Here are some examples illustrating the use of enums in TypeScript:

Example 1: Basic Enum

```typescript
enum Direction {
  Up,
  Down,
  Left,
  Right,
}

let playerDirection: Direction;
playerDirection = Direction.Up;

console.log(playerDirection); // Output: 0
```

Example 2: Enum with Custom Values

```typescript
enum Direction {
  Up = 1,
  Down = 2,
  Left = 3,
  Right = 4,
}

let playerDirection: Direction;
playerDirection = Direction.Up;

console.log(playerDirection); // Output: 1
```

Example 3: Enum with String Values

```typescript
enum Direction {
  Up = "UP",
  Down = "DOWN",
  Left = "LEFT",
  Right = "RIGHT",
}
```

```
let playerDirection: Direction;
playerDirection = Direction.Up;

console.log(playerDirection); // Output: "UP"
```

Example 4: Using Enums in Functions

```typescript
enum Direction {
  Up,
  Down,
  Left,
  Right,
}

function move(direction: Direction): void {
  console.log("Moving", Direction[direction]);
}

move(Direction.Right); // Output: Moving Right
```

Example 5: Enum Iteration

```typescript
enum Direction {
  Up,
  Down,
  Left,
  Right,
}

for (let key in Direction) {
  if (isNaN(Number(key))) {
    console.log(key, "->", Direction[key]);
  }
}
```

Output:

```
Up -> 0
Down -> 1
```

```
Left -> 2
Right -> 3
```

These examples demonstrate various aspects of working with enums in TypeScript, including defining enums, assigning values, using enums as types, and iterating over enum members.