Here are five TypeScript function examples covering various scenarios:

1. **Simple Function**:
   ```typescript
   // Function to calculate the square of a number
   function square(num: number): number {
       return num * num;
   }

   // Usage
   console.log(square(5)); // Output: 25
   ```

2. **Function with Optional Parameter**:
   ```typescript
   // Function to greet a person with an optional prefix
   function greet(name: string, prefix?: string): void {
       if (prefix) {
           console.log(`${prefix}, ${name}!`);
       } else {
           console.log(`Hello, ${name}!`);
       }
   }

   // Usage
   greet("Alice"); // Output: Hello, Alice!
   greet("Bob", "Hi"); // Output: Hi, Bob!
   ```

3. **Function with Rest Parameters**:
   ```typescript
   // Function to calculate the sum of numbers
   function sum(...numbers: number[]): number {
       return numbers.reduce((total, num) => total + num, 0);
   }

   // Usage
   console.log(sum(1, 2, 3, 4, 5)); // Output: 15
   ```

4. **Higher-Order Function**:
   ```typescript
   // Higher-order function to perform an operation on two numbers
   ```

```typescript
function applyOperation(x: number, y: number, operation: (a: number, b: number) => number): number {
    return operation(x, y);
}

// Functions for operations
function add(a: number, b: number): number {
    return a + b;
}

function subtract(a: number, b: number): number {
    return a - b;
}

// Usage
console.log(applyOperation(5, 3, add)); // Output: 8
console.log(applyOperation(5, 3, subtract)); // Output: 2
```

5. **Recursive Function**:
```typescript
// Recursive function to calculate factorial
function factorial(n: number): number {
    if (n === 0 || n === 1) {
        return 1;
    } else {
        return n * factorial(n - 1);
    }
}

// Usage
console.log(factorial(5)); // Output: 120
```