

ASP Development Foundation

Prüfungscode IFW-ASP-100-01 · Fassung 1.0 · Stand 18.09.2026

Der Einstieg in ASP.NET Core: .NET-Plattform und Werkzeuge, C#-Grundlagen mit Typen, Klassen und Eigenschaften sowie das MVC-Muster.

Die Prüfung im Überblick

Bearbeitungszeit 40 Minuten	Fragen 30
Bestehensgrenze 70 %	Fragetypen Einfach- und Mehrfachauswahl
Prüfungssprache Deutsch	Zertifikat gültig Unbegrenzt
Durchführung Online, zeitlich überwacht	Preis Kostenlos

Themenblöcke

Block	Thema	Gewichtung	Fragen
1	Plattform, Projekt und Werkzeuge	30 %	9
2	C#: Typen, Ausgabe und Operatoren	20 %	6
3	C#: Klassen, Objekte und Eigenschaften	20 %	6
4	Das MVC-Muster	30 %	9
Gesamt		100 %	30

Die Gewichtung gibt an, welchen Anteil ein Block an den Prüfungsfragen hat. Die Fragen werden beim Start zufällig und anteilig je Block aus dem Fragenpool gezogen.

Plattform, Projekt und Werkzeuge

Was ASP.NET Core ist und wie ein Projekt mit der .NET CLI angelegt, gestartet und erweitert wird.

- 1.1 ASP.NET Core einordnen
plattformübergreifendes Open-Source-Framework für Webanwendungen, APIs und Dienste
- 1.2 Projekte mit der .NET CLI anlegen und starten
`dotnet new mvc`, `dotnet run`
- 1.3 Den Aufbau eines Projekts kennen
Einstiegspunkt `Program.cs`, Ordner `Controllers`, `Views`, `Models` und `wwwroot`
- 1.4 Pakete und Webserver einordnen
NuGet als Paketverwaltung, Kestrel als eingebauter Webserver

C#: Typen, Ausgabe und Operatoren

Die Grundbausteine der Sprache C#, in der ASP.NET-Core-Anwendungen geschrieben werden.

- 2.1 Variablen deklarieren
`int`, `double`, `bool`, `string`; statische Typisierung
- 2.2 Die Grundregeln der Syntax anwenden
Semikolon, Groß- und Kleinschreibung bei Bezeichnern
- 2.3 Ausgaben erzeugen
`Console.WriteLine()` und `Console.Write()`
- 2.4 Operatoren anwenden
== gegenüber `=`, `!=`, `&&`, `||`

C#: Klassen, Objekte und Eigenschaften

Objektorientierte Grundbegriffe, auf denen Controller und Models aufbauen.

- 3.1 Klassen und Objekte unterscheiden
Klasse als Bauplan, Objekte mit `new` erzeugen
- 3.2 Eigenschaften und Methoden definieren
Auto-Properties mit `get` und `set`
- 3.3 Zugriffsmodifizierer einsetzen
`public` und `private`

Das MVC-Muster

Wie Model, View und Controller eine Anfrage gemeinsam beantworten.

- 4.1 Die Aufgaben von Model, View und Controller unterscheiden
Daten, Darstellung und Steuerung
- 4.2 Controller und Actions schreiben
Namenskonvention ...Controller, return View();
- 4.3 HTTP-Methoden zuordnen
[HttpGet], [HttpPost]
- 4.4 Razor-Views anlegen
Dateiendung .cshtml, Ablage unter Views/<Controller>/

Institut für Webentwicklung · unabhängiges Lernangebot ohne Verbindung zu den genannten Technologieanbietern. Maßgeblich ist die Fassung auf der Zertifizierungsseite.