

Razor

Tabela: Znaczniki Razor

Znacznik	Opis	Przykład użycia
@inject	Używane do wstrzykiwania zależności (np. serwisów) do komponentów lub stron.	@inject NavigationManager Navigation
@attribute	Używane do przypisania atrybutów do komponentu w Blazorze.	@attribute [Authorize] (przypisanie atrybutu, np. do autoryzacji)
@page	Określa ścieżkę URL dla strony w Blazorze.	@page "/home"
@bind	Używane do powiązania właściwości z elementami formularza, umożliwiając automatyczne aktualizowanie wartości.	@bind="inputValue"
@code	Umożliwia wstawienie bloku kodu C# w komponentach Razor.	@code { private string inputValue; }
@if	Używane do warunkowego renderowania zawartości w komponencie.	@if (isLoggedIn) { <h1>Witaj! </h1> }
@foreach	Używane do renderowania kolekcji lub tablicy w pętli w komponencie.	@foreach (var item in items) { <p>@item</p> }
@else	Używane w połączeniu z @if, aby określić, co ma się wyświetlić, jeśli warunek jest fałszywy.	@if (isLoggedIn) { <h1>Witaj! </h1> } @else { <h1>Zaloguj się! </h1> }
@while	Używane do renderowania zawartości w pętli while w komponencie.	@while (counter < 10) { <p>@counter</p> counter++; }
@foreach	Używane do iterowania po kolekcji i generowania HTML dla każdego elementu.	@foreach (var product in products) { <p>@product.Name</p> }
@ref	Umożliwia przypisanie referencji do elementu DOM lub komponentu, co pozwala na bezpośredni dostęp do niego w kodzie C#.	<input @ref="myInput" />

Znacznik	Opis	Przykład użycia
@inherits	Umożliwia dziedziczenie komponentów Blazor po innych klasach. Dzięki temu możemy rozszerzać funkcjonalność komponentu bazowego.	@inherits MyBaseComponent (komponent dziedziczy funkcjonalność klasy MyBaseComponent)
@layout	Określa layout, który ma być używany w komponencie, co pozwala na stworzenie wspólnego układu dla całej aplikacji.	@layout MainLayout (przypisanie layoutu do komponentu)

Wyjaśnienie:

- **@inject** : Umożliwia wstrzykiwanie serwisów lub innych zależności do komponentów Blazor. Na przykład, `NavigationManager` pozwala na nawigację po aplikacji.
- **@attribute** : Pozwala przypisać atrybuty do komponentu Blazor, takie jak `[Authorize]` do weryfikacji uprawnień.
- **@page** : Określa URL, pod którym dostępna będzie dana strona. Jest to konieczne w przypadku stron Blazor.
- **@bind** : Używane do powiązania właściwości w komponentach z elementami formularzy, np. z polem tekstowym. Zapewnia dwukierunkowe powiązanie danych.
- **@code** : Umożliwia wstawienie kodu C# w komponencie Razor, w sekcji, gdzie logika komponentu może być zapisana.
- **@if** i **@else** : Służą do renderowania zawartości na podstawie warunku. Może być używane do warunkowego wyświetlania danych lub komponentów.
- **@foreach** : Wykorzystuje pętlę do renderowania kolekcji elementów w komponencie. Dzięki temu można dynamicznie generować HTML na podstawie danych.
- **@while** : Używane w sytuacjach, gdy potrzebujemy pętli `while` do renderowania zawartości.
- **@ref** : Umożliwia przypisanie referencji do elementu DOM lub komponentu, umożliwiając późniejszy dostęp do niego z poziomu kodu C#.