

# Övningar - Typescript

## Komma igång

1. Skapa en katalog
2. Öppna med VSCode
3. Kör i terminalen:

```
npm init
npm install typescript
npm install nodemon
npm install ts-node
```

4. Skapa tsconfig.json

```
{
  "compilerOptions": {
    "target": "es2017",
    "module": "CommonJS",
    "lib": ["es2017", "es7", "es6", "dom"],
    "declaration": true,
    "outDir": "dist",
    "strict": true,
    "esModuleInterop": true,
    "typeRoots": ["./types"]
  },
  "exclude": [
    "node_modules",
    "dist"
  ]
}
```

5. Skapa en fil för att köra din kod, t.ex. code.ts
6. Lägg till start-script i package.json:
7. Kör **npm start**

**Nu är det bara att börja skriva typescript-kod!**

## 1) Typa variabler / objekt

```
//Skriv av och ersätt alla any med en passande typ

const animal: any = "bird";

const animals: any = [ "bird", "cat", "lion" ];

const age: any = 45;

const isOpen: any = false;

const sum = (tal1: any, tal2: any) : any => tal1 + tal2;

const sayHello = (name: any): any => { console.log(`Hej ${name}!`) }

const room: any = {
  name: "Stora konferensen",
  seats: 50,
  dropInAvailable: false,
  hourlyFee: 350
}
```

## 2) Enum

```
const trafficLight = (color: string) => {
  switch (color) {
    case "Red":
      console.log('Du måste stanna!');
      break;
    case "Yellow":
      console.log('Kör om du inte hinner stanna!');
      break;
    case "Green":
      console.log('Kör så det ryker!');
      break;
  }
}

trafficLight("Red");

// a) Skapa en enum=> Color så att du istället anropar funktionen så här:

trafficLight(Color.Red);

// b) Ändra funktionen trafficLight så att den tar emot ett värde av typen Color

// c) Skriv om koden i funktionen så att den använder enum istället för string i switch (color)
```

### 3) Funktion

```
const getPersonList = (firstname: any, lastname: any, yearOfBirth: any, isFemale: any) : any => {
  return `


    <li><b>Förnamn:</b> ${firstname}</li>
    <li><b>Efternamn:</b> ${lastname}</li>
    <li><b>Födelseår:</b>${yearOfBirth}</li>
    <li><b>Man / kvinna:</b>${isFemale ? "Kvinna" : "Man"}</li>
  </ul>`;
}

const personHTML: string = getPersonList("Jonatan", "Hallenberg", 1987, false);

console.log(getPersonList("Jonatan", "Hallenberg", 1987, false));

//Ersätt alla any och lägg till rätt typer för inparametrar samt för returvärde
```

### 4) Interface

```
const getPersonList = (firstname: any, lastname: any, yearOfBirth: any, isFemale: any) : any => {  
  return `

  
- <b>Förnamn:</b> ${firstname}</li>  
- <b>Efternamn:</b> ${lastname}</li>  
- <b>Födelseår:</b>${yearOfBirth}</li>  
- <b>Man / kvinna:</b>${isFemale ? "Kvinna" : "Man"}</li>  
  </ul>`;  
}

```

```
// a. Skriv ett interface => Person som innehåller all data som funktionen getPersonList tar emot.  
// b. Skapa ett objekt av typen Person och lägg till alla attribut  
// c. Ändra getPersonList så att den bara tar emot en inparameter => getPersonList(person: Person)  
// d. Uppdatera koden i getPersonList  
// e. Skriv kod för att anropa getPersonList med objektet du skapade i punkt b)  
//    - lägg koden i console.log för att se att det funkar
```