

Övningsuppgift 4

I denna ävningen skall du träna på att använda TypeScript. Du skall med andra ord lösa uppgiften genom att deklarera typer för variabler och funktioner.

I uppgiften arbetar vi med Pokemon kort och skall implementera funktionalitet för att

- skapa en ny pokemon
- spara våra Pokemons till en fil
- söka efter en pokemon
- läsa våra Pokemons från en fil

Skapa det grundläggande

1. Skapa en ny ts-fil i din arbetskatalog (src) - du kan kalla den pokemon.ts.
2. I GitHub (klassrepot) finns det en fil som heter pokemondb.ts - kopiera den till din src katalog. Titta på den och försök förstå vad den gör.
3. Skapa en package.json fil i din katalog (samma katalog som tsconfig.json finns i) genom att köra "npm init". Du kan trycka på enter och välja alla förslagna svar i frågorna som dyker upp på skärmen.
4. Lägg till följande i slutet (efter *license* - som ett nytt attribut)

```
"dependencies": {  
  "@types/node": "18.11.15"  
}
```

Detta gör att vi får typ-definitioner till TypeScript för Node.js. Detta vill man göra om man skriver TypeScript kod för NodeJs.

```
1  {  
2    "name": "l2-typescript",  
3    "version": "1.0.0",  
4    "description": "",  
5    "main": "index.js",  
6    "scripts": {  
7      "test": "echo \"Error: no test specified\" && exit 1"  
8    },  
9    "author": "",  
10   "license": "ISC",  
11   "dependencies": {  
12     "@types/node": "18.11.15"  
13   }  
14 }  
15
```

5. Installera alla dependencies genom att köra "npm install".

Skapa en Pokemon

1. Skapa en egen typ (type) som du kallar för Pokemon. Den skall ha tre attribut:
 - name - sträng
 - hp - nummer
 - type - valfri sträng
2. Skapa en funktion createPokemon()
 - Funktionen tar tre parametrar
 - name - sträng
 - hp - nummer
 - type - valfri sträng
 - Funktionen returnerar en Pokemon (typen som du skapade i 1)
 - Skriv kod som skapar en Pokemon (objekt) och populerar attributen från funktionens parametrar. Det skapade objektet skall returneras.
3. Skapa en variabel som är en array av Pokemons. Vi kan kalla den för myPokemons. Denna skall deklareras globalt eftersom den kommer att innehålla våra Pokemons efterhand som vi skapar dem.
4. Skapa två Pokemons med hjälp av createPokemon() funktionen från övning 2 - spara dem till din myPokemons array på valfritt sätt.

Spara Pokemons till en fil

1. Importera savePokemonToFile från "Pokemondb".

```
import { savePokemonToFile } from "./pokemondb";
```

2. Spara din myPokemon array som du skapade i förra uppgiften till en fil som du kallar för "minapokemons.json". Använd savePokemonToFile funktionen som du importerade i 1. Tänk på att funktionen förväntar sig en sträng - du behöver därför konvertera din myPokemon array till ett JSON objekt innan du använder den funktionen.
3. Hittar du dina Pokemon i filen efter att du har sparat den? Det skall bli en text fil som du kan öppna i visual studio. Filen kommer att skapas relativt till den mappen du kör node ifrån.

Sök efter en Pokemon

1. Skapa en funktion searchPokemon()
 - Funktionen tar två parametrar
 - searchName - sträng
 - database - en array av Pokemons

- Funktionen returnerar en Pokemon (den första den kan hitta) eller null om ingen pokemon kan hittas.
 - Skriv kod som söker i "database" efter en Pokemon som har namnet "searchName". Om den hittar en eller flera träffar skall den första returneras. Om den inte hittar någon Pokemon som matchar skall den returnera null.
2. Använd searchPokemon() - funktionen för
 1. Hitta en Pokemon i din myPokemons och skriva ut den till consollen.
 2. Leta efter en Pokemon som inte finns och skriva ut att du inte hittad den i consollen.

Läsa in Pokemons från en fil

1. I kurs repot finns det en fil som heter extra.json. Kopiera den till mappen du kör koden ifrån.
2. Importera loadPokemonFromFile från pokemondb.
3. Läs in innehållet från extra.json med hjälp av loadPokemonFromFile. Om du lyckades konvertera din inlästa strängen till en ny array med hjälp av JSON.parse.
4. Använd kod som vi har skrivit tidigare för att hitta och skriva ut "Eva zu Beck" från den inlästa datan.