***Day 3 TypeScript***

TypeScript adds static typing to JavaScript, which helps **catch errors before runtime.**

* **Basic Types & Functions**:

Started with simple functions and type annotations

let name: string = "aaryan";

* **Interfaces**:

Learned how to define object structures (similar to Java interfaces)

interface User {

name: string;

id: number;

role: roles;

}

* **Custom Types**:

Created union types to restrict values

type roles = "Trainer" | "Developer" | "Tester";

* **Classes**:

Implemented classes with constructors and properties

class UserAccount {

name: string;

id: number;

}

* **Interface Implementation**:

Made classes implement interfaces

class Vendor implements VendorInterface {...}

* **Optional Properties**:

Used the ? symbol for optional properties

type Vendor = {

lname?: string; *// optional*

}

* **Arrays & Sorting**:

Worked with typed arrays and sorting functions

const vendors: Vendor[] = [...];

vendors.sort((a, b) => a.id - b.id);

* **Type 'any'**:

Used for values where the type isn't known

productName: any;

* **String Interpolation**:

Used template literals for string formatting

console.log(`ID: ${e.id}, Name: ${e.name}`);

* **Conditional Expressions**:

Implemented ternary operators

e.lname != undefined ? e.name + " " + e.lname : e.name

* **Promise to Get the response from API:**

Get the data from url in json format data and display the json data using Promise<Response>.

function getFacts(url: string): Promise<Response> {

   return fetch(url).then(res => res.json());

getFacts("[https://cataas.com/cat?width=200;height=200;json=true").then(data](https://cataas.com/cat?width=200;height=200;json=true%22).then(data) => console.log(data));