

TypeScript Types

Computing

The type vocabulary, from primitives to the utility types worth knowing.

The types you will actually write, grouped by what they do.

Basics

`string · number · boolean`

`string[]` or `Array<string>`

`[string, number]` – tuple

`'a' | 'b'` – union of literals

`unknown` – safe any

`never` – cannot happen

Shapes

`interface X { a: string }`

`type Y = { a: string }`

`a?: string` – optional

`readonly a: string`

`[key: string]: number` – index

Utility types

`Partial<T> · Required<T>`

`Pick<T, 'a'> · Omit<T, 'a'>`

`Record<K, V>`

`ReturnType<typeof f>`

Narrowing

`typeof x === 'string'`

`'a' in obj`

`x instanceof Cls`

`x is Foo` – type predicate

`'unknown'` forces you to check before using. `'any'` disables checking entirely — it is a hole in the type system, not a type.