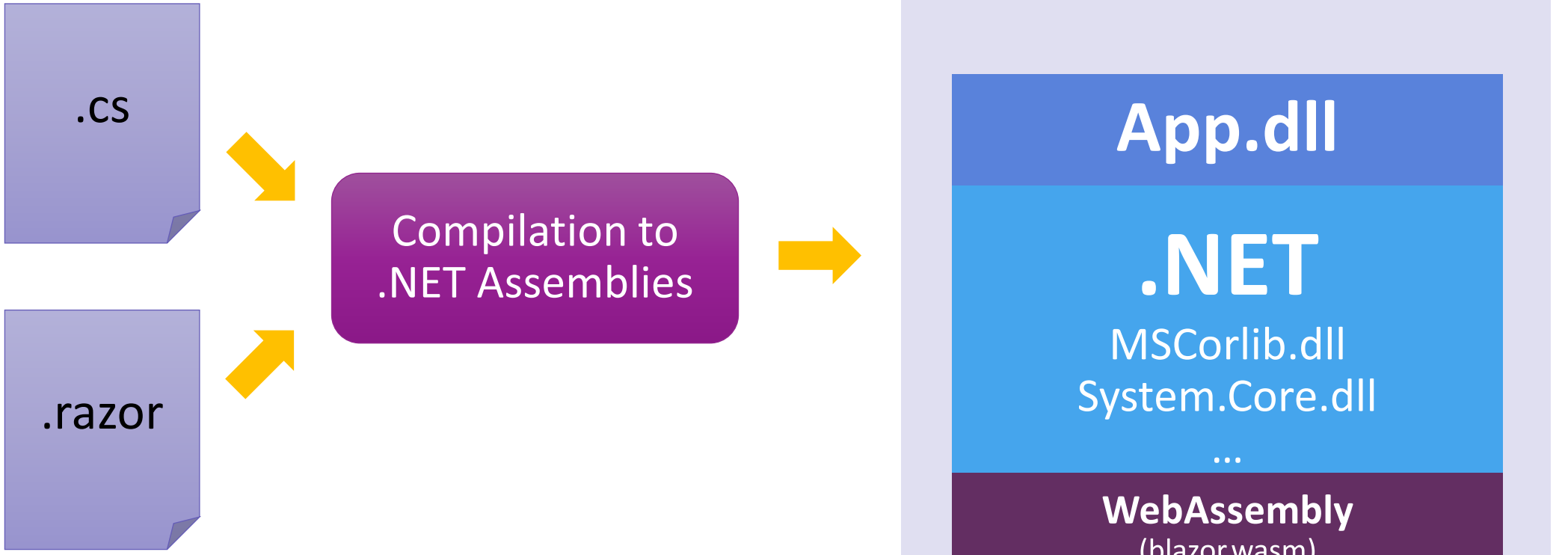
Blazor

La mort de JavaScript ?

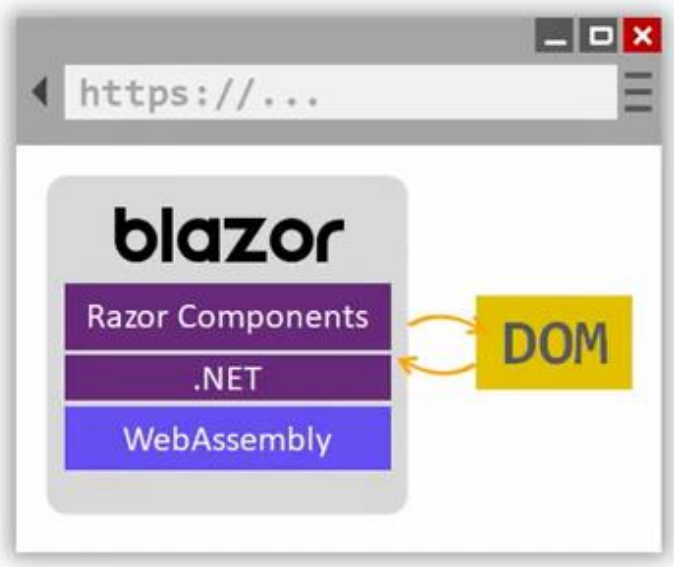
Michaël Fiorito - Denis **Voituron**


.cs
.razor
Compilation to
.NET Assemblies
BROWSER
App.dll
.NET
MSCorlib.dll
System.Core.dll
...
WebAssembly
(blazor.wasm)

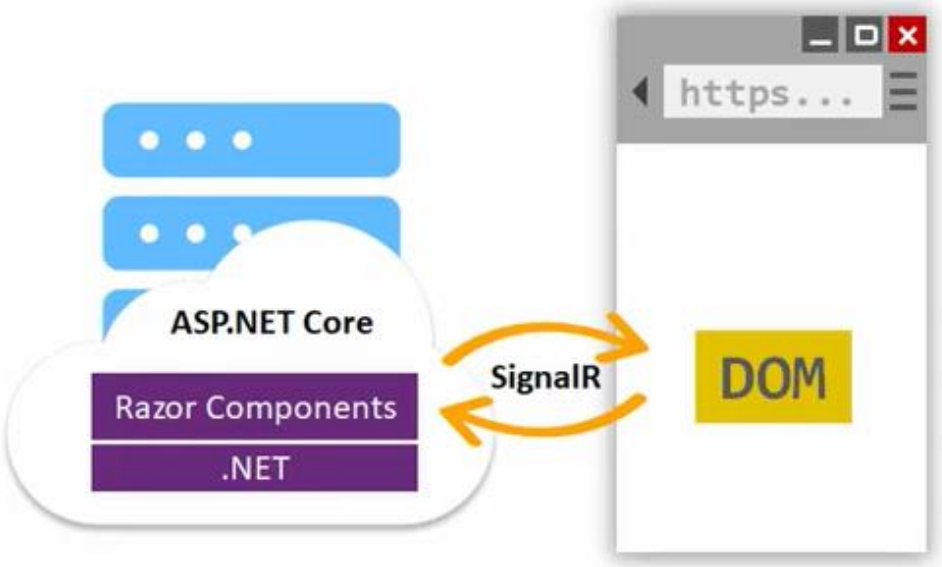
Planning


July 2019
.NET Core 3.0
RC
Sept 2019
.NET Core 3.0
GA
Nov 2019
.NET Core 3.1
LTS
Nov 2020
.NET 5.0
GA
Nov 2021
.NET 6.0
LTS
Nov 2022
.NET 7.0
GA
Nov 2023
.NET 8.0
LTS

Blazor… WebAssembly vs Serveur



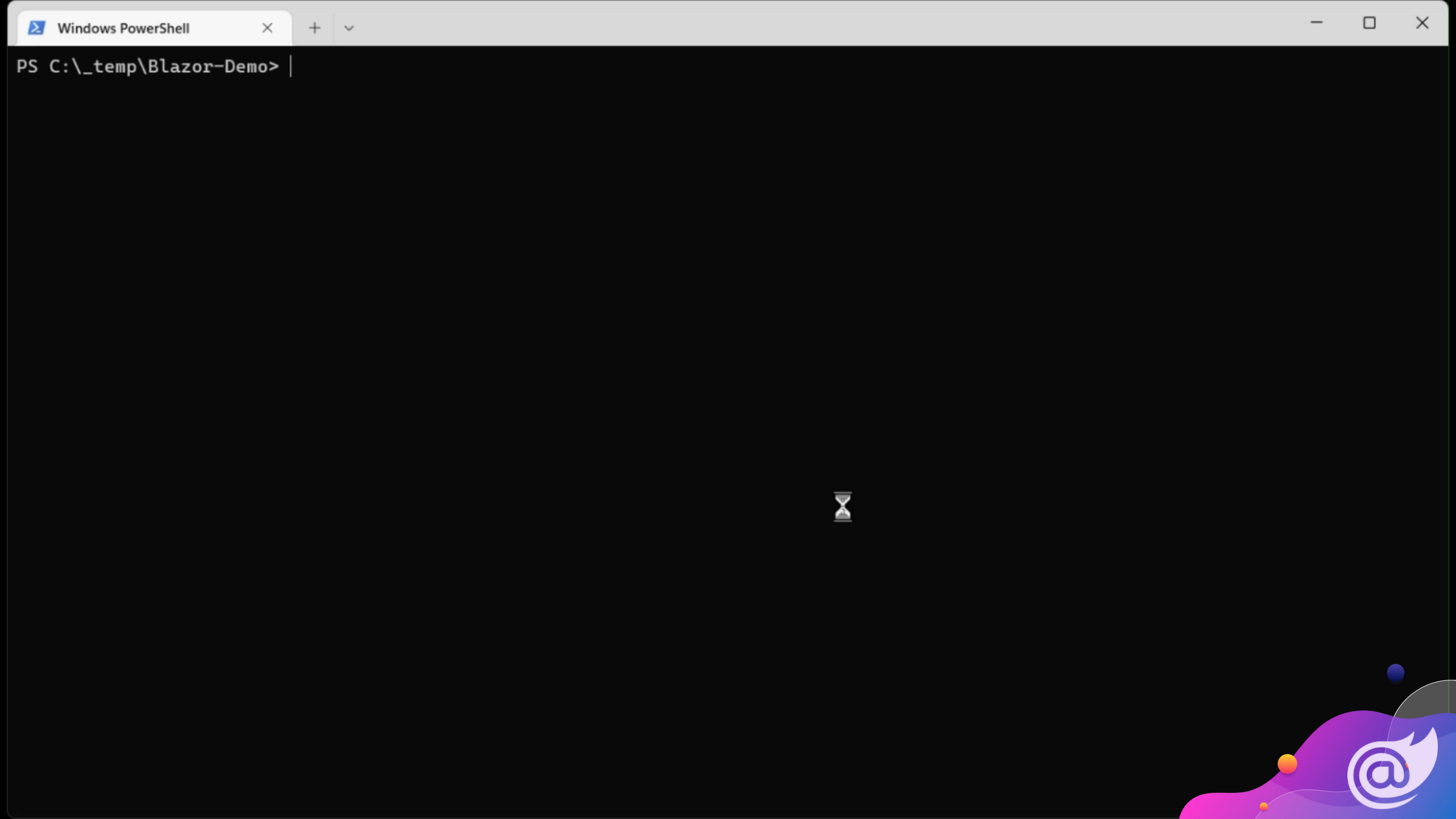
VS

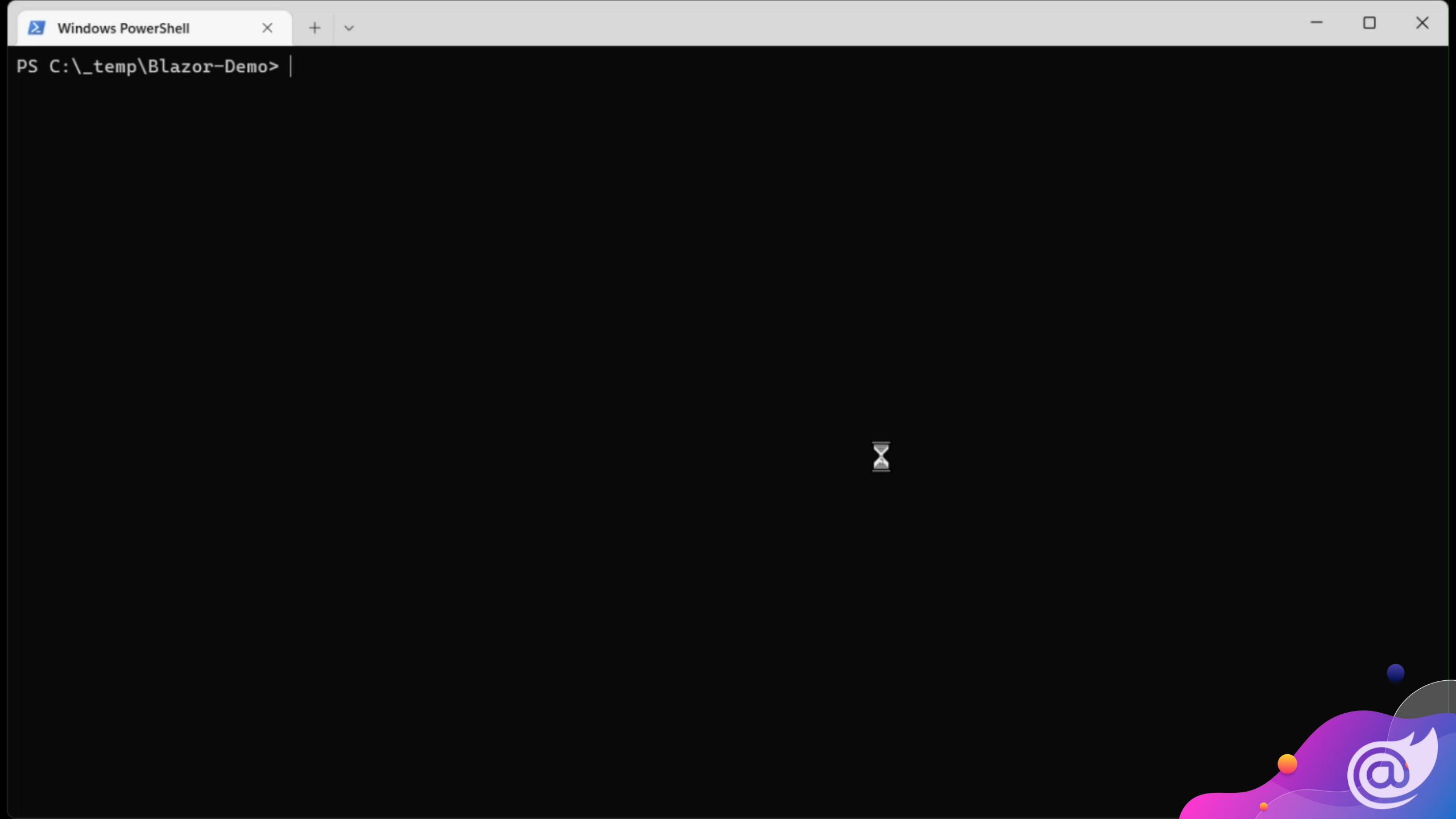


Single Page Applications

Applications hébergées
ASP.NET Core + SignalR

https://docs.microsoft.com/en-us/aspnet/core/host-and-deploy/blazor/server


Windows PowerShell
PS C:_temp\Blazor-Demo>


Windows PowerShell
PS C:_temp\Blazor-Demo>

Blazor Client ou Serveur ?

Blazor WebAssembly

- SPA, interactivité complète
- Utilisation des ressources clientes
- Support Offline, sites statiques…

- Taille à télécharger importante
- (Demande un navigateur compatible)

Blazor Server

- Peu de téléchargement, rapide à télécharger
- Architecture simplifiée
- Code ne quitte pas le serveur

- Latence
- Pas de support Offline
- Consommation de ressources sur le serveur

5,000 concurrent users: https://docs.microsoft.com/en-us/aspnet/core/blazor/host-and-deploy/server

Blazor

Sept 2019

Blazor Server

Web App Server
Chaque interaction est interceptée sur le serveur

Nov 2020

Blazor WebAssembly

Web App avec interactions sur le client
Chargée depuis un serveur web

Blazor PWA

Affichée comme une application native
Fonctionne offline et online

Blazor Hybrid

Rendus natifs via Electron ou WebView
Affichée comme une application native
Fonctionne offline et online

Blazor Native

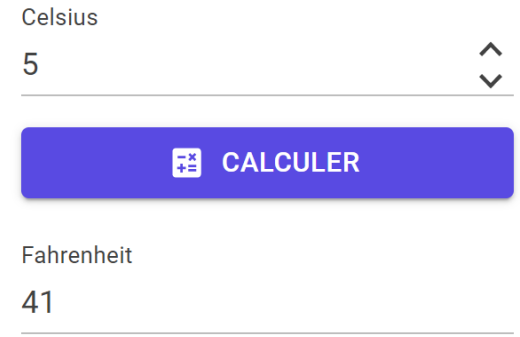
Même modèle de programmation,
Mais basé sur un rendu graphique non-HTML

$$F = \frac{9}{5}C + 32$$

Agenda

- Créer un composant
- Gérer une route
- Ajouter des [Parameter]
- Data binding
- Utiliser MudBlazor
- Ajouter un UnitTest

https://github.com/dvoituron/temperature-wasm



Composant

TemperaturePage.razor

```
@page "/temperature"
@page "/temperature/{Celsius:int}"

<MudStack>

    <MudNumericField T="int" Label="Celsius" @bind-
Value="@Celsius" />

    <MudButton Variant="MudBlazor.Variant.Filled"
               StartIcon="@Icons.Material.Filled.Calculate"
               Color="Color.Primary"
               OnClick="btnCalculate_Click">
        Calculer
    </MudButton>

    <MudTextField T="int" Label="Fahrenheit"
                  Value="@Fahrenheit"
                  ReadOnly="true" />

</MudStack>
```

TemperaturePage.razor.cs

```
using Microsoft.AspNetCore.Components;
namespace Temperature.Pages;

public partial class TemperaturePage
{
    [Parameter]
    public int Celsius { get; set; } = 0;

    public int Fahrenheit { get; set; } = 0;

    protected override void OnInitialized()
    {
        btnCalculate_Click();
    }

    public void btnCalculate_Click()
    {
        Fahrenheit = Convert.ToInt32((9d / 5d)
                    * Celsius + 32);
    }
}
```

Composant

TemperatureTest

```
[Fact]
public void Temperature_20Celsius_68Fahrenheit()
{
    // Arrange
    using var ctx = new TestContext();
    ctx.Services.AddMudServices();
    ctx.JSInterop.Mode = JSRuntimeMode.Loose;

    // Act
    var page = ctx.RenderComponent<TemperaturePage>(parameters =>
    {
        parameters.Add(p => p.Celsius, 20);
    });

    // Assert
    Assert.Contains("value=\"68\"", page.Markup);

    var fahrenheit = page.FindComponent<MudTextField<int>>();
    Assert.Equal(68, fahrenheit.Instance.Value);
}
```

dvoituron@microsoft.com
michael.fiorito@oniryx.be

https://github.com/dvoituron/temperature-wasm

Merci