**Q1) What are some differences between interfaces and types in TypeScript?**

Answer: In typescript, to define the structure of data we use interfaces and types. But there is various differences between them as they differ in flexibility and usages. Some differences are given below:

1. By using interfaces we can not define primitive (Boolean, number, string, bigint, symbol), unions and tuple types of data. We can only define object types of data. But using type, we can define primitive type as well as object type of data.

Examle:

*//using interfaces*

  interface Vehicle {

    company: string;

    model: string;

  }

  const car: Vehicle = {

    company: "Toyota",

    model: "Camry",

  };

  console.log(car); *//output: { company: 'Toyota', model: 'Camry' }*

*//using type*

  type Fruit = string;

  const fruit = "Watermelon";

  console.log(fruit); *// logs "Watermelon"*

  type Fruits = {

    name: string;

    taste: string;

  };

  const fruits = {

    name: "Watermelon",

    taste: "Sweet",

  };

  console.log(fruits) *//logs { name: 'Watermelon', taste: 'Sweet' }*

1. In case of Interfaces two interfaces with the same name get merged. But in case of type two types with same name arise an exception.
2. Interfaces support implementing and extending union types. On the other hand type does not support implementing or extending union types.

How does TypeScript help in improving code quality and project maintainability?

Answer: Javascript is a dynamically typed language. This means we can assign any type of valy to a variable. For small projects this can be okay, but for larger projects type checking is necessary for error free code. Here comes the typescript, providing a strong typing system, this can be solved. Using type checking, its easier to find bugs. As javascript can catch error in runtime but in typescript we can catch error while writing code. Typescript also provides a huge benefit. That is for developers who learned js from es6 he/she might not know about older version of javascript. Some browsers might not support es6. So for that reason code needs to be converted to older versions of js. By using typescript we can convert code to any versions of js code. Thus can improve code quality and project maintainability.