

1. Project Setup

Create a new folder and initialize TypeScript:

```
sh
mkdir task-manager
cd task-manager
npm init -y
npm install typescript ts-node --save-dev
npx tsc --init
```

2. JavaScript Code (Before Conversion)

app.js (Original JavaScript Code)

```
javascript
const tasks = [];

function addTask(title, status = "pending") {
    tasks.push({ title, status });
}

function completeTask(index) {
    if (tasks[index]) {
        tasks[index].status = "completed";
    }
}

function listTasks() {
    tasks.forEach((task, index) => {
        console.log(`${index + 1}. ${task.title} - ${task.status}`);
    });
}

// Example Usage
addTask("Learn TypeScript");
addTask("Build a project");
completeTask(0);
listTasks();
```

3. Convert to TypeScript

app.ts (Converted TypeScript Code)

```
typescript
// Define Task Status Enum
enum TaskStatus {
    Pending = "pending",
    Completed = "completed",
}

// Define Task Object Type
type TaskType = {
    title: string;
    status: TaskStatus;
};

// Task Class
class Task {
    title: string;
    status: TaskStatus;

    constructor(title: string, status: TaskStatus = TaskStatus.Pending) {
        this.title = title;
        this.status = status;
    }

    markCompleted() {
        this.status = TaskStatus.Completed;
    }
}

// Task Manager Class
class TaskManager {
    private tasks: Task[] = [];

    addTask(title: string) {
        this.tasks.push(new Task(title));
    }

    completeTask(index: number) {
        if (this.tasks[index]) {

```

```
        this.tasks[index].markCompleted();
    } else {
        console.log("Invalid task index.");
    }
}

listTasks(): void {
    this.tasks.forEach((task, index) => {
        console.log(`${index + 1}. ${task.title} - ${task.status}`);
    });
}
}

// Example Usage
const taskManager = new TaskManager();
taskManager.addTask("Learn TypeScript");
taskManager.addTask("Build a project");
taskManager.completeTask(0);
taskManager.listTasks();
```

4. Compile & Run

Compile TypeScript to JavaScript

```
sh
npx tsc
```

1. This generates a compiled **app.js** file.

Run the JavaScript Output

```
sh
node app.js
```