

Typescript övningar

Gå till: <https://www.typescriptlang.org/play/>, här kan ni utföra övningarna i detta dokument. Alternativt följ instruktionerna från genomgången för att skapa ett lokalt projekt på din dator.

TypeScript Övningar – Deklaration av Variabler och Grundläggande Typning

Övning 1: Deklaration av Variabler

Uppgift:

1. Skapa typade konstant-variabler för varje typ: boolean, number, string
2. Skapa en variabel med namnet `message` utan att ange typ explicit, sätt värdet till `"Hello World"`. Skapa en ny variabel `msglength` av typen number. Sätt den till värdet av längden på `message`

Lösning 1: Deklaration av Variabler

```
const isActive: boolean = false;  
const launchYear: number = 2022;  
const userName: string = "Anna";  
  
let message = "Hello world";  
let messageLength: number = message.length;  
console.log(isActive, launchYear, userName, messageLength);
```

Övning 2: Type Assertion

Uppgift:

Skapa en variabel med namnet `unknownValue` av typen `unknown` och sätt värdet till `"Unknown"`. Använd sedan type assertion för att skapa en variabel `textLength` av typen `number` som innehåller längden på `unknownValue` när den tolkas som en sträng.

Lösning 2: Type Assertion

```
let unknownValue: unknown = "Generiska meddelanden";  
let textLength: number = (unknownValue as string).length;  
console.log(textLength);
```

Övning 3: Arrayer och Listor

Uppgift:

1. Skapa en array med namnet `fruitList` som innehåller strängar
2. Skapa en array med namnet `releaseYears` som innehåller tal

Lösning 3: Arrayer och Listor

```
const fruitList: string[] = ["äpple", "päron", "banan"];  
const releaseYears: number[] = [2001, 2008, 2015];  
console.log(fruitList, releaseYears);
```


Övning 4: Union Types

Uppgift:

Deklarera en variabel med namnet `mixedValue` som kan vara antingen number, string eller boolean. Tilldela den olika värden.

Lösning 4: Union Types

```
let mixedValue: number | string | boolean;

mixedValue = 25;          // OK
console.log(mixedValue);

mixedValue = "tjugo-fem"; // OK
console.log(mixedValue);

mixedValue = true;        // OK
console.log(mixedValue);

// mixedValue = { value: 25 }; // Detta ger ett fel, ej tillåtet
```

Övning 5: Tuples

Uppgift:

Skapa en tuple med namnet `orderInfo` som representerar ett ordernummer (number) och en produktnamn (string). Använd värdena `[1001, "Smartphone"]` och logga ordernumret.

Lösning 5: Tuples

```
let orderInfo: [number, string] = [1001, "Smartphone"];  
console.log(orderInfo[0]); // Förväntat: 1001
```

Övning 6: Enums

Uppgift:

Skapa ett enum med namnet `Direction` med värdena `North`, `East`, `South` och `West`. Skapa en variabel `currentDir` av typen `Direction` och sätt den till `Direction.East`. Logga `currentDir`

Lösning 6: Enums

```
enum Direction { North, East, South, West }  
const currentDir: Direction = Direction.East;  
console.log(currentDir); // Förväntat: 1 (om North = 0)
```

Övning 7: Funktioner med Typade Parametrar och Returvärden

Uppgift:

Skapa en funktion med namnet `isValid` som tar en parameter `email` (string) och returnerar en boolean. Funktionen ska returnera `true` om `email` innehåller tecknet `"@"`, annars `false`. Sätt returvärdet explicit till `boolean`.

Lösning 7: Funktioner med Typade Parametrar

```
const isValidEmail = (email: string): boolean => email.includes("@");  
console.log(isValidEmail("test@example.com")); // true  
console.log(isValidEmail("testexample.com"));  // false
```


Övning 8: Funktioner utan Returvärde (void)

Uppgift:

Skapa en funktion med namnet `printStatus` som tar en parameter `status` (string) och loggar den till konsolen. Funktionen ska ha returtypen `void`.

Lösning 8: Funktioner utan Returvärde

```
const printStatus = (status: string): void => {  
  console.log(status);  
};  
printStatus("Systemet är online");
```

Övning 9: Valfria Parametrar

Uppgift:

Skapa en funktion med namnet `welcomeUser` som tar en obligatorisk parameter `user` (string) och en valfri parameter `salutation` (string). Om `salutation` ges, logga `${salutation}, ${user}!`, annars logga `Hello, ${user}!`. Testa funktionen med båda varianterna.

Lösning 9: Valfria Parametrar

```
const welcomeUser = (user: string, salutation?: string): void => {  
  if (salutation) {  
    console.log(`${salutation}, ${user}!`);  
  } else {  
    console.log(`Hello, ${user}!`);  
  }  
};  
  
welcomeUser("Emma", "Greetings");  
welcomeUser("Emma");
```

Övning 10: Typning av Objekt

Uppgift:

Skapa en type alias med namnet `User` med egenskaperna `firstName` (string), `lastName` (string) och en valfri egenskap `age` (number). Skapa sedan en instans med namnet `userInstance` med bara `firstName` och `lastName`.

Lösning 10: Typning av Objekt

```
type User = {  
  firstName: string;  
  lastName: string;  
  age?: number;  
};  
  
const userInstance: User = {  
  firstName: "Lina",  
  lastName: "Andersson"  
};  
  
console.log(userInstance);
```

