

Dart Programmeren

Doelgroep cursus Dart Programmeren

De cursus Dart Programmeren is bedoeld voor een ieder die wil leren programmeren in de programmeer taal Dart.

Voorkennis Cursus Dart Programmeren

Om aan deze cursus te kunnen deelnemen is basiskennis van programmeren in een andere programmeertaal bevorderlijk voor de begripsvorming maar niet vereist.

Uitvoering Training Dart Programmeren

De theorie wordt behandeld op basis van presentatie slides. De theorie wordt verduidelijkt door middel van demo's. Na bespreking van een module, is er de mogelijkheid te oefenen. De cursustijden zijn van 9.30 tot 16.30.

Certificering Cursus Dart Programmeren

De deelnemers krijgen na het goed doorlopen van de cursus een officieel certificaat Dart Programmeren.

Duur: 3 dagen

Prijs: € 1999

Open Rooster



Dart Programming



Inhoud Cursus Dart Programmeren

Dart is een general purpose programmeer taal die oorspronkelijk is ontwikkeld door Google. Dart is open source en inmiddels een ECMA standaard. Hierdoor wordt het ontstaan van een actief ecosysteem rond de taal bevorderd. De kracht van Dart ligt vooral bij de ontwikkeling van web applicaties en de verwachting is dat browsers Dart direct gaan ondersteunen. Dart staat ook aan de basis van het Flutter Framework voor mobiele applicaties.

Dart Intro

In de cursus Dart Programmeren leren de deelnemers de kenmerken van applicatie ontwikkeling met Dart. Ingegaan wordt op de Dart SDK en de transpiler dart2js die een JavaScript equivalent van Dart Script genereert.

Dart Syntax

Vervolgens wordt aandacht besteed aan data types, generics en control flow in Dart. Dart is type safe, ondersteunt type inference en voorkomt null pointer exceptions met een null safety mechanisme.

Functies en Data Structures

Ook komen de karakteristieken van functies en data structures in Dart aan de orde. Zo wordt ingegaan op data structures uit de core library, runes en het mirror system en komen bij functies optionele parameters en lambda functies aan bod.

Classes en Objects

Dart is een object georiënteerde taal en de implementatie van constructors, interfaces, exceptions en inheritance in Dart komt aan de orde. Hierbij worden ook extension methods besproken.

Concurrency

Tenslotte wordt uitgebreid ingegaan op concurrency in Dart waarbij parallelle taken door meerdere concurrent threads kunnen worden uitgevoerd. Hierbij wordt ook aandacht besteed aan asynchrone I/O met Dart Futures en het Async Package.

Modules Cursus Dart Programmeren

Module 1 : Dart Intro	Module 2 : Language Syntax	Module 3 : Data Structures
What is Dart? Install Dart SDK Dartpad Editor IDE Support Dart to JavaScript dart2js Program Execution Static checker Checked Mode Dart Keywords Dart Identifiers	Type Syntax Numbers Strings Booleans Dynamic Types Final and Const Operators Type Test Operators Iterations with Loops Selections with If Using Labels	Core Library Fixed Length List Growable List List Operations Map Literals Map Constructor Dart Symbols Runes Enumerations String.codeUnits MirrorSystem
Module 4 : Functions and Interfaces	Module 5 : Classes and Objects	Module 6 : Collections and Generics
Function Definition Calling Functions Passing Parameters Return Values Optional Parameters Recursive Functions Lambda Functions Interfaces Implementing Interface Multiple Interfaces	Declaring Classes Fields and Methods Getters and Setters Constructors Named Constructors this Keyword Class Inheritance Types of Inheritance Method Overriding static and super	Dart Collections Set and Queue Iterating Collections Optional Typing Type Safeness Parameterized Types Generic Map Generic List Isolates TypeDefs
Module 7 : Packages	Module 8 : Exceptions	Module 9 : Concurrency
Packaging Programming Units Package Manager pub Package Metadata pubsec.yaml Installing Packages pub get Command Importing Libraries Encapsulation Custom Libraries	Exception Handling try Block on/catch block finally Block ON Block Exception Class Built-in Exceptions Throwing Exceptions Custom Exceptions	Parallel Tasks Multiple Threads Isolates versus Threads Isolate Class spawn Method Dart Futures Async Package Asynchronous I/O readLineSync Method