***Day 5 Operators***

1. **Keyof operator**

The keyof operator in TypeScript is used to extract the keys of an object type as a union of string literal types. It provides a way to ensure type safety when working with object properties dynamically.

function getProperty<T, K extends keyof T>(obj: T, key: K): T[K] {

    return obj[key];

}

1. **Overloading**

Define **multiple function signatures** for a single method, each with different parameter types or counts.

In TypeScript, **you define overloads with multiple function signatures**, and then provide **one actual implementation** that handles all cases.

speak(s: string): string;

speak(n: number): string;

speak(b: boolean): string;

These are the **overload declarations**. They tell TypeScript what calls are allowed

speak(arg: any): any {

    if (typeof arg === 'number') {

        return `Meow number ${arg}`;

    }

    if (typeof arg === 'string') {

        return `Meow string ${arg}`;

    }

    if (typeof arg === 'boolean') {

        return `Meow boolean ${arg}`;

    }

}

This is the **actual implementation** that handles **both overloads**. TypeScript only allows **one implementation**, and it must be compatible with all the declared signatures.