TypeScript Advanced

Generics, mapped types, template literals and branded types – TypeScript that catches bugs before the code runs.

1 OR 2 DAYS | ADVANCED | REMOTE OR IN-HOUSE | GERMAN & ENGLISH | UP TO 12 PARTICIPANTS (MORE ON REQUEST)

WHO IT'S FOR

- Developers who use TypeScript daily and want to go beyond the basics
- Teams maintaining shared libraries or complex domain models

AFTERWARDS YOUR TEAM CAN

- Model domain rules in the type system instead of in runtime checks
- Write generic, reusable utilities with precise types
- Read and extend advanced types in third-party libraries

PREREQUISITES

- Practical TypeScript experience
- Comfortable with interfaces, unions and generics basics

Level up your TypeScript skills with this TypeScript Advanced Course. Get a deeper understanding of how TypeScript works and how you can use advanced concepts and types to enhance your software projects. Writing more robust and expressive code while applying best practices will improve your overall developer experience, make you and your team more efficient, and reduce maintenance effort.

AGENDA

The order and the depth of each block are adapted to your team – topics you already master are skipped, your questions get the time instead.

01 **Basics**
A small summary of the basic abilities and limitations of TypeScript.

02 **Type System**
Learn about type systems in different languages and how TypeScript's type system works.

03 **Advanced Types**
Create more flexible as well as expressive types with Union and Intersection Types.

04 **Utility Types**
Leverage TypeScript's Utility Types, which cover some of the most common type transformations.

05 **Never**
Gain knowledge about the concept of the Never Type and how you can use this type to tell TypeScript about situations that should never happen.

06 **Generics**
Master generics with a deep dive into generic constraints, inferring types from other types, as well as type recursion.

07 **Literals & Template Literals**
Learn how to use more specific types than just string and number. Get a better understanding of how to combine these types and specifically tell TypeScript to narrow down objects and arrays to their most specific type on its own.

08 **Mapped Types**
Create types based on other types to avoid redundancy and enforce changes across connected types.

09 **Type Guards**
Explore the possibilities of how to securely recognize and narrow down specific types.

10 **Modern TypeScript Features**
The satisfies operator, const type parameters, standard decorators and using declarations – recent additions that make types more precise and code safer.

11 **Branded Types**
Explore a more advanced concept on how to create types for more complex patterns like emails or social security numbers.

YOUR TRAINER



Tim Honermann
SENIOR SOFTWARE ENGINEER
Senior Software Engineer for Angular and Java, builds full-stack applications for clients and writes about it on his blog.

REQUEST A TRAINING

- For your team only – Remote or in-house
- up to 12 participants (more on request)
- from CHF 2,400 per day for the whole team
- On site: plus travel at cost
- Fixed quote within one business day
- The first call is free and without obligation
- German or English
- Based near Zurich – on site in CH, DE, AT; remote everywhere

Request this training: **offering.solutions/training-typescript-advanced/** · mail@offering.solutions