LE INTERFACCE IN

1. DEFINIZIONE DELLE INTERFACCE

CONCETTO DI INTERFACCE

LE INTERFACCE IN TYPESCRIPT SONO DEI MODELLI CHE DEFINISCONO LA STRUTTURA DI OGGETTI E DATI

SINTASSI DELLE INTERFACCE

STRUTTURA DELLE INTERFACCE

LE INTERFACCE IN TYPESCRIPT SEGUONO UNA SPECIFICA STRUTTURA CHE INCLUDE PROPRIETÀ E METODI

ESEMPIO DI INTERFACCIA

UN ESEMPIO DI INTERFACCIA IN TYPESCRIPT È QUELLA CHE DEFINISCE UN OGGETTO DI TIPO UTENTE CON PROPRIETÀ E METODI SPECIFICI

```
interface Utente {
nome: string;
cognome: string;
eta: number;
stampaDati(): void;
}
```

UTILITÀ DELLE INTERFACCE

LE INTERFACCE IN TYPESCRIPT MIGLIORANO LA LEGGIBILITÀ DEL CODICE, LO RENDONO PIÙ ROBUSTO, FAVORISCONO LA RIUTILIZZABILITÀ E LA COLLABORAZIONE TRA SVILUPPATORI

2. FUNZIONALITÀ DELLE INTERFACCE

EREDITARIETÀ

LE INTERFACCE POSSONO EREDITARE DA ALTRE INTERFACCE, MENTRE I TIPI PERSONALIZZATI NO

```
interface Animale {
nome: string;
eta: number;
mangiare(): void; }

interface Cane extends
Animale {
razza: string;
abbaiare(): void;
}
```

UNIONE

LE INTERFACCE POSSONO ESSERE UNITE PER CREARE NUOVE INTERFACCE, MENTRE I TIPI PERSONALIZZATI NO

```
interface Voliante {
volare(): void; }

interface Nuotante {
nuotare(): void; }

interface Anfibio extends
Voliante, Nuotante {}
```

RIAPERTURA

LE INTERFACCE POSSONO ESSERE RIAPERTE PER AGGIUNGERE NUOVE PROPRIETÀ O METODI, MENTRE I TIPI PERSONALIZZATI NO

LE INTERFACCE IN

SINTASSI

```
interface NomeInterfaccia
{
proprietà1: tipoDato1;
proprietà2: tipoDato2;
...
metodo1(parametro1:
tipoDato1, parametro2:
tipoDato2): tipoDato3;

metodo2(): void;
...
}
```