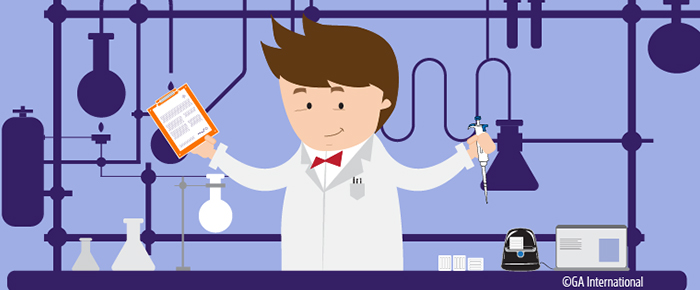
05. Labwerk Basis TypeScript met oplossingen

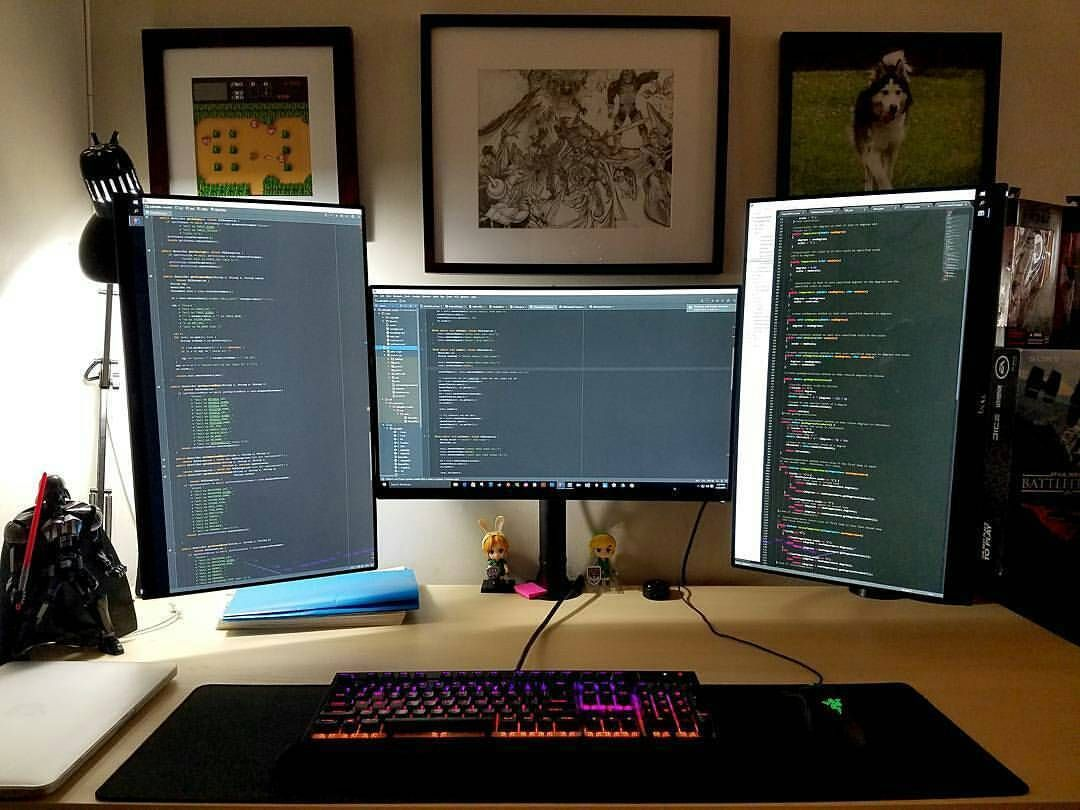
## Inleiding



In deze labwerk gaan we de basisprincipes van TypeScript leren. TypeScript is een superset van JavaScript die extra functies biedt zoals statische typen en interfaces om ontwikkelaars te helpen fouten eerder in het ontwikkelingsproces te detecteren.

Hier zijn de stappen die we zullen volgen voor deze labwerk:

## TypeScript installeren



Zorg ervoor dat je TypeScript wereldwijd installeert met behulp van npm door npm install -g typescript in je opdrachtregel uit te voeren.

Voer de volgende commando uit console:

npm install -g typescript  
Copy

## Maak een TypeScript-bestand

Maak een nieuw TypeScript-bestand, bijvoorbeeld index.ts.

mkdir -p JavaScriptProjects  
cd JavaScriptProjects  
mkdir -p LabwerkBasisJS  
cd LabwerkBasisJS  
New-Item -Name "index.ts"  
Copy

## Compileer het TypeScript-bestand

Compileer het TypeScript-bestand met behulp van het tsc-commando in je opdrachtregel: tsc index.ts. Dit zou een JavaScript-bestand met dezelfde naam als het TypeScript-bestand moeten genereren.

tsc index.ts  
Copy

## Typen aan variabelen toevoegen

Declareer een variabele met een type in TypeScript, bijvoorbeeld let myString: string = "Hallo, wereld!";.

let myString: string = "Hallo, wereld!";   
let myNumber: number = 42;   
let myBoolean: boolean = true;  
Copy

## Maak een interface

Maak een interface voor een persoon-object met eigenschappen zoals naam, leeftijd en e-mail.

interface Person {   
 name: string;   
 age: number;   
 email: string;   
}  
Copy

## Maak een functie met parameters en retourtypen

Maak een functie die twee parameters aanneemt en een string retourneert. De parameters moeten typen hebben en het retourtype moet ook worden gespecificeerd.

function addStrings(str1: string, str2: string): string {  
 return str1 + str2;   
}  
Copy

## Gebruik de interface

Maak een object met behulp van de interface die je hebt gemaakt in stap 5. Gebruik het object om de eigenschappen aan te roepen en weer te geven.

let person: Person = {   
 name: "John Doe",   
 age: 30,   
 email: "john.doe@example.com"   
};   
  
console.log(person.name);   
console.log(person.age);   
console.log(person.email);  
Copy

## Gebruik voorwaardelijke verklaringen

Maak een voorwaardelijke verklaring die controleert of een variabele van een bepaald type is. Bijvoorbeeld, als een variabele een string is, geef dan een bericht weer waarin staat "De variabele is een string".

let myVariable: any = "Dit is een string.";   
  
if (typeof myVariable === "string") {   
 console.log("De variabele is een string.");   
}  
Copy

## Gebruik loops

Maak een loop die door een array van getallen gaat en elk getal weergeeft.

let myArray: number[] = [1, 2, 3, 4, 5];   
  
for (let i = 0; i < myArray.length; i++) {   
 console.log(myArray[i]);   
}  
Copy

## Gebruik klassen

Maak een klasse met eigenschappen en methoden. Gebruik de klasse om een ​​object te maken en roep de methoden aan.

class Car {   
 private make: string;   
 private model: string;   
 private year: number;  
  
 constructor(make: string, model: string, year: number) {   
 this.make = make;   
 this.model = model;   
 this.year = year;   
 }  
  
 public getMake(): string { return this.make; }  
  
 public getModel(): string { return this.model; }  
  
 public getYear(): number { return this.year; }   
  
}  
  
let myCar: Car = new Car("Toyota", "Corolla", 2022);  
  
console.log(myCar.getMake());   
console.log(myCar.getModel());   
console.log(myCar.getYear());  
Copy



Onthoud dat je de tijd moet nemen en vragen moet stellen als je vastloopt. Veel plezier en succes met coderen!