TypeScript

STRIKT JAVASCRIPT

Vad är TypeScript

TypeScript är JavaScript fast lite mer strikt.

I JavaScript har man väldigt mycket frihet och mycket ansvar över vad man skriver.

JavaScript är ett ”tolkande” språk, den kontrolleras medan den körs.

TypeScript är ett kompilerad språk, så fel upptäcks redan innan koden körs.

Man kan skriva JavaScript och hantera det som TypeScript utan problem.

TypeScript omvandlas till JavaScript när det körs.

Filer

JavaScript filer sparas med ändelsen .js

TypeScript sparas som .ts

Man kan ändra en js fil att bli ts, det kommer att fungera lika fint.

Installation

I terminalen, skriv

```
Set-ExecutionPolicy -Scope Process -ExecutionPolicy Bypass
```

Och sedan skriver du

```
npm install -g typescript
```

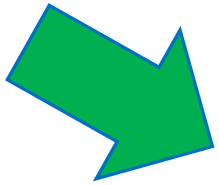
Slutligen kan du testa med att skriva

```
tsc --version
```

```
tsc --help
```

Ts omvandlas till Js

```typescript
var name:string = "John";
var score1:number = 50;
var score2:number = 42.50
var sum = score1 + score2
console.log("name"+name)
console.log("first score: "+score1)
console.log("second score: "+score2)
console.log("sum of the scores: "+sum)
```



```javascript
//Generated by typescript 1.8.10
var name = "John";
var score1 = 50;
var score2 = 42.50;
var sum = score1 + score2;
console.log("name" + name);
console.log("first score: " + score1);
console.log("second score : " + score2);
console.log("sum of the scores: " + sum);
```

Källa: https://www.tutorialspoint.com/typescript/typescript_variables.htm

var vs let

JavaScript

```
var name = "James Sunderland"
var name = "Mary Shepherd"
console.log(`The name is ${name}`)
```

```
let name = "James Sunderland"
let name = "Mary Shepherd"
console.log(`The name is ${name}`)
```

TypeScript

```
var name:string="James Sunderland";
var name:string="Mary Shepherd";
console.log(name);
```

Typsäker

```
var num          = 50;
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

Config fil

I TypeScript kan man påverka hur kompilatorn ska bete sig genom att skapa en fil

tsconfig.json

```
{
    "compilerOptions": {
        "target": "es5",
        "module": "es2020",
        "outdir": "js",
    }
}
```

Configfil

Target = vilken ECMAScript den ska använda,
es3, es5, es6, es2015, es2016, es2017,es2018, esnext

Outdir = var JavaScript filerna ska hamna efter att de TypeScript kompilerats.

Configfil - module

CommonJS

```
"use strict";
Object.defineProperty(exports, "__esModule", { value: true });
exports.twoPi = void 0;
const constants_1 = require("./constants");
exports.twoPi = constants_1.valueOfPi * 2;
```

ES2020

```
import { valueOfPi } from "./constants";
export const twoPi = valueOfPi * 2;
```

Config fil

Så en rekommenderad tsconfig.json vore...

```
{
    "compilerOptions": {
        "target": "es5",
        "module": "es2020",
        "outdir": "js",
    }
}
```

Klasser

```
class Heather
{
    name: string;
    age: number;
    era: string;
    quote:string;
    constructor()
    {
        this.name="Alessa Guillespie";
        this.age=5;
        this.era="Silent Hill 1";
        this.quote="Daddy, help me! Daddy, where are you?";
    }

    public Hello()
    {
        console.log(`Hello, my name is ${this.name} and I am ${this.age} years old. I live in the era of ${this.era}`);
        console.log(`- ${this.quote}`);
        console.log("");
    }
}
```

```
console.clear();
var heather = new Heather();
heather.Hello();

heather.name="Heather Morris";
heather.age=18;
heather.era="Silent Hill 3";
heather.quote="My father is dead! He's murdered! Get out! This is all your fault!";
heather.Hello();

heather.name="Cheryl Mason";
heather.age=25;
heather.era="Silent Hill: Shattered Memories";
heather.quote="Why did you have to die? It wasn't my fault. Someone has to take the blame.";
heather.Hello();
```

Enkel klass

```
class Point {
    x: number;
    y: number;

   public toString() {return `(${this.x}, ${this.y})`;}
}

var p = new Point();
p.x = 10;
p.y = 12;
console.log(p.toString());
```

Constructor

```
class Point {
    x: number;
    y: number;

    constructor(x = 0, y = 0) {
        this.x = x;
        this.y = y;
    }


    public toString() {return `(${this.x}, ${this.y})`;}
}


var p = new Point(10,12);
console.log(p.toString());
```

För mer info…

Mer läsning…

https://www.w3schools.blog/typescript-tutorial

https://www.youtube.com/watch?v=NjN00cM18Z4&ab_channel=ProgrammingwithMosh

Alternativt…

https://www.javatpoint.com/typescript-tutorial

https://www.youtube.com/watch?v=BCg4U1FzODs&ab_channel=TraversyMedia