

TypeScript



[GitHub Repo](#)

[View code on CodePen](#)

Marcello Comandulli

Front End Web Developing Student

Contents

Code Analysis 03
---------------	----------

TS CODE ANALYSIS

I created 3 different interfaces with their properties and methods.

1. **Istartup** that is implemented everytime I need to add a StartUp that recives a new Incentive
2. **IIncentivo** that is implemented everytime I need to add a new Incentive given by the Govern
3. **Icittadino** that is implemented everytime a citizen join a sport event organized by a StartUp

```
1  //Creazione interfacce
2  interface IStartup {
3      nome: string;
4      settoreDiFocus: string;
5      descrizione: string;
6      servizi: string[];
7
8      riceviIncentivo(incentivo: IIncentivo): void;
9  }
10
11 interface IIncentivo {
12     codiceIdentificativo: string;
13     descrizione: string;
14     valoreIncentivo: number;
15     criteriEleggibilita: string;
16
17     assegnaAStartup(startup: IStartup): void;
18 }
19
20 interface ICittadino {
21     nome: string;
22     cognome: string;
23     eta: number;
24     interessiSportivi: string[];
25
26     partecipaAttivita(startup: IStartup): void;
27 }
```

TS CODE ANALYSIS

I created **3 Classes** with their respective Interface and methods instanced.

I used public properties inside the class **constructor** because the code is clearer and I didn't need them private or protected.

```
29 //Creazione classi
30 class Startup implements IStartup {
31     constructor (public nome: string,
32     public settoreDifocus: string,
33     public descrizione: string,
34     public servizi: string[])
35     {}
36
37     riceviIncentivo(incentivo: IIncentivo): void {
38         console.log(`La Startup ${this.nome} ha ricevuto un incentivo di: €${incentivo.valoreIncentivo}.`);
39     }
40 }
41
42 class Incentivo implements IIncentivo {
43     constructor (public codiceIdentificativo: string,
44     public descrizione: string,
45     public valoreIncentivo: number,
46     public criteriEleggibilita: string)
47     {}
48
49     assegnaAStartup(startup: IStartup): void {
50         console.log(`L'incentivo n. ${this.codiceIdentificativo} di €${this.valoreIncentivo} è stato assegnato a questa Startup: ${startup.nome}.`);
51     }
52 }
53
54 class Cittadino implements ICittadino {
55     constructor (public nome: string,
56     public cognome: string,
57     public eta: number,
58     public interessiSportivi: string[]){}
59
60     partecipaAttivita(startup: IStartup): void {
61         console.log(`Il cittadino ${this.nome} ${this.cognome} partecipa all'attività di ${this.interessiSportivi[0]} con questa Startup: ${startup.nome}.`);
62     }
63 }
```

TS CODE ANALYSIS

In the end I created **2 instances** for every class to check if everything was ok.

The **program run normally** and everything works.

```
65 //Istanze
66 const sporter = new Startup("Sporter", "Tech", "Nuova azienda tecnologica di sport italiana", ["Attrezzi sportivi", "Integratori", "Sensori"])
67
68 const salewa = new Startup("Salewa", "sport", "azienda numero uno per la montagna", ["Scarponi", "Abbigliamento tecnico", "Zaini"])
69
70
71 const incentivoTecnico = new Incentivo("INC503", "supporto tecnico per l'integrazione di nuove funzionalità", 1000, 'app per il fitness');
72
73 const incentivoFiscale = new Incentivo("INC429", "Innovazioni fiscali per l'innovazione nel settore sportivo", 3000, "attrezzature sportive innovative")
74
75
76 const mario = new Cittadino("Mario", "Rossi", 40, ["calcio", "rugby", "pallavolo"])
77 const serena = new Cittadino("Serena", "Blu", 30, ["paracadutismo", "sci", "arrampicata"])
78
79 sporter.riceviIncentivo(incentivoTecnico)
80 salewa.riceviIncentivo(incentivoFiscale)
81
82 incentivoTecnico.assegnaAStartup(sporter)
83 incentivoFiscale.assegnaAStartup(salewa)
84
85 mario.partecipaAttivita(sporter)
86 serena.partecipaAttivita(salewa)
```