**TYPESCRİPT**

TypeScript stands in an unusual relationship to JavaScript. TypeScript offers all of JS’s features, and an additional layer on top of these: TS’ type system.

For Example, JS provides language primitives like string and number but it does not check that you have consistently assigned these. TS does. This means that your existing working JS code is also TS code. The main benefit of TS is that it can highlight unexpected behavior in your code, lowering the chance of bugs.

Readonly: Similar to arrays, the readonly keyword can prevent class members from being changed.

private readonly name: string;

Partial: Changes all the properties in an object to be optional.

let pointPart: Partial<Point>={};

Record: Record is a shortcut to defining an object type with a specific key type and value type.

Omit: Removes keys from an object type.

Pick : Removes all but the specified keys from an object type.

Exclude : Removes types from union.

keyof : when used on object type with explicit keys, keyof creates a uninon type with those keys.

…

Function printPersonProp(person:Person, property: keyof Person(){

Console.log(`Printing person prop ${property}: “${person[property]}”`);

}

let person ={

name:”max”

age: 27

}

prinPersonProp(person,”name”);