Appendix A. Type Operators

TypeScript supports a rich set of type operators for working with types. Use Table A-1 as a handy reference for when you want to learn more about an operator.

Table A-1. Type operators

Type operator	Syntax	Use it on	Learn more
Type query	`typeof`, `instanceof`	Any type	"Refinement", "Classes Declare Both Values and Types"
Keys	`keyof`	Object types	"The keyof operator"
Property lookup	`O[K]`	Object types	"The keying-in operator"
Mapped type	`[K in O]`	Object types	"Mapped Types"
Modifier addition	`+`	Object types	"Mapped Types"
Modifier subtraction	`-`	Object types	"Mapped Types"
Read-only modifier	`readonly`	Object types, array types, tuple types	"Objects", "Classes and Inheritance", "Read-only arrays and tuples"
Optional modifier	`?`	Object types, tuple types, function parameter types	"Objects", "Tuples", "Optional and Default Parameters"
Conditional type	`?`	Generic types, type aliases, function	"Conditional Types"

Type operator	Syntax	Use it on	Learn more
		parameter types	
Non `null` assertion	`!`	Nullable types	“Nonnull Assertions”, “Definite Assignment Assertions”
Generic type parameter default	`=`	Generic types	“Generic Type Defaults”
Type assertion	`as` , `<>`	Any type	“Type Assertions”, “The const type”
Type guard	`is`	Function return types	“User-Defined Type Guards”