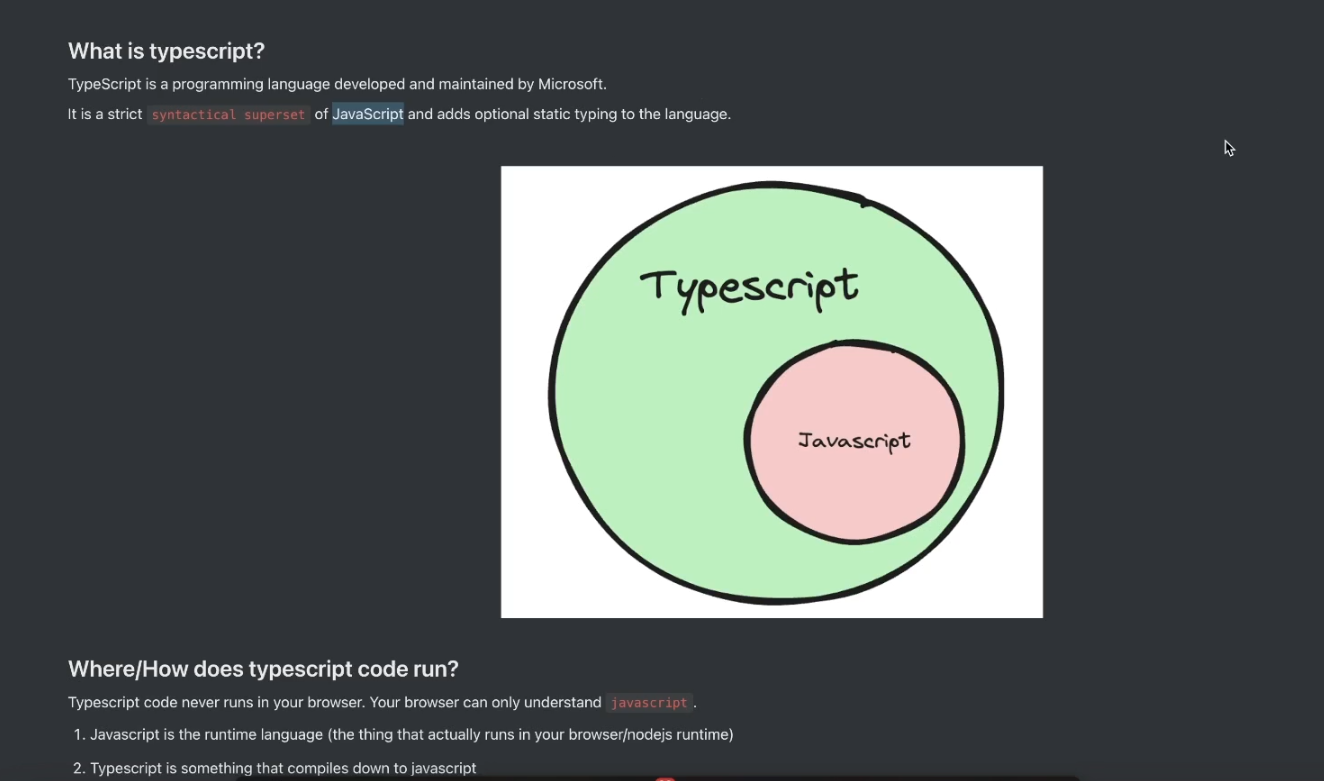
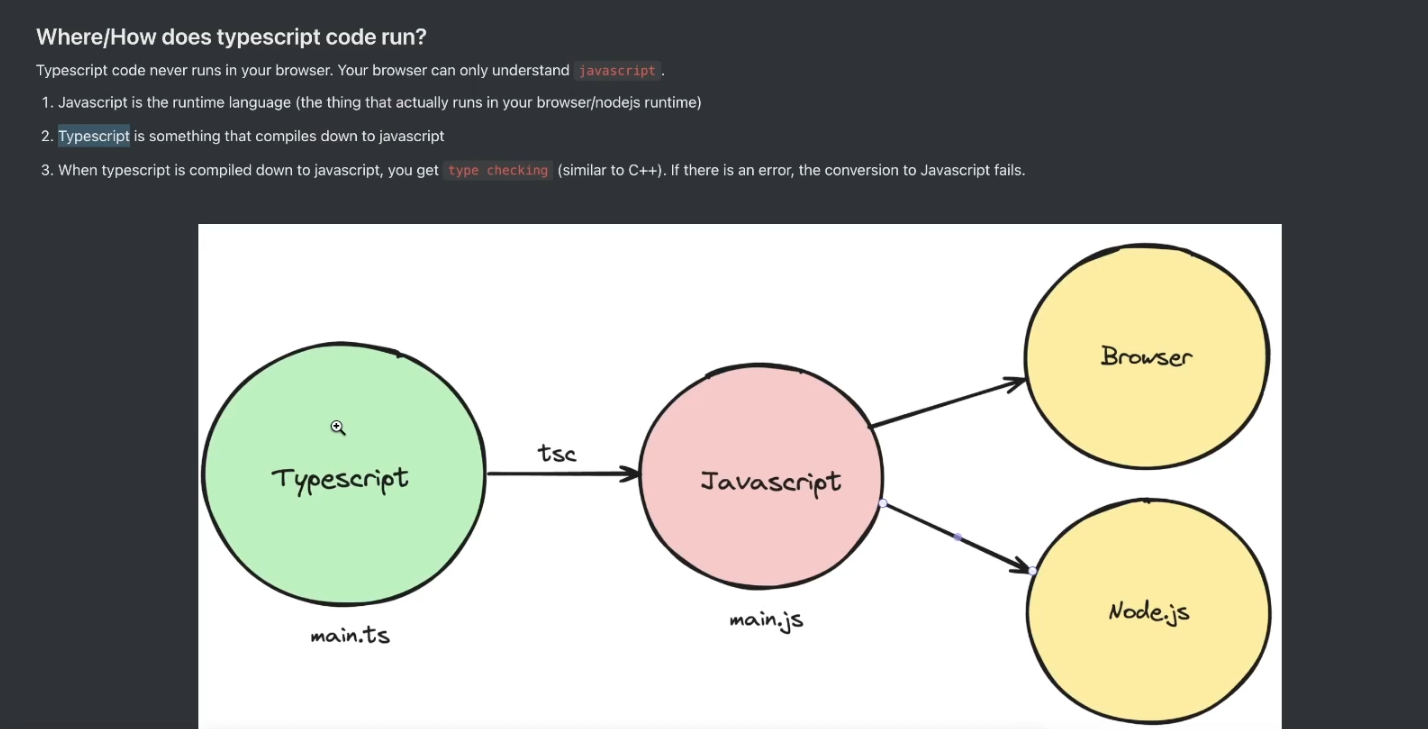
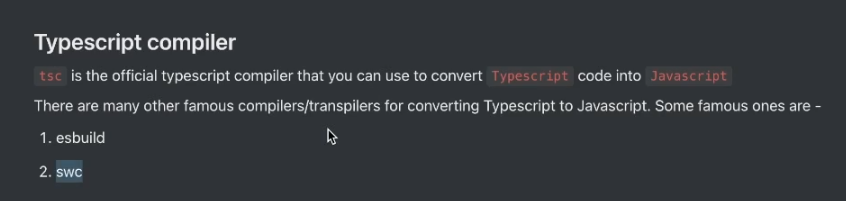
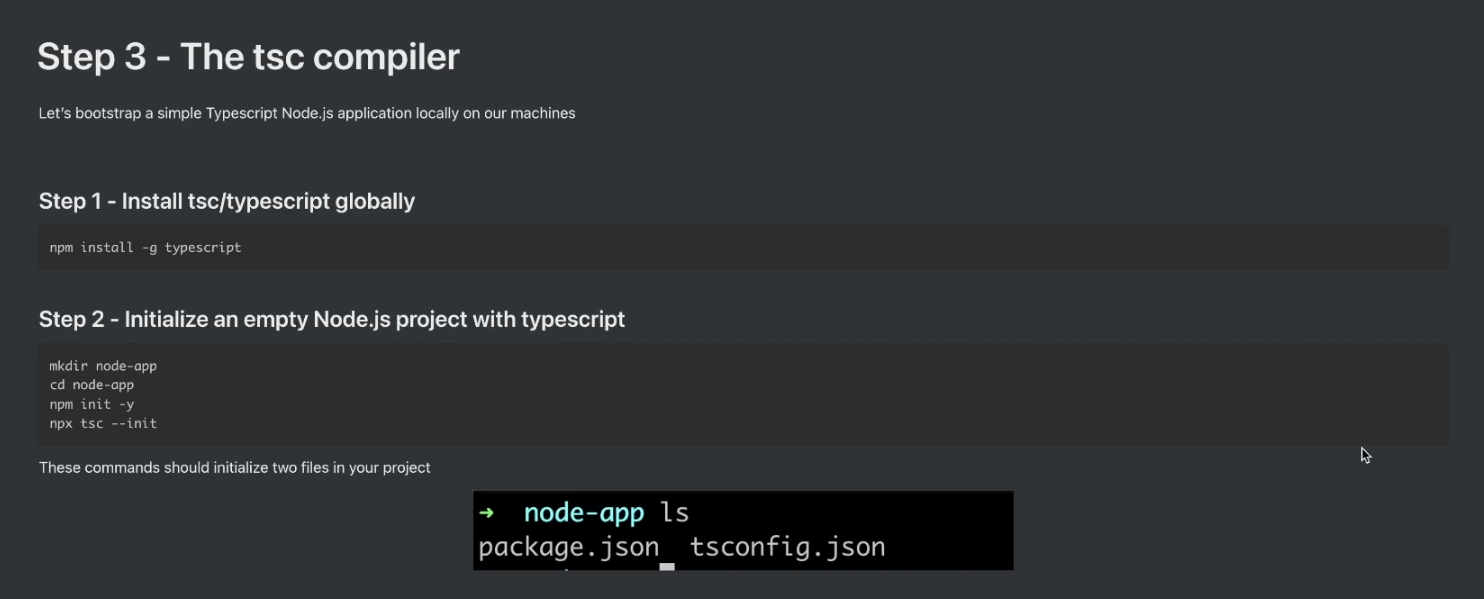
**TypeScript**

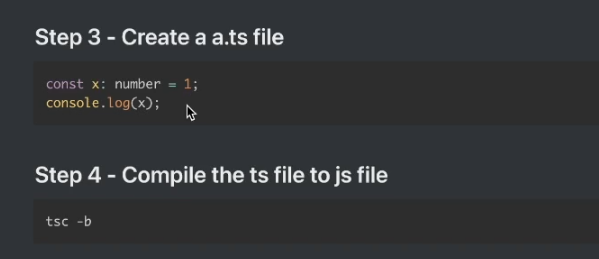
****

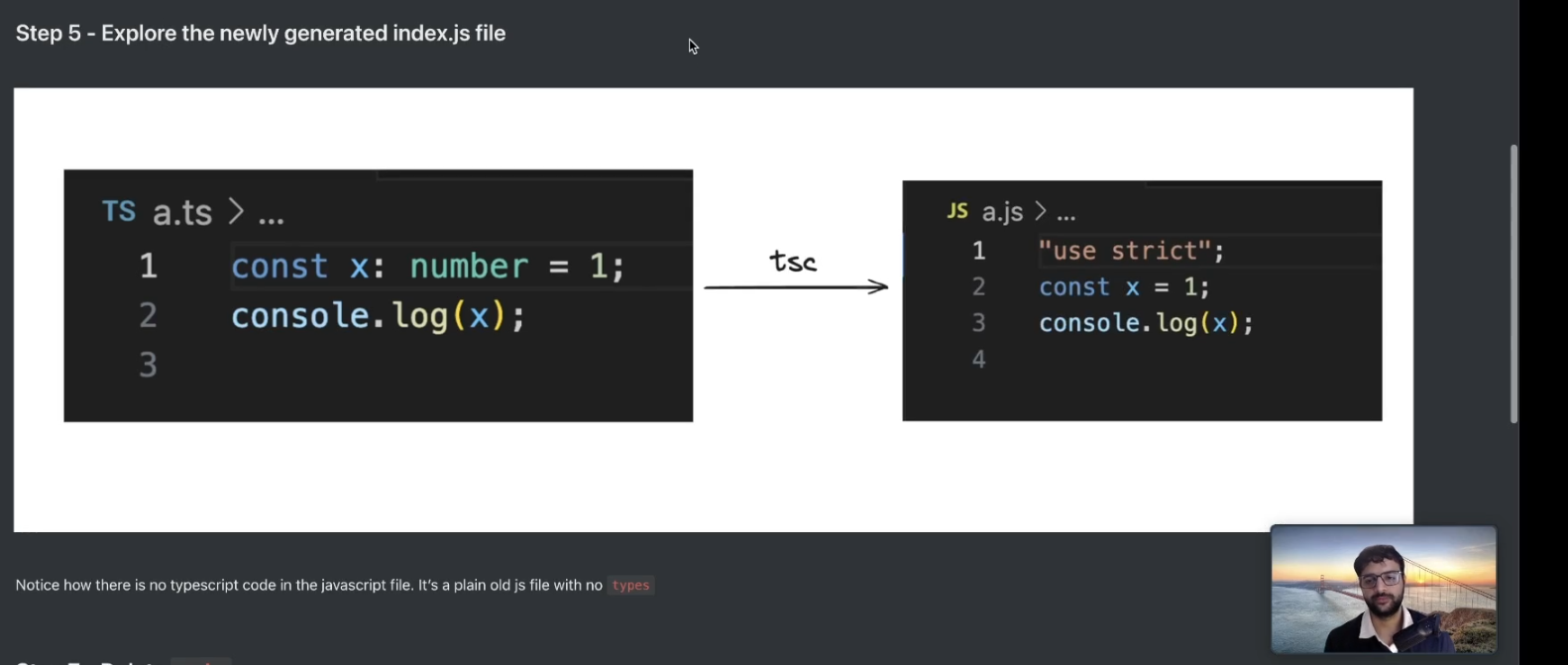












In TypeScript, you get all the JavaScript types **plus additional types and features** that improve safety and clarity.

**✅ Built-in Data Types in TypeScript**

**1. Primitive Types (same as JavaScript)**

ts

Copy code

let id: number = 1;

let name: string = "Alice";

let isActive: boolean = true;

let nothing: null = null;

let notDefined: undefined = undefined;

let symbolValue: symbol = Symbol("key");

let bigValue: bigint = 123n;

**2. Object Types**

ts

Copy code

let person: { name: string; age: number } = {

name: "Bob",

age: 30,

};

**3. Array**

ts

Copy code

let numbers: number[] = [1, 2, 3];

let names: Array<string> = ["Alice", "Bob"];

**4. Tuple**

ts

Copy code

let user: [number, string] = [1, "Alice"];

**5. Enum**

ts

Copy code

enum Status {

Pending,

InProgress,

Done,

}

let taskStatus: Status = Status.InProgress;

**6. Any**

Disables type checking — try to avoid unless necessary.

ts

Copy code

let data: any = "could be anything";

**7. Unknown**

Like any, but safer — forces you to check before using.

ts

Copy code

let input: unknown = "text";

if (typeof input === "string") {

console.log(input.toUpperCase());

}

**8. Void**

Used in functions that return nothing.

ts

Copy code

function logMessage(msg: string): void {

console.log(msg);

}

**9. Never**

Used for functions that never return (e.g., throw error or infinite loop).

ts

Copy code

function throwError(): never {

throw new Error("Something went wrong");

}

**10. Literal Types**

ts

Copy code

let direction: "left" | "right" = "left";

**11. Union & Intersection Types**

ts

Copy code

let value: string | number = "hello"; // union

type Admin = { role: string };

type User = { name: string };

type AdminUser = Admin & User; // intersection

**12. Function Types**

ts

Copy code

let add: (a: number, b: number) => number = (a, b) => a + b;

**Bonus: Type Aliases & Interfaces**

ts

Copy code

type User = {

id: number;

name: string;

};

interface Product {

id: number;

title: string;

}

