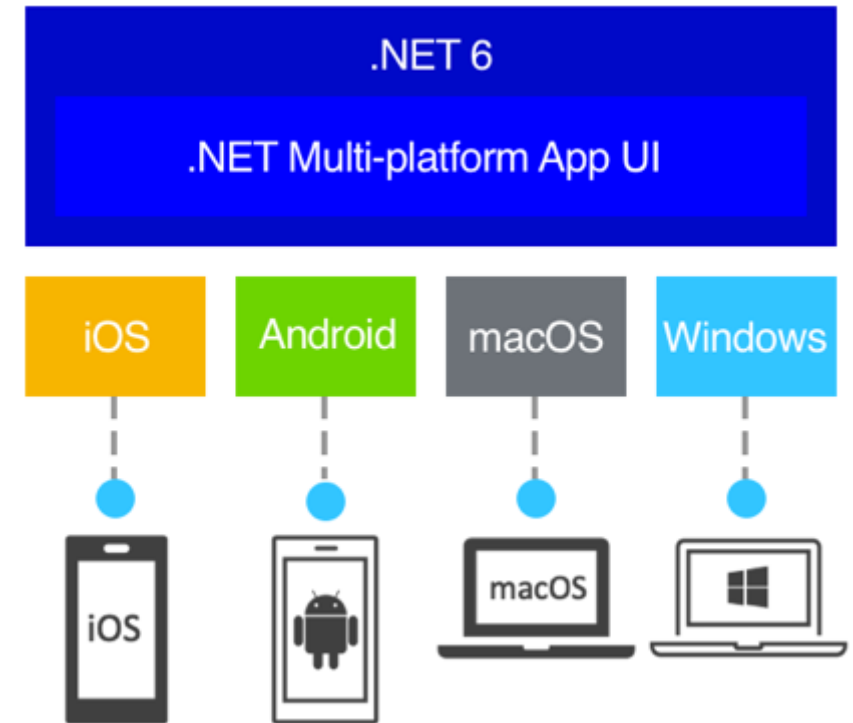
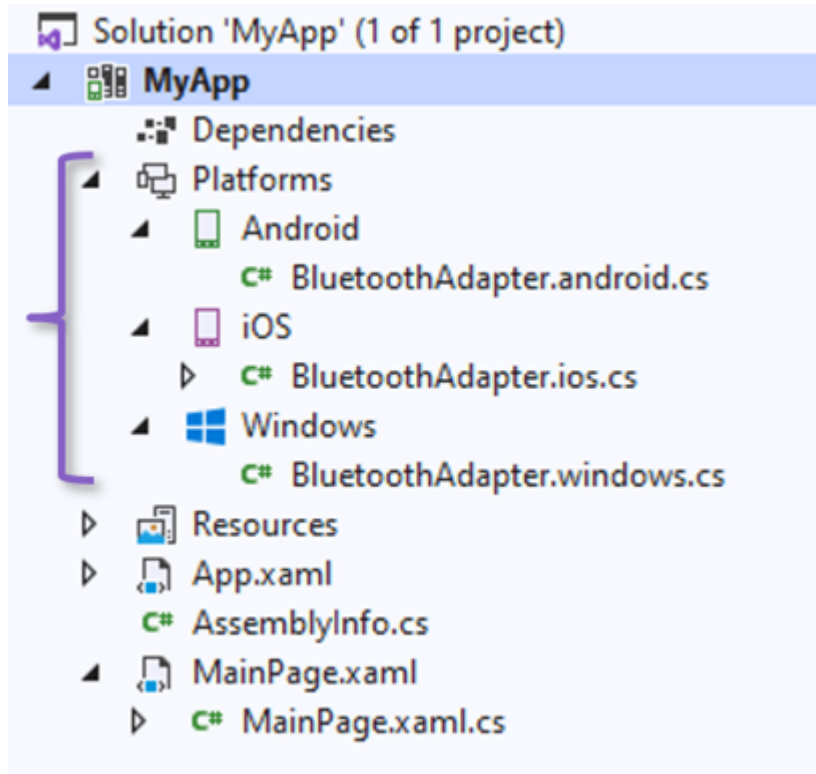
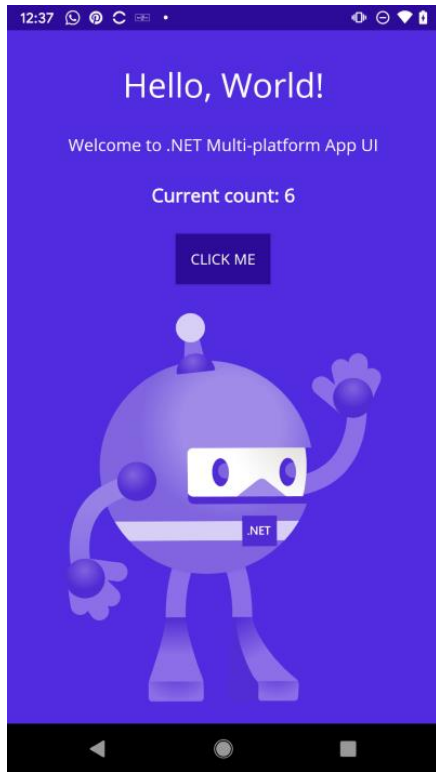


EF Core újdonságok kulcsszavakban

- Fordított (compiled) modell: 10x teljesítmény, gyors indulás nagy modellnél
- Migrációs csomagok (migration bundles): DB update .exe fájl
- Új attribútumok: **Unicode**, **Precision**, **EntityTypeConfiguration**
- Új **DateOnly** és **TimeOnly** típusok támogatása
- Nullable referenciatípusok támogatása
- Új **EF.Functions**, SQL fordítás lehetőségek: **Random()**, **string.Concat()**
- SQL Server: ritka oszlopok, temporális táblák
- In-Memory DB: kötelezőségi validáció
- Scaffolding: DB kommentekből kód kommentek, több-több kapcsolatok
- Teljesítményjavítások (30-70%)



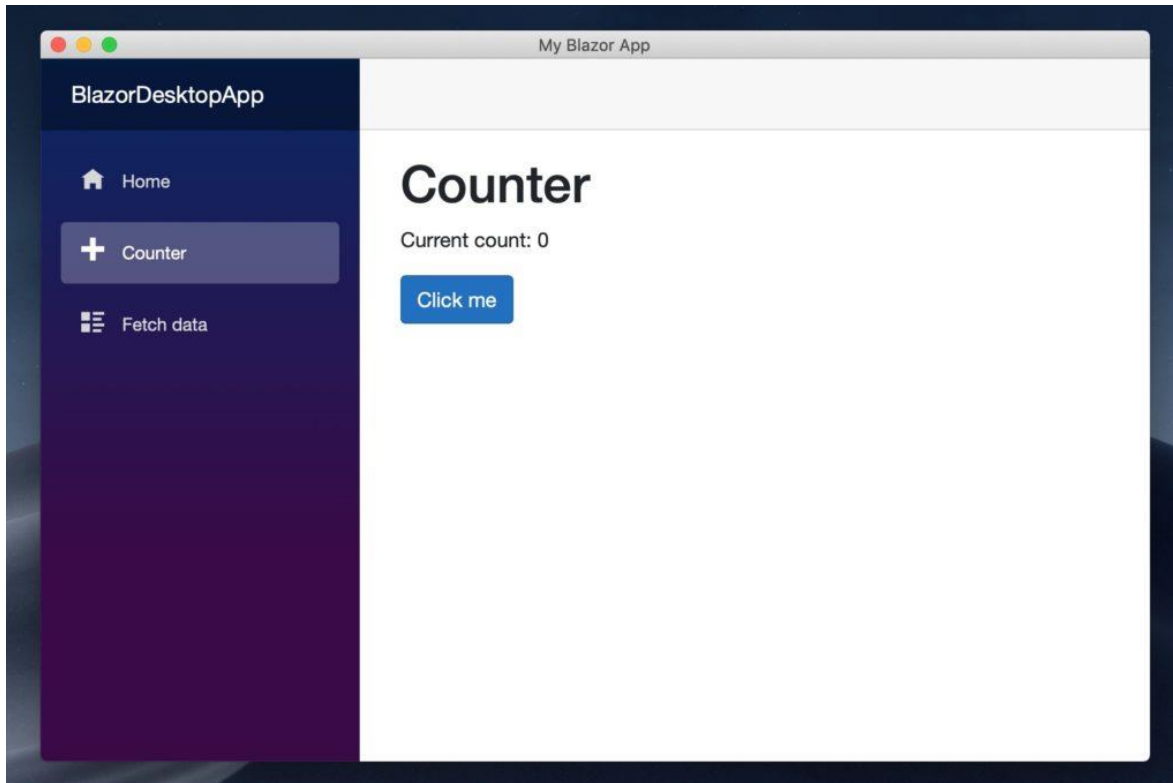
.NET Multi-platform App UI: MAUI



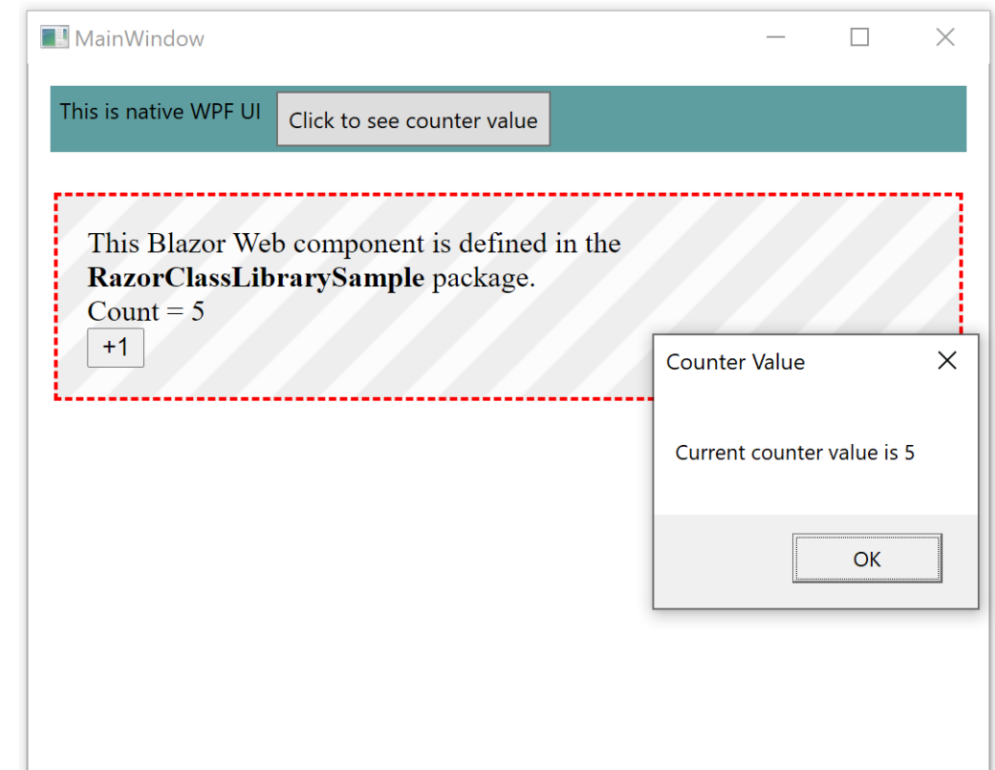
MAUI ETA: 2022 Q2



MAUI + Blazor?



```
<BlazorWebView
  HostPage="wwwroot/index.html"
  Services="{StaticResource Services}">
  <BlazorWebView.RootComponent>
    <RootComponent
      Selector="#app"
      ComponentType="{x:Type local:Main}"
    />
  </BlazorWebView.RootComponent>
</BlazorWebView>
```



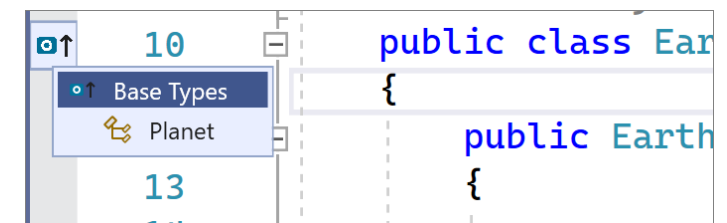
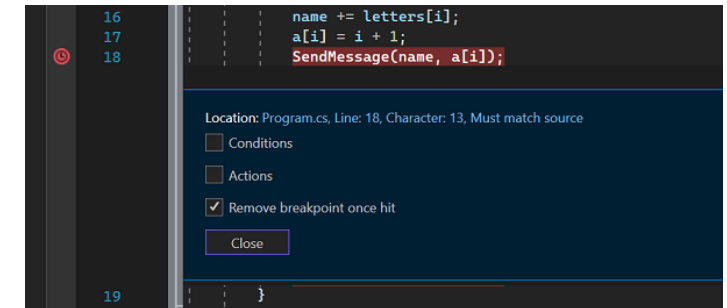
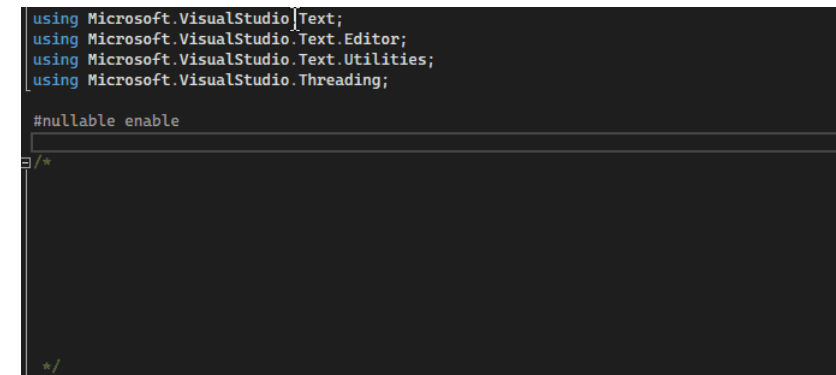
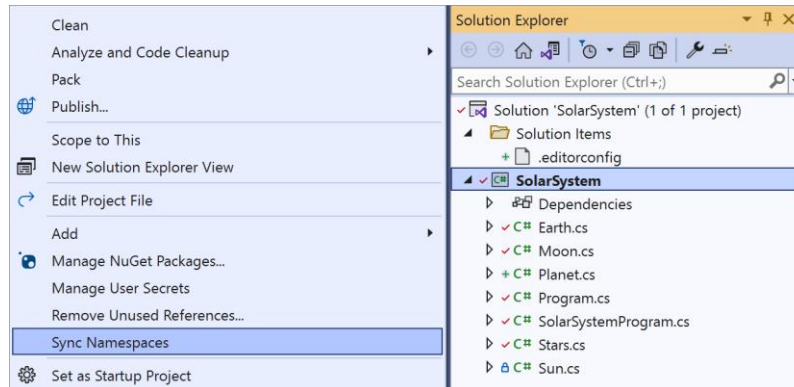
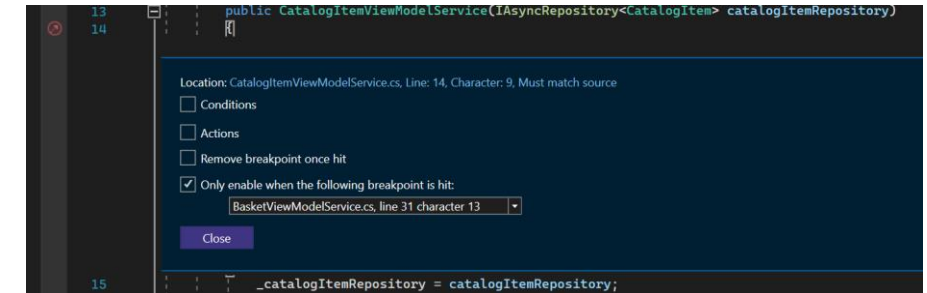
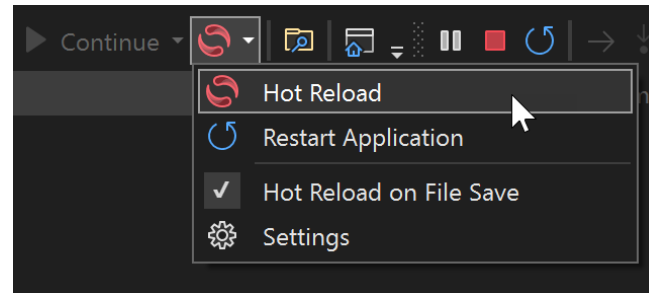
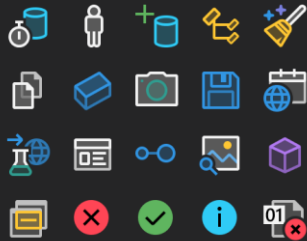


Visual Studio 2022

Light background



Dark background



Olvasnivalók, linkek, hivatkozások

Visual Studio 2022 Launch

<https://visualstudio.microsoft.com/launch/>

.NET Conf 2021 (dotnetconf.net)

<https://www.dotnetconf.net/>

.NET swag sweepstakes!: <https://www.dotnetconf.net/swag>

Free digital swag!: <https://github.com/dotnet-presentations/dotNETConf/tree/master/2021>

What's new in C# 10.0 - C# Guide

<https://docs.microsoft.com/en-us/dotnet/csharp/whats-new/csharp-10>

What's New in EF Core 6.0 | Microsoft Docs

<https://docs.microsoft.com/en-us/ef/core/what-is-new/ef-core-6.0/whatsnew>

.NET Blog

<https://devblogs.microsoft.com/dotnet/>

Preview 1: [.NET](#), [ASP.NET Core](#), [EF Core](#)

Preview 2: [.NET](#), [ASP.NET Core](#), [EF Core](#)

Preview 3: [.NET](#), [ASP.NET Core](#)

[ASP.NET Core 6 and Authentication Servers](#)

Preview 4: [.NET](#), [ASP.NET Core](#), [EF Core](#), [.NET MAUI](#), [Hot Reload](#)

Preview 5: [.NET](#), [ASP.NET Core](#), [EF Core](#), [.NET MAUI](#)

Preview 6: [.NET](#), [ASP.NET Core](#), [EF Core](#), [.NET MAUI](#)

Preview 7: [.NET](#), [ASP.NET Core](#), [.NET MAUI](#), [String Interpolation](#), [Generic Math](#)

[Introducing DevOps-friendly EF Core Migration Bundles](#)

[Performance Improvements in .NET 6](#)

[New .NET 6 APIs driven by the developer community](#)

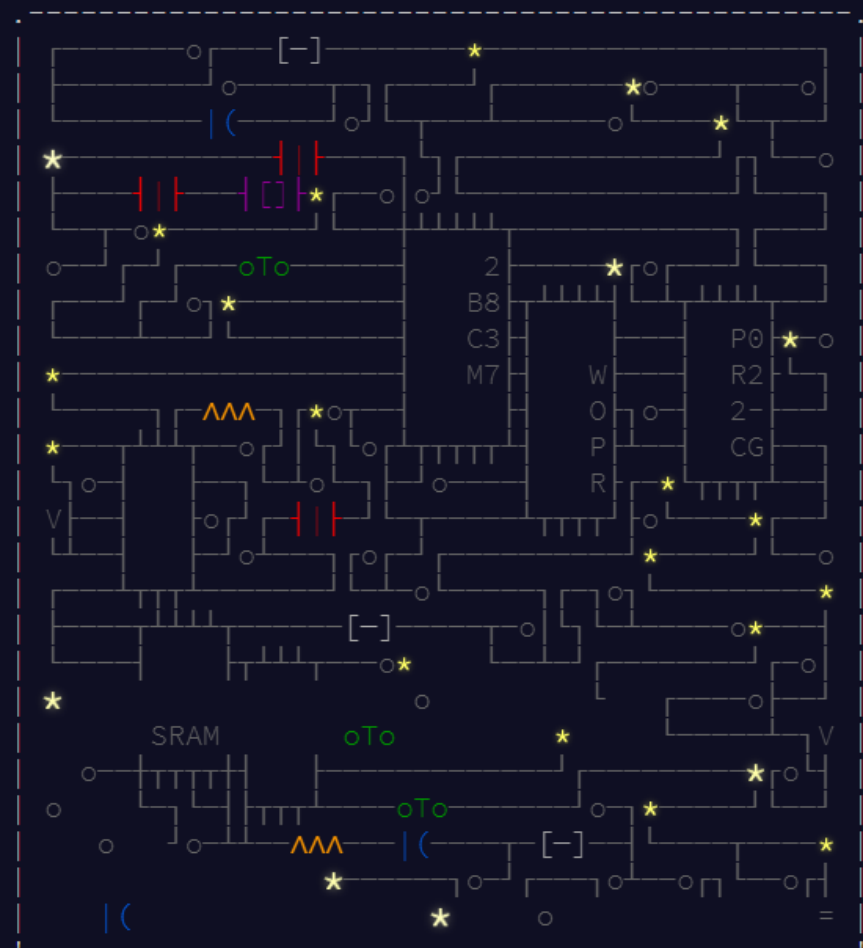
Release Candidate 1: [.NET](#), [ASP.NET Core](#), [.NET MAUI](#)

Release Candidate 2: [.NET](#), [ASP.NET Core](#), [.NET MAUI](#)

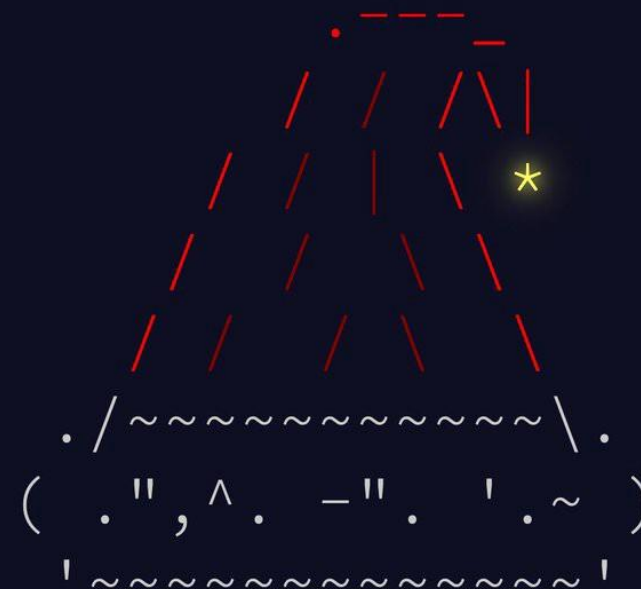
Visual Studio 2022 RC and Preview Release Notes

<https://docs.microsoft.com/en-us/visualstudio/releases/2022/release-notes-preview>

0	Naughty	Nice	0
0	-----		
0	The Easter Bunny	yugabe	0



25 **
24 **
23 **
22 **
21 **
20 **
19 **
18 **
17 **
16 **
15 **
14 **
13 **
12 **
11 **
10 **
9 **
8 **
7 **
6 **
5 **
4 **
3 **
2 **
1 **



Advent of Code

<https://adventofcode.com/>

My leaderboard join code: 262120-78a0e92f

<https://github.com/yugabe/Tidy.AdventOfCode>



Gábor Szabó
Assistant Lecturer
Microsoft Evangelist
Disruptive Software Architect
[linkedin.com/in/gabor-szabo/](https://www.linkedin.com/in/gabor-szabo/)

.NET 6, C# 10 és ezeken túl

Mit hoz a jövő?

2021. november 2.

.NET Conf 2021

November 9-11

6 days 23 hours 0 minutes

Save the Date! [f](#) [t](#) [e](#)

