Typescript test 2

Answers

1) Install TypeScript globally (npm command)
npm install -g typescript

2) Type alias for “string or number”:
type StringOrNumber = string | number;

3)Interface vs type (declaration merging):
Interfaces can declaration-merge (same name can merge). Types cannot.

4)noImplicitAny: true in tsconfig.json does what?
It throws an error when TypeScript would otherwise infer an implicit any type.

5)Mark an object property as read-only:
Use readonly
readonly id: number;

6)Purpose of never type + where used:
never means it never returns / can’t happen. Used for functions that throw errors or run forever.
Example: function fail(): never { throw new Error("x"); }

7)How optional chaining ?. works + why safer:
It stops and returns undefined if the left side is null/undefined, instead of crashing.
Example: user?.profile?.name

8)Why unknown is safer than any:
With unknown, you must check/narrow the type before using it. With any, TypeScript skips type checking.

9)What are Generics + why useful:
Generics let you write reusable, type-safe code by using a type variable like T.
Example: function wrap<T>(value: T): T { return value; }

10)Use keyof to get property names as a union:
type Keys = keyof MyInterface;