

Namespace in TypeScript

Docente

Mario Perna

prof.perna.mario@darzo.net

A.S.

2025/2026

Materia

TEPSIT (Laboratorio)

Introduzione

Un "**namespace**" è come un contenitore che raggruppa vari elementi (come funzioni, variabili e classi) sotto un nome comune all'interno di un contesto specifico.

L'obiettivo principale di un namespace è evitare conflitti di nomi, poiché fornisce un modo di organizzare e categorizzare elementi in gruppi separati.

Il Namespace in TypeScript

Sintassi di base:

```
namespace Identificatore {  
    let variabile: tipo;  
    export function funzione() {  
        // Istruzioni  
    }  
    export class Classe {  
        // Istruzioni  
    }  
}
```

II Namespace in TypeScript

Consideriamo questo esempio:

```
namespace Colori {  
    let b: string = "FF";  
  
    export function coloreBlu(r: string, g: string): any {  
        return document.body.style.backgroundColor = "#" + r + g + b;  
    }  
}  
  
window.onload = () => {  
    let btn: HTMLButtonElement = document.getElementById("btn") as HTMLButtonElement;  
    btn.onclick = () => Colori.coloreBlu("00", "FF");  
};
```

Il Namespace in TypeScript

Con la parola chiave **namespace** definiamo lo spazio in cui verranno definiti gli elementi; a seguire definiamo l'identificatore.

```
namespace Colori {  
    let b: string = "FF";  
  
    export function coloreBlu(r: string, g: string): any {  
        return document.body.style.backgroundColor = "#" + r + g + b;  
    }  
}
```

Il Namespace in TypeScript

Con la parola chiave **export** facciamo sì che la funzione sia accessibile quando verrà invocata tramite l'identificatore "Colori".

```
namespace Colori {  
    let b: string = "FF";  
  
    export function coloreBlu(r: string, g: string): any {  
        return document.body.style.backgroundColor = "#" + r + g + b;  
    }  
}
```

Il Namespace in TypeScript

Infine, nella funzione "onload" viene richiamata la funzione "coloreBlu" tramite la classica struttura delle **arrow function**.

```
window.onload = () => {  
    let btn: HTMLButtonElement = document.getElementById("btn") as HTMLButtonElement;  
    btn.onclick = () => Colori.coloreBlu("00", "FF");  
};
```

Il Namespace in TypeScript

È anche possibile annidare una gerarchia di namespace dentro altre.

```
namespace Colori {  
    export namespace Rosso {  
        let r: string = "FF";  
        export function funzColore(g: string, b: string): any {  
            return document.body.style.backgroundColor = "#" + r + g + b;  
        }  
    }  
    export namespace Verde {  
        let g: number = 255;  
        export function funzColore(r: number, b: number): any {  
            return document.body.style.backgroundColor = "rgb("+r+", "+g+", "+b+")";  
        }  
    }  
    export namespace Blu {  
        export function funzColore(colore: string): any {  
            return document.body.style.backgroundColor = colore;  
        }  
    }  
}
```


II Namespace in TypeScript - Esempio

```
window.onload = () => {  
    let btn1: HTMLButtonElement = document.getElementById("btn1") as HTMLButtonElement;  
    let btn2: HTMLButtonElement = document.getElementById("btn2") as HTMLButtonElement;  
    let btn3: HTMLButtonElement = document.getElementById("btn3") as HTMLButtonElement;  
    btn1.onclick = () => Colori.Rosso.funzColore("00", "FF");  
    btn2.onclick = () => Colori.Verde.funzColore(128, 100);  
    btn3.onclick = () => Colori.Blu.funzColore("orchid");  
};
```

Sitografia

<https://profsiciliano.altervista.org>

<https://www.typescriptlang.org>