Intermediate TypeScript – Beyond the basics
Duration: 3 days

framework training

In a nutshell

This course takes experienced JavaScript wranglers into the ever-evolving world of ECMAScript both in and out of the browser

Prerequisites

A sound practical working knowledge of JavaScript is essential to attend this course. If you already have significant Typescript experience, elements of this course may be covering things you already know.

Day 1

Quick Review of TypeScript

Why Typescript is such a stonkingly good idea
Remember – Typescript is design-time only
Using tsc and ts-node (and package.json)

How to get there from here

Adding TypeScript to existing JS files
Use as little or as much TypeScript as you like
Towards an agreed code style: using TypeScript consistently
Get out of jail free with <any> type and its relatives (and when not to use them)

Advanced TypeScript

Checking against permitted members with Type Guards
Partial, Required and other Utility types
Assigning Type based on conditions
Deriving Types from collection index members
New types for old with map types

Close of Day Discussion and Q&A

Day 2

Leading-Edge ECMAScript

Keeping up with the code-ashians: ECMAScript just keeps getting better every year
What's new since ES6:
…and you *are* making use of back ticks and interpolation, aren't you?

Functional Programming

Why pure functions turn out to be a good idea
Scalable encapsulation, testability and stateful apps
What that means for coding style
The ReactJS template for TypeScript

Generics and Observables

When we need generic types
Are we there yet? What the RxJS library is teaching ECMAScript
Observables make event streams easy to handle
Consuming end-point APIs

Close of Day Discussion and Q&A

Day 3

Class, Interface, Type, Enum

When to use the various type definitions
The Typescript language engine improves code
These types are *really* cool … and flexible!

Modules and Namespaces

Import, Export (and not a trade deal in sight)
Dependency injection, inversion of dependency and all that jazz
These aren't just and old components, they're *namespaced* components

Day 3 (cont.)

Project Structuring

Towards architectural conventions
Automated tree-shaking, dehydration and module refactoring (oh my!)
When you say compile, you *really* mean transpile

ES2020 and beyond

Functions – let me count the ways
Destructuring and manipulating structures
What's coming

NB this suggested content is intended as a stating point for discussion rather than a prescriptive list.