1.What are the primitive types in TypeScript?

- boolean - true or false values.
- number - whole numbers and floating point values.
- string - text values like "TypeScript Rocks"

2.Explain how the arrays work in TypeScript.

1. Using square brackets. This method is similar to how you would declare arrays in JavaScript.

```
let fruits: string[] = ['Apple', 'Orange', 'Banana'];
```

2. Using a generic array type, Array<elementType>.

```
let fruits: Array<string> = ['Apple', 'Orange', 'Banana'];
```

Both methods produce the same output.

Of course, you can always initialize an array like shown below, but you will not get the advantage of TypeScript's type system.

```
let arr = [1, 3, 'Apple', 'Orange', 'Banana', true, false];
```

3.What is any type, and when to use it?

In Typescript, **any value can be assigned to the unknown type, but without a type assertion, unknown can't be assigned to anything but itself and the any type**. Similarly, no operations on a value with its type set as unknown are allowed without first asserting or restricting it to a more precise type.

4. What is void, and when to use the void type?

The void type denotes the absence of having any type at all. It is a little like the opposite of the any type.

Typically, you use the void type as the return type of functions that do not return a value.The syntax (a: string) => void means “**a function with one parameter, named a , of type string, that doesn't have a return value**”. Just like with function declarations, if a parameter type isn't specified, it's implicitly any . Note that the parameter name is required.

5. What is an unknown type, and when to use it in TypeScript?

The unknown type is commonly used to avoid the any type. Instead of having no type or any type, we assign it to unknown . Everything assigned to this type will result in an error unless you assign it to another type at some point.

6. What are the different keywords to declare variables TypeScript?

A **variable** is a named location in memory to store data. In TypeScript, an extension of Javascript, there are three different keywords to define variables: `var`, `let`, and `const`.

The **`var`** keyword is used to define a variable and has scope outside the block in which it is used. we can access it anywhere in the program.
The **`let`** keyword is used to define a variable whose scope is limited to the block it is defined in.
`const` is a keyword that is used to declare variables where the value of the variable is restricted to change.

7.Provide the syntax of a function with the type annotations.

We annotate a variable by using a colon (:) followed by its type.

Ex: var StringVar : any= "hello world";

```
var message : string = "hello world";       //string
var NumberVar: number = 100;                //number
var BooleanVar : boolean = true;           //boolean
var ArrayVar: string[]                     //arrays
```

8. How to create objects in TypeScript?

An **object** is an instance which contains set of key value pairs. The values can be scalar values or functions or even array of other objects.

```
2.      var object_name = {
3.       key1: “value1”, //scalar value
4.       key2: “value”,
5.       key3: function() {
6.          //functions
7.       },
8.       key4:[“content1”, “content2”] //collection
9.    };
```

9.How to specify optional properties in TypeScript?

the optional property is that properties can be undefined or null, and we can initialize them whenever required.

If try to use the data without getting it due to the database server being down or if there is any other problem, it will raise an error. In such case, we can make the data property optional and check that if data is available, only move ahead with other code.

the question mark is used to make the property optional. The optionalProp is the optional property in the below syntax.

```
interface sample {
  prop1: number;
  optionaProp?: string;
}
```

10. Explain the concept of null and its use in TypeScript.

'*null*' refers to the data type or value. The *null* is a keyword in TypeScript, which we can use to represent the absent or empty value. So, we can use '*null*' to define the variable's data-type or initialize the variable.

Users can follow the syntax below to use the *null* keyword as a data type or value.

```
let null_var: null = null;
```

In the above syntax, 'null' refers to the data type of the null_var named variable, and we have initialized it with the null value.

TypeScript is a type-strict language.

11.What is undefined in TypeScript?

Undefined is the default value for uninitialized variables
Whenever we declare a variable without initializing it with a value, TypeScript initializes it as undefined . But TypeScript never assigns null to any variable. We have to assign Null to variable to make it null.

12.Explain the purpose of the never type in TypeScript.

TypeScript introduced a new type never, which **indicates the values that will never occur**. The never type is used when you are sure that something is never going to occur. For example, you write a function which will not return to its end point or always throws an exception.

13. Explain how enums work in TypeScript

enums, or enumerated types, are data structures of constant length that hold a set of constant values. Each of these constant values is known as a member of the enum. Enums are useful when setting properties or values that can only be a certain number of possible values.

14.Explain the TypeScript class syntax.

A class keyword is used to declare a class in TypeScript. We can create a class with the following syntax:

```
1. class <class_name>{
2.     field;
3.     method;
4. }
```

15.Explain the arrow function syntax in TypeScript.

It has addition of data types or return types along with the function syntax and also along with the arguments passed inside that function.

```
let addition = (number1 : number, number2 : number) :
number => {
    return number1 + number2;
}
let function_name = (
    parameter_1 : data_type,
) : return_type => {
}
```