MODIFICATORI DI ACCESSO IN TYPESCRIPT

1. PUBLIC

ACCESSIBILITÀ

LA PROPRIETÀ O IL METODO È ACCESSIBILE DA OVUNQUE

```
classExample {
public name: string;
constructor(name: string) {
this.name = name; }
publicgreet() {
console.log(`Hello, my
name is ${this.name}`);
}
}
const example =
newExample('Alice');
console.log(example.name);
// Accessibile dall'esterno
example.greet(); //
Accessibile dall'esterno
```

E' IL COMPORTAMENTO PREDEFINITO (possiamo quindi ometterlo)

2. PRIVATE

ACCESSIBILITÀ

LA PROPRIETÀ O IL METODO È ACCESSIBILE SOLO ALL'INTERNO DELLA CLASSE IN CUI È DEFINITO

```
classExample {
private secret: string;
constructor(secret: string) {
this.secret = secret; }
public revealSecret() {
console.log(`The secret is
${this.secret}`);
}
}
const example =
newExample('I love
TypeScript');
//
console.log(example.secret);
// Errore: non accessibile
dall'esterno
example.revealSecret(); //
Accessibile perché è un
metodo pubblico
```

NON ACCESSIBILE DALLE SOTTOCLASSI O DALL'ESTERNO DELLA CLASSE

NON È POSSIBILE ACCEDERE ALLA PROPRIETÀ O AL METODO DALLE SOTTOCLASSI O DALL'ESTERNO DELLA CLASSE

3. READONLY

MODIFICABILITÀ

LA PROPRIETÀ PUÒ ESSERE ASSEGNATA SOLO UNA VOLTA, TIPICAMENTE NEL COSTRUTTORE, E NON PUÒ ESSERE MODIFICATA SUCCESSIVAMENTE

```
classExample {
public readonly id:
number;
constructor(id: number) {
this.id = id; }
publicshowId()
{ console.log(`ID:
${this.id}`);
}
}
const example =
newExample(123);
console.log(example.id); //
Accessibile dall'esterno//
example.id = 456; //
Errore: non può essere
modificato perché è
readonly
example.showId(); //
Accessibile dall'esterno
```

ACCESSIBILITÀ

PUÒ ESSERE COMBINATO CON PUBLIC, PRIVATE O PROTECTED PER DEFINIRE ANCHE IL LIVELLO DI ACCESSIBILITÀ

MODIFICATORI DI ACCESSO IN TYPESCRIPT

4. RIEPILOGO

PUBLIC

LA PROPRIETÀ O IL METODO È ACCESSIBILE OVUNQUE

PRIVATE

LA PROPRIETÀ O IL METODO È ACCESSIBILE SOLO ALL'INTERNO DELLA STESSA CLASSE

READONLY

LA PROPRIETÀ PUÒ ESSERE ASSEGNATA SOLO UNA VOLTA E NON PUÒ ESSERE MODIFICATA SUCCESSIVAMENTE