Sure, here are a few more examples showcasing additional features and use cases of enums in TypeScript:

Example 6: Using Enums with Switch Statements

```typescript
enum Day {
  Sunday,
  Monday,
  Tuesday,
  Wednesday,
  Thursday,
  Friday,
  Saturday,
}

function getDayName(day: Day): string {
  switch (day) {
    case Day.Sunday:
      return "Sunday";
    case Day.Monday:
      return "Monday";
    case Day.Tuesday:
      return "Tuesday";
    case Day.Wednesday:
      return "Wednesday";
    case Day.Thursday:
      return "Thursday";
    case Day.Friday:
      return "Friday";
    case Day.Saturday:
      return "Saturday";
    default:
      return "Invalid day";
  }
}

console.log(getDayName(Day.Monday)); // Output: Monday
```

Example 7: Using Enums with Objects

```typescript
enum Color {
  Red = "#FF0000",
  Green = "#00FF00",
  Blue = "#0000FF",
}

let favoriteColor = Color.Red;
console.log("My favorite color is", favoriteColor); // Output: My favorite color is #FF0000
```

Example 8: Combining Enums

```typescript
enum Animal {
  Dog,
  Cat,
}

enum Breed {
  Labrador,
  Poodle,
  Persian,
}

function getAnimalInfo(animal: Animal, breed: Breed): string {
  return `Animal: ${Animal[animal]}, Breed: ${Breed[breed]}`;
}

console.log(getAnimalInfo(Animal.Dog, Breed.Labrador)); // Output: Animal: Dog, Breed: Labrador
```

Example 9: Enums with Computed Values

```typescript
enum LogLevel {
  Error = 1,
  Warn = 2,
  Info = 3,
  Debug = 4,
}

function logMessage(level: LogLevel, message: string): void {
  if (level <= LogLevel.Warn) {
    console.log("[WARN]", message);
  } else if (level <= LogLevel.Info) {
    console.log("[INFO]", message);
  } else if (level <= LogLevel.Debug) {
    console.log("[DEBUG]", message);
  } else {
    console.log("[ERROR]", message);
  }
}

logMessage(LogLevel.Warn, "This is a warning message."); // Output: [WARN] This is a warning message.
```

These examples should give you a comprehensive understanding of how enums can be utilized in various scenarios within TypeScript, including switch statements, object properties, combining enums, and even using enums with computed values.