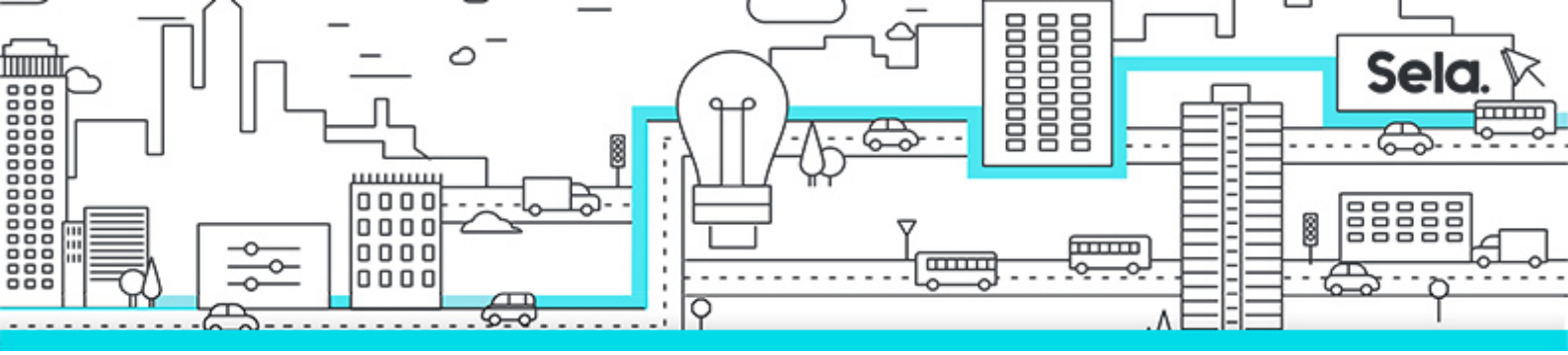
Sela.

FSharp

F# Programming

college@sela.co.il 03-6176666



F# Programming

FSharp - Version: 1

4 days course

Description:

This 4 days course provides students with knowledge and skills to develop applications in .NET using F# and functional programming language. F# has a great community and open source libraries can make you more productive and write application with much less bugs. The course features an overview of language related features as well as using Type Providers, stream processing, Asynchronous programming and Reactive programming.

Intended Audience:

The course is intended for developers with good knowledge of .NET ecosystem and practical experience of at least 3 years.

Prerequisites:

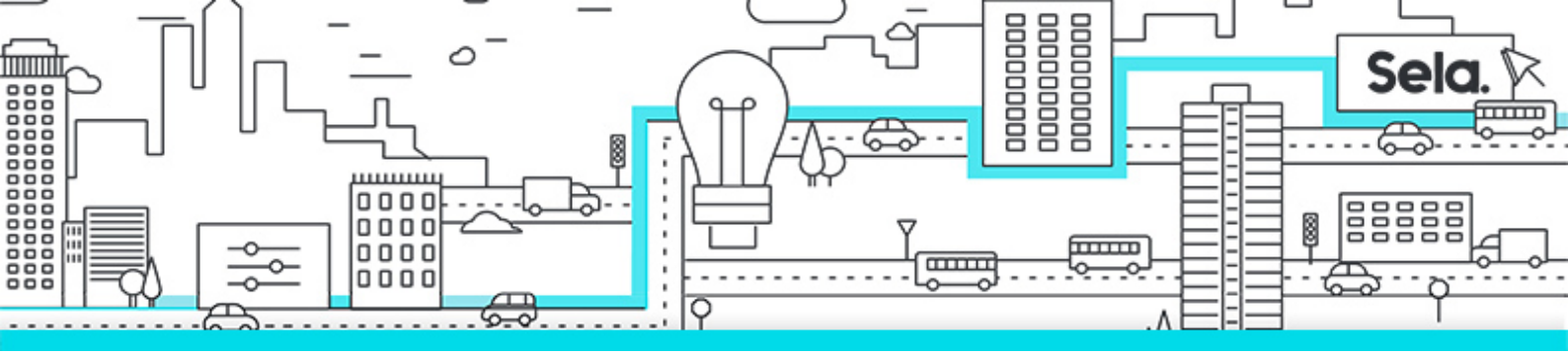
• C# Developers with at least 3 years of experience

Objectives:

• Develop applications with F# in the .NET ecosystem
• Apply functional programming principles to .NET applications
• Write less code and less bugs with functional programming and property based testing
• Use asynchronous and reactive programming to utilize resources efficiently
• Use metaprogramming to explore and integrate data into your applications

Topics:

• **Module 01 - Introduction to FP and F#**



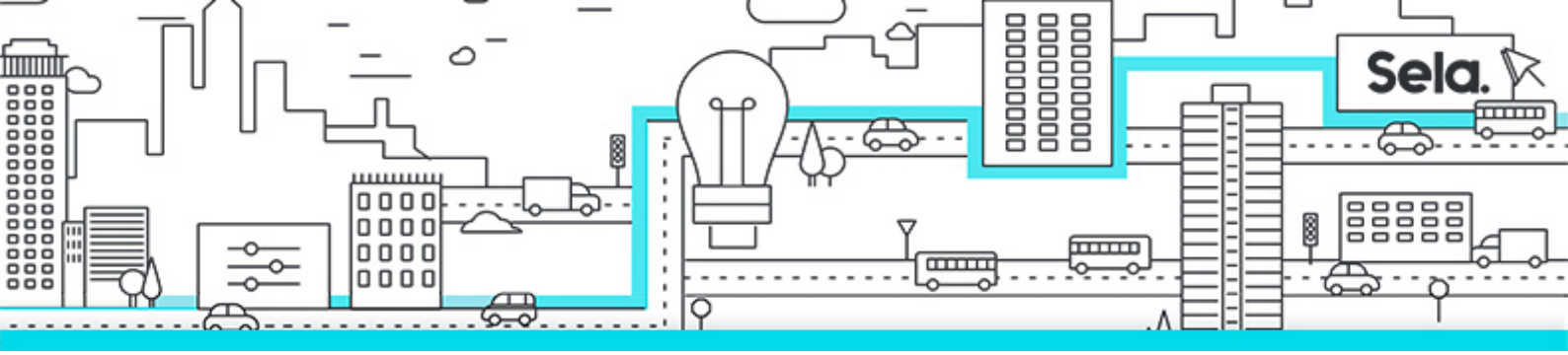
- What is functional Programming
- What is F#
- Functional programming and F# motivations
- Functional programming foundations

• **Module 02 - Data Modeling**
- Data Modeling motivations
- Tuples
- Records
- List & Seq
- Discriminated Union
- Pattern matching
- Active pattern
- Unit of measures

• **Module 03 - Type Providers & Data Visualization**
- Type Providers
- Data Visualization

• **Module 04 - Async Programming**
- Async Programming motivation
- Async Workflow
- Cancellation
- C# vs F# Similarity
- C# vs F# Difference
- Async and parallel programming
- Async and UI

• **Module 05 - Appendix I**
- F# vs C#
- Using C# code from F# code



- Using F# code form C# code
- F# project structure
- Nuget vs Paket
- MSBuild vs FAKE

• **Module 06 - Property Based Testing**
- F# and Unit Test
- Property Based Testing

• **Module 07 - Appendix II - Advanced Functional Design Pattern**
- OOP Design Patterns vs Functional programming
- Fluent interface vs Functional programming
- Functional Design Patterns – Monoid
- Functional Design Patterns – Option
- Functional Design Patterns – Monad
- Computation Expression

• **Module 08 - Agents**
- Agents motivations
- What is Agents
- Agents vs Actors
- Agents patterns

• **Module 09 - Stream Processing**
- Stream processing motivation
- Stream processing basics
- Real world examples
- F# other streams libraries