

TypeScript Grundlagen

Seminarunterlage

Version: 1.04



Dieses Dokument wird durch die ORDIX AG veröffentlicht.

Copyright ORDIX AG. Alle Rechte vorbehalten.

Alle Produkt- und Dienstleistungs-Bezeichnungen sind Warenzeichen oder eingetragene Warenzeichen der jeweiligen Firmen und beziehen sich auf Eintragungen in den USA oder USA-Warenzeichen.

Weitere Logos und Produkt- oder Handelsnamen sind eingetragene Warenzeichen oder Warenzeichen der jeweiligen Unternehmen.

Kein Teil dieser Dokumentation darf ohne vorherige schriftliche Genehmigung der ORDIX AG weitergegeben oder benutzt werden.

Adressen der ORDIX AG

Die ORDIX AG besitzt folgende Geschäftsstellen

ORDIX AG
Karl-Schurz-Straße 19a
D-33100 Paderborn
Tel.: (+49) 0 52 51 / 10 63 - 0
Fax.: (+49) 01 80 / 1 67 34 90

ORDIX AG
An der alten Ziegelei 5
D-48157 Münster
Tel.: (+49) 02 51 / 9 24 35 – 00
Fax.: (+49) 01 80 / 1 67 34 90

ORDIX AG
Welser Straße 9
D-86368 Gersthofen
Tel.: (+49) 08 21 / 507 492 – 0
Fax.: (+49) 01 80 / 1 67 34 90

ORDIX AG
Kreuzberger Ring 13
D-65205 Wiesbaden
Tel.: (+49) 06 11 / 7 78 40 – 00
Fax.: (+49) 01 80 / 1 67 34 90

ORDIX AG
Wikingerstraße 18-20
D-51107 Köln
Tel.: (+49) 02 21 / 8 70 61 – 0
Fax.: (+49) 01 80 / 1 67 34 90

Internet: <http://www.ordix.de>

Email: seminare@ordix.de

Inhaltsverzeichnis

- 1 Überblick 5**
 - 1.1 Was ist TypeScript 6
 - 1.2 Daten und Fakten..... 7
 - 1.3 Geschichte von TypeScript 8
 - 1.4 TypeScript: Führend bei den CompileToJS-Sprachen 9
 - 1.5 TypeScript & Angular: Eine Erfolgsgeschichte 10
 - 1.6 TypeScript Verwendung..... 11
 - 1.7 Warum brauchen wir TypeScript?..... 12
 - 1.8 Limitationen von JavaScript 13
 - 1.9 Die TS Antwort: ein statisches Typensystem..... 14
 - 1.10 Maximale Kompatibilität: Der Transpileransatz..... 15
 - 1.11 Keine ungesunde Bindung: Opt-In..... 16
 - 1.12 TypeScript im Verhältnis zu JavaScript 17
- 2 Installation & Setup..... 18**
 - 2.1 Das Herzstück: Der TypeScript Compiler 19
 - 2.2 Erster Gebrauch des Compilers..... 20
 - 2.3 Editoren & IDEs..... 21
 - 2.4 Dokumentation 22
- 3 Projekt Setup 23**
 - 3.1 Compileroptionen 24
 - 3.2 Die Projektkonfiguration: tsconfig.json 25
 - 3.3 Quell-Dateien und Ordner konfigurieren 26
 - 3.4 Ziel Ordner konfigurieren 27
 - 3.5 Zielversion und Modulsystem 28
 - 3.6 Buildtasks in Visual Studio Code 29
- 4 Typen..... 30**
 - 4.1 Strukturelles Typensystem..... 31
 - 4.2 Typensicherheit..... 32
 - 4.3 Basistypen..... 33
 - 4.4 Erweiterte Basistypen 34
 - 4.5 Beispiele: Deklaration Basistypen..... 35
 - 4.6 Any 36
 - 4.7 Enums 37
 - 4.8 Arrays..... 38
 - 4.9 Tuple 39
 - 4.10 Void und Never 40
 - 4.11 Union Types 41
 - 4.12 Komplexe Typen 42
 - 4.13 Typ Kompabilität 43
 - 4.14 Type Inference 45
 - 4.15 Type Inference: Best Common Type 46
 - 4.16 Kontextuelle Typisierung..... 47
 - 4.17 Type Assertion 48
- 5 Grundlegende Sprachfeatures..... 49**
 - 5.1 Variablen Deklaration..... 50
 - 5.2 Das Problem mit var..... 51
 - 5.3 const bei Objekten 52
 - 5.4 Kontrollstrukturen 53
 - 5.4.1 if-else 54
 - 5.4.2 Switch 55
 - 5.4.3 for Schleifen 56
 - 5.4.4 while Schleifen 57
 - 5.5 Fehlerbehandlung 58

6	Funktionen	59
6.1	Benannte Funktionen und Anonyme Funktionen.....	60
6.2	Funktionsstypen.....	61
6.3	optionale und default Parameter.....	62
6.4	Rest Parameter.....	63
6.5	Funktions Overloading.....	64
6.6	Arrow Funktionen (Lambdas).....	65
7	Klassen und Interfaces	66
7.1	Interfaces.....	67
7.2	Implizite Interfaces.....	68
7.3	Benannte Interfaces.....	69
7.4	optionale Properties eines Interfaces.....	70
7.5	Readonly Properties.....	71
7.6	Funktions Interfaces.....	72
7.7	Indexable Typen.....	73
7.8	Klassen.....	74
7.9	Vererbung.....	75
7.10	Access Modifier.....	76
	7.10.1 private Properties.....	77
	7.10.2 protected Properties.....	78
7.11	Konstruktor: Parameter Properties.....	79
7.12	Accessoren: Getter und Setter.....	80
7.13	Statische Properties.....	81
7.14	Abstrakte Klassen.....	82
7.15	Klassen als Interface verwenden.....	83
8	Debugging	84
8.1	Live Demo Debugging.....	85
9	Module und Namespaces	86
9.1	Module.....	87
9.2	Moduleloader.....	88
9.3	Namespaces.....	89
9.4	Verwendung von Modulen.....	90
9.5	Beispiele: export einer Deklaration.....	91
9.6	Beispiele: import von Deklarationen.....	92
9.7	Weitere Beispiele.....	93
9.8	Barrels.....	94
9.9	Namespaces vs. Module.....	95
10	Externe Bibliotheken	96
10.1	Externe Libraries in TypeScript.....	97
10.2	Eigene Deklarationen.....	98
11	Fortgeschrittene Typen	99
11.1	Iterables.....	100
11.2	Union Types.....	101
11.3	Type Guards.....	102
11.4	instanceof TypeGuards.....	103
11.5	Generics.....	104
12	Asynchrone Funktionen	106
12.1	async – await.....	107
13	Decorators	109
13.1	Decorators.....	110
13.2	Decorators in TypeScript.....	111
13.3	Decorator implementieren.....	112