Vskills CERTIFIED

www.vskills.in

Certified Typescript Professional

Certified Typescript Professional

TypeScript is an open source programming language maintained by Microsoft. TypeScript extends JavaScript by adding type checking and host of other features as interfaces, type aliases, abstract classes, function overloading, tuple, generics, etc. TypeScript has gained widespread acceptance with the rise of e-commerce and mobile applications as it offers many developer friendly features and integration with JavaScript and popular frameworks like Angular, etc.

Why should one take Vskills Typescript Professional certification?

Vskills Typescript Professional certification provides a thorough knowledge on Typescript language covering from basics to advanced level by providing videos for a hand-on approach.

The certification covers

- Classes
- Interfaces
- Generics

Who will benefit from taking Vskills Typescript Professional certification?

Vskills Typescript Professional certification has been extremely beneficial for IT professionals and students for their career growth.

The certification is apt for

- Software Developers
- Programmers
- IT managers, team leads
- Students interested in web development

Test Details

- **Duration:** 60 minutes
- **No. of questions:** 50
- **Maximum marks:** 50, Passing marks: 25 (50%)

There is no negative marking in this module.

Fee Structure

Rs. 3,499/- (Excludes taxes)*

*Fees may change without prior notice, please refer http://www.vskills.in for updated fees

Companies that hire Vskills Certified Typescript Professionals

Typescript professionals are in demand at software companies, outsourcing companies like TCS, Accenture, IBM, Capgemini, Tech Mahindra, TCS, etc.

Table of Contents

4. Next-generation JavaScript & TypeScript

4.1 'let' and 'const'
4.2 Arrow Functions
4.3 Default Function Parameters
4.4 The Spread Operator (...)
4.5 Rest Parameters
4.6 Array & Object Destructuring
4.7 How Code Gets Compiled & Wrap Up

5. TypeScript Classes & Interfaces

5.1 What are Classes?
5.2 Creating a First Class
5.3 Compiling to JavaScript
5.4 Constructor Functions & The 'this' Keyword
5.5 'private' and 'public' Access Modifiers
5.6 Shorthand Initialization
5.7 'readonly' Properties
5.8 Inheritance
5.9 Overriding Properties & The 'protected' Modifier
5.10 Getters & Setters
5.11 Static Methods & Properties
5.12 Abstract Classes
5.13 Singletons & Private Constructors
5.14 Classes - A Summary
5.15 A First Interface
5.16 Using Interfaces with Classes
5.17 Why Interfaces?
5.18 Readonly Interface Properties
5.19 Extending Interfaces
5.20 Interfaces as Function Types
5.21 Optional Parameters & Properties
5.22 Compiling Interfaces to JavaScript

6. TypeScript Advanced Types

6.1 Intersection Types
6.2 More on Type Guards
6.3 Discriminated Unions
6.4 Type Casting
6.5 Index Properties
6.6 Function Overloads
6.7 Optional Chaining
6.8 Nullish Coalescing

7. TypeScript Generics

7.1 Built-in Generics & What are Generics?
7.2 Creating a Generic Function
7.3 Working with Constraints
7.4 Another Generic Function
7.5 The 'keyof' Constraint
7.6 Generic Classes
7.7 A First Summary
7.8 Generic Utility Types
7.9 Generic Types vs Union Types

8. TypeScript Decorators

8.1 A First-Class Decorator
8.2 Working with Decorator Factories
8.3 Building More Useful Decorators
8.4 Adding Multiple Decorators
8.5 Diving into Property Decorators
8.6 Accessor & Parameter Decorators
8.7 When Do Decorators Execute?
8.8 Returning (and changing) a Class in a Class Decorator
8.9 Other Decorator Return Types
8.10 Example: Creating an 'Autobind' Decorator
8.11 Validation with Decorators - First Steps
8.12 Validation with Decorators - Finished

9. TypeScript Practice Project - Drag & Drop Project

9.1 Getting Started
9.2 DOM Element Selection & OOP Rendering
9.3 Interacting with DOM Elements
9.4 Creating & Using an 'Autobind' Decorator
9.5 Fetching User Input
9.6 Creating a Re-Usable Validation Functionality
9.7 Rendering Project Lists
9.8 Managing Application State with Singletons
9.9 More Classes & Custom Types
9.10 Filtering Projects with Enums
9.11 Adding Inheritance & Generics
9.12 Rendering Project Items with a Class
9.13 Using a Getter
9.14 Utilizing Interfaces to Implement Drag & Drop
9.15 Drag Events & Reflecting the Current State in the UI
9.16 Adding a Droppable Area
9.17 Finishing Drag & Drop

10. TypeScript Modules & Namespaces

10.1 Writing Module Code - Your Options
10.2 Working with Namespaces
10.3 Organizing Files & Folders
10.4 A Problem with Namespace Imports
10.5 Using ES Modules
10.6 Understanding various Import & Export Syntaxes
10.7 How Does Code In Modules Execute?

11. Using Webpack with TypeScript

11.1 What is Webpack & Why do we need it?
11.2 Installing Webpack & Important Dependencies
11.3 Adding Entry & Output Configuration
11.4 Adding TypeScript Support with the ts-loader Package
11.5 Finishing the Setup & Adding webpack-dev-server
11.6 Adding a Production Workflow

12. 3rd Party Libraries & TypeScript

12.1 Using JavaScript (!) Libraries with TypeScript
12.2 Using 'declare' as a 'Last Resort'
12.3 No Types Needed: class-transformer
12.4 TypeScript-embracing: class-validator

13. TypeScript Practice Project - "Select & Share a Place" App (incl. Google Maps)

13.1 Project Setup
13.2 Getting User Input
13.3 Setting Up a Google API Key
13.4 Using Axios to Fetch Coordinates for an Entered Address
13.5 Rendering a Map with Google Maps (incl. Types!)

14. React.js & TypeScript

14.1 Setting Up a React + TypeScript Project
14.2 How Do React + TypeScript Work Together?
14.3 Working with Props and Types for Props
14.4 Getting User Input with 'refs'
14.5 Cross-Component Communication
14.6 Working with State & Types
14.7 Managing State Better
14.8 More Props & State Work
14.9 Adding Styling
14.10 Types for other React Features (e.g. Redux or Routing)

15. Node.js + Express & TypeScript

Sample Questions

1. What is the inherited type for the variable example in 'const example = ['Dylan']'?

A. any[]
B. string[]
C. string
D. unknown[]

2. What does the 'readonly' access modifier do for an array variable assignment like: 'const codeNames: readonly string[] = ['Coding', 'God']'?

A. Makes you read it for better clean code
B. Allows only adding but not deleting elements in the array
C. Makes it private and can only be used in the file its created
D. Allows no changes and is there simply to be read from and not modified

3. Which is a successful example of this tuple '[number, string]'?

A. const ourTuple = [101]
B. const ourTuple = ['Coding God', 101]
C. const ourTuple = [101, 101, 'Coding God', 'Coding God']
D. const ourTuple = [101, 'Coding God']

4. Type Aliases are mostly used with _____.

A. Strings
B. Booleans
C. Numbers
D. None of the above

5. _____ an interface will have the same properties as that interface.

A. Idolizing
B. Improving
C. Extending
D. Duplicating

Answers: **1 (B), 2 (D), 3 (D), 4 (A), 5 (C)**

Certifications

Accounting, Banking & Finance
- Certified GST Professional
- Certified AML-KYC Compliance Officer
- Certified Business Accountant
- Certified BASEL III Professional
- Certified GAAP Accounting Standards Professional
- Certified Treasury Markets Professional

Big Data
- Certified Hadoop and Mapreduce Professional

Cloud Computing
- Certified Cloud Computing Professional

Design
- Certified Interior Designer

Digital Media
- Certified Social Media Marketing Professional
- Certified Inbound Marketing Professional
- Certified Digital Marketing Professional

Foreign Trade
- Certified Export Import (Foreign Trade) Professional

Health, Nutrition and Well Being
- Certified Fitness Instructor

Hospitality
- Certified Restaurant Team Member (Hospitality)

Human Resources
- Certified HR Compensation Manager
- Certified HR Staffing Manager
- Certified Human Resources Manager
- Certified Performance Appraisal Manager

Office Skills
- Certified Data Entry Operator
- Certified Office Administrator

Project Management
- Certified Master in Project Management
- Certified Scrum Specialist

Real Estate
- Certified Real Estate Consultant

Marketing
- Certified Marketing Manager

Quality
- Certified Six Sigma Green Belt Professional
- Certified Six Sigma Black Belt Professional
- Certified TQM Professional

Logistics & Supply Chain Management
- Certified International Logistics Professional
- Certified Logistics & SCM Professional
- Certified Supply Chain Management Professional

Legal
- Certified IPR & Legal Manager
- Certified Labour Law Analyst
- Certified Business Law Analyst
- Certified Corporate Law Analyst

Information Technology
- Certified Angular JS Professional
- Certified Basic Network Support Professional
- Certified Business Intelligence Professional
- Certified Core Java Developer
- Certified E-commerce Professional
- Certified IT Support Professional
- Certified PHP Professional
- Certified Selenium Professional

Mobile Application Development
- Certified Android Apps Developer
- Certified iPhone Apps Developer

Security
- Certified Ethical Hacking and Security Professional
- Certified Network Security Professional

Management
- Certified Corporate Governance Professional
- Certified Corporate Social Responsibility Professional
- Certified Leadership Skills Professional

Life Skills
- Certified Business Communication Specialist
- Certified Public Relations Officer

Media
- Certified Advertising Manager
- Certified Advertising Sales Professional

Sales, BPO
- Certified Sales Manager
- Certified Telesales Executive

& many more job related certifications

Contact us at:
V-Skills
011-473 44 723 or info@vskills.in
www.vskills.in