

VERSION CONTROL WITH GIT AND PROJECT IMPLEMENTATION➤ ASSIGNMENT 02Comprehensive Overview of TypeScript: Key Features, Syntax, And Usage

This document provides a concise summary of TypeScript, focusing on its key features, syntax, and practical usage. The following topics are covered:

1. Key Features of TypeScript

- Statically Typed: TypeScript introduces static typing, allowing developers to define variable types explicitly, which helps catch errors during development.
- Enhanced Tooling: Improved IDE support with features like autocompletion and type checking enhances developer productivity.
- Compatibility with JavaScript: TypeScript is a superset of JavaScript, meaning any valid JavaScript code is also valid TypeScript code.

2. Basic Syntax

- Type Annotations: Variables can have explicit types.
[typescript]
let age: number = 30;
- Type Inference: TypeScript can infer types automatically.
[typescript]
let name = "Alice"; // Inferred as string

3. Static Typing

- Benefits: Static typing reduces runtime errors and improves code clarity.
- Declaring Types: Use type annotations to declare variable types.
[typescript]
let isActive: boolean = true;

4. Interfaces

- Definition: Interfaces define the structure of objects.
- Usage: Create interfaces for Object Shapes:

[typescript]

```
interface User {
  name: string;
  age: number;
  email?: string; // optional property
}
```

5. Classes

- OOP concepts: TypeScript supports object-oriented programming with classes.

- Defining classes:

[typescript]

```
class Person {
  constructor(public name: string) {}
  greet() {
    console.log('Hello, my name is ' + this.name);
  }
}
```

- Inheritance (classes can extend other classes)

[typescript]

```
class Employee extends Person {
  constructor(name: string, public jobTitle: string) {
    super(name);
  }
  greet() {
    console.log('Hello, I'm ' + this.name + ' and I work as a ' + this.jobTitle);
  }
}
```

6. Enums

- Introduction: Enums allow you to create components that maintain type safety.

- Example:

[typescript]

```
enum Gender {
  Male, Female, Other
}
```

7. Advanced TypeScript Concepts

B.Tech. [2nd year] Student

- Union Types: Allow variables to hold multiple types:

typescript

let value: string | number;

- Intersection Types: Combine multiple Types:

typescript

```
type Admin = { admin: true };  
type User = { name: string };  
type AdminUser = Admin & User;
```

- Type Guards: Narrow down union types at runtime:

typescript

```
function printId (id: number | string) {  
  if (typeof id === "string") {  
    console.log (id.toUpperCase());  
  } else {  
    console.log (id);  
  }  
}
```

◆ Conclusion

This document summarizes the essential features and syntax of TypeScript, highlighting its advantages over plain JavaScript. By understanding static typing, interfaces, classes, generics, and advanced concepts, I can leverage TypeScript to write safer and more maintainable code in my projects.