

CROSS-PLATFORM DEVELOPMENT MODUL 4 – TYPESCRIPT

© Mahmud Al Hakim
www.alhakim.se
mahmud@alhakim.se

VAD ÄR TYPESCRIPT?

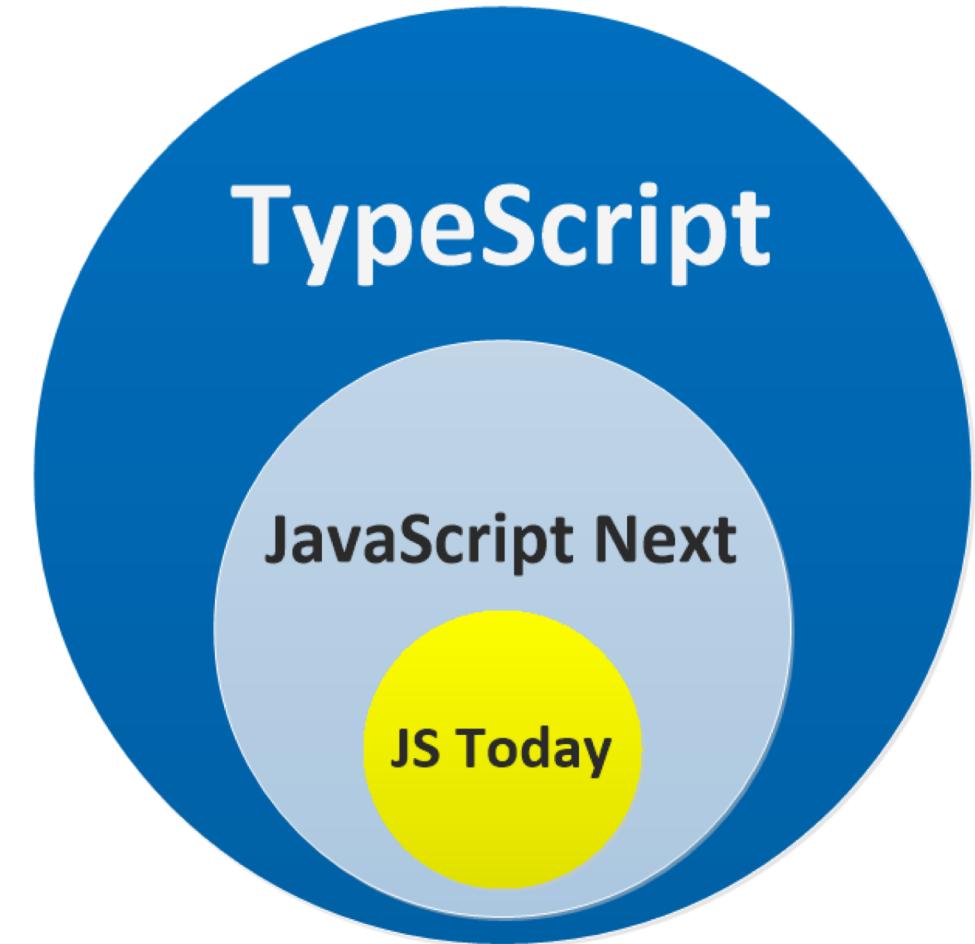
TypeScript är en superset av JavaScript.

TypeScriptkod kompileras till standard
JavaScript.

TypeScript är Open Source och utvecklas
av Microsoft.

Officiell hemsida:

<https://www.typescriptlang.org>



Bildkälla: <https://basarat.gitbooks.io/typescript/content/docs/javascript/recap.html>

TYPESCRIPT PLAYGROUND

[HTTPS://WWW.TYPESCRIPTLANG.ORG/PLAY](https://www.typescriptlang.org/play)

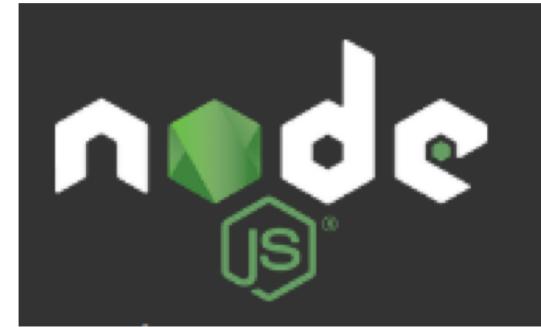
The screenshot shows the TypeScript Playground interface. At the top, there is a toolbar with a dropdown menu labeled "Select...", tabs for "TypeScript" (which is selected), "Share", "Run", and "JavaScript". Below the toolbar is a code editor containing the following TypeScript code:

```
1 var a = 1
2 let b = 2;
3 const c = 3;
4 c = 4;
```

The "c = 4;" line is highlighted with a red underline, indicating a syntax error. To the right of the code editor is an output pane showing the corresponding JavaScript code that would be generated if the "Run" button were clicked:

```
1 var a = 1;
2 var b = 2;
3 var c = 3;
4 c = 4;
5
```

INSTALLERA TYPESCRIPT VIA NPM (NODE PACKAGE MANAGER)



1. Ladda ner och installera NodeJS <https://nodejs.org>
2. Starta kommandotolken/terminalen och skriv detta kommando för att installera TypeScript globalt (tillgängligt i alla projekt)

(PC) `npm install -g typescript`

(Mac) `sudo npm install -g typescript`

SKAPA EN TS-FIL



EXPLORER

▲ OPEN EDITORS

TS demo.ts

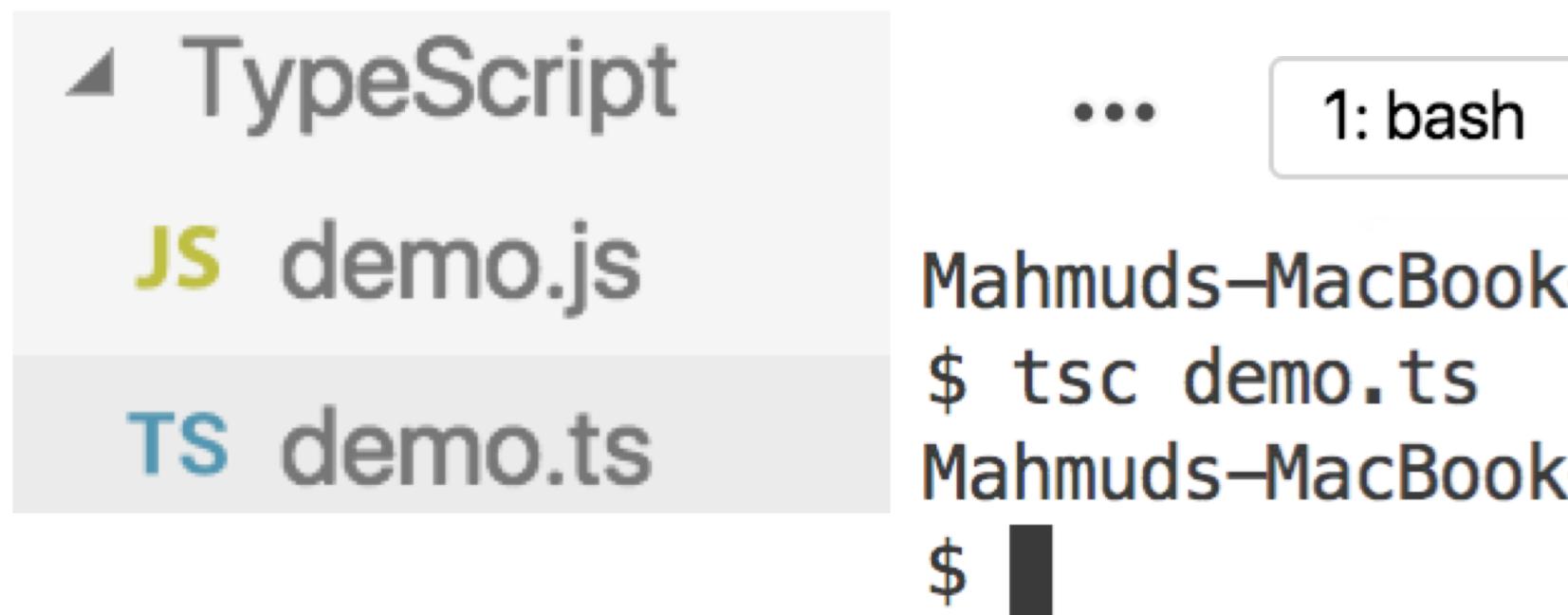
▲ TYPESCRIPT

TS demo.ts

TS demo.ts ×

```
1 var a = 1;
2 let b = 2;
3 const c = 3;
```

1. KOMPILERA TYPESCRIPT MED KOMMANDOT TSC
2. EN NY JS-FIL SKAPAS

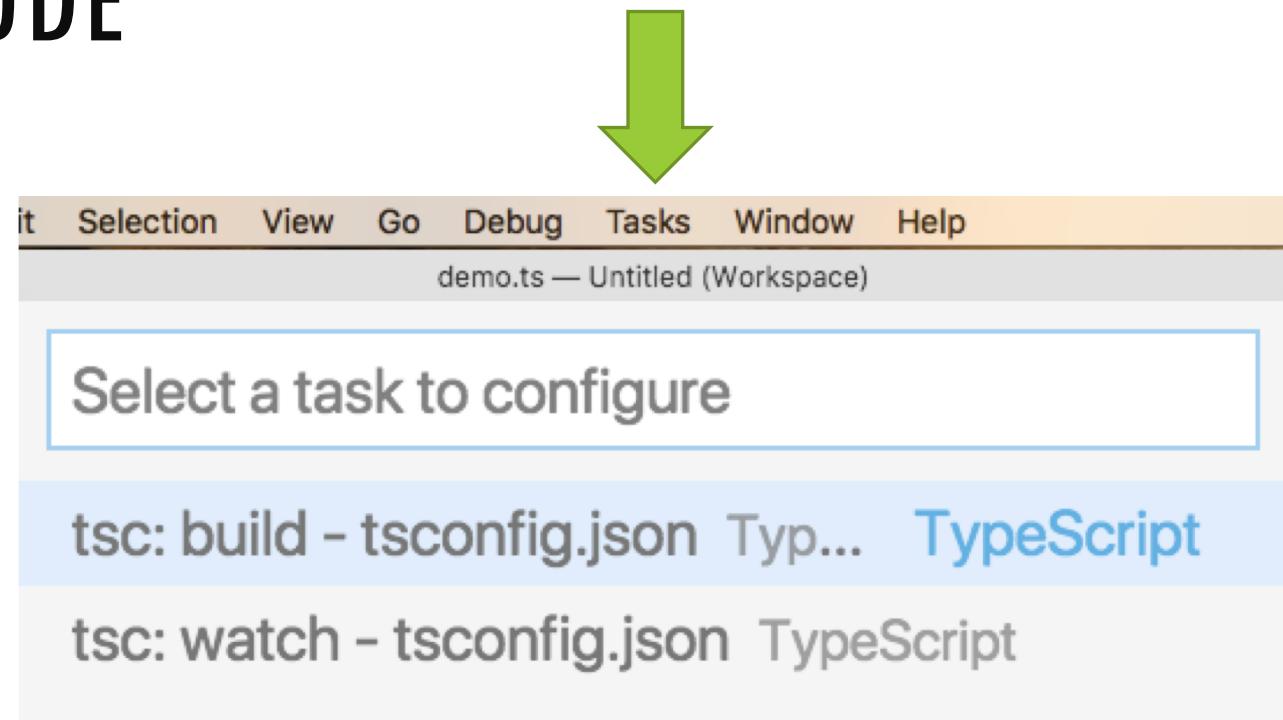


SKAPA EN INSTÄLLNINGSFIL TSCONFIG.JSON

```
{  
  "compilerOptions": {  
    "target": "es5",  
    "watch": true  
  }  
}
```

AUTOKOMPILEERA I VS CODE

1. Öppna "Configure Task..." under menyn Tasks.
2. Välj "tsc:build – tsconfig.json"
3. Filen tasks.json skapas.
Kör denna fil genom att gå in under menyn Tasks igen.
Välj "Run task..."



BASIC TYPES

```
// Boolean  
let done: boolean = false;  
  
// Number  
let userId: number = 6;  
  
// String  
let color: string = "blue";  
  
// any  
let notSure: any = 4;
```

ARRAYS

```
let list1: number[] = [1, 2, 3];
let list2: string[] = ["Ett", "Två", "Tre"];
```

ENUM

```
enum Color {Red, Green, Blue}  
let c: Color = Color.Blue;
```

TEMPLATE STRINGS

```
let fullName: string = "Mahmud Al Hakim";
let age: number = 45;
let sentence: string = `Hello, my name is ${ fullName }.
I'll be ${ age + 1 } years old next year.`;
```

FUNCTION TYPES

```
function add(x: number, y: number): number {  
    return x + y;  
}  
console.log(add(1,2));
```

ARROW FUNCTION

```
let add = (x: number, y: number) => x + y;
```

Argumenten till vänster



Retun till höger

TypeScript

```
var add = function (x, y) { return x + y; };
```

JavaScript

CLASSES

```
class Person {  
    firstName: string;  
    lastName : string;  
  
    constructor(firstName: string, lastName: string) {  
        this.firstName = firstName;  
        this.lastName = lastName;  
    }  
    fullname() {  
        return this.firstName + " " + this.lastName;  
    }  
}
```

OBJECTS

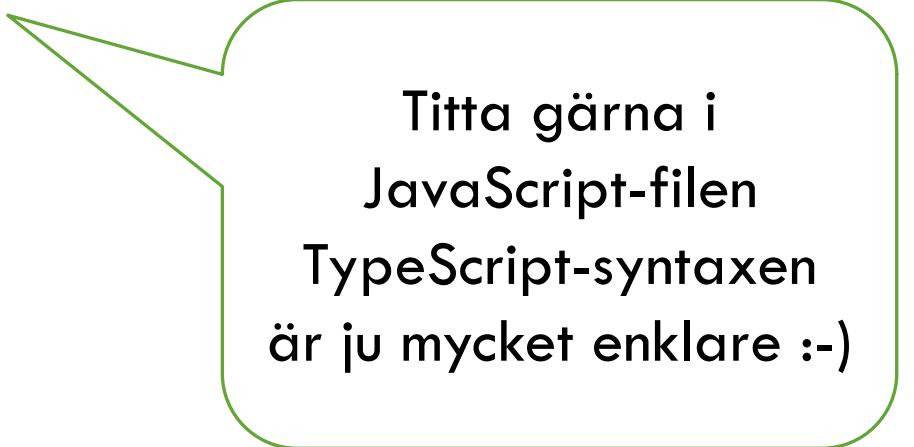
```
window.onload = () => {
    var p1 = new Person("Mahmud", "Al Hakim");
    var el = document.getElementById('content');
    el.innerHTML = p1.fullname();
};
```

SKAPA INSTANSVARIABLER DIREKT I KONSTRUKTORN

```
class Person {  
    // firstName:string;  
    // lastName : string;  
    constructor(public firstName: string, public lastName: string) {  
        // this.firstName = firstName;  
        // this.lastName = lastName;  
    }  
    fullname() {  
        return this.firstName + " " + this.lastName;  
    }  
}
```

INKAPSLING (ENCAPSULATION)

```
class Person {  
    constructor(private fName: string, private lName: string){  
    }  
    get firstName() {  
        return this.fName;  
    }  
    set firstName(value: string) {  
        this.fName = value;  
    }  
}
```



Titta gärna i
JavaScript-filen
TypeScript-syntaxen
är ju mycket enklare :-)

STATISKA VARIABLER

```
class Person {  
    public static antalObjekt : number = 0;  
    constructor(private fName: string, private lName: string) {  
        Person.antalObjekt++;  
    }  
}  
window.onload = () => {  
    var p1 = new Person("Mahmud", "Al Hakim");  
    var p2 = new Person("Kalle", "Anka");  
    var el = document.getElementById('content');  
    el.innerHTML = Person.antalObjekt.toString();  
};
```

ARV (INHERITANCE)

```
class Person {  
    constructor(public name: string) {}  
}  
  
class Employee extends Person {  
    constructor(public name: string, private department: string) {  
        super(name);  
    }  
    public getInfo() {  
        return this.name + " arbetar på " + this.department;  
    }  
}
```

INTERFACE

```
interface Figur {  
    area: () => number;  
    // area är en metod som returnerar number  
}
```

INTERFACE – FORTSÄTTNING

```
class Rektangel implements Figur {  
    constructor(public b: number, public h: number) { }  
    area() {  
        return this.b * this.h;  
    }  
}  
  
window.onload = () => {  
    var rek = new Rektangel(10,5);  
    var el = document.getElementById('content');  
    el.innerHTML = rek.area().toString();  
};
```

ÖVNINGAR

- Definiera en klass som beskriver en Punkt (vanlig 2D-punkt)
- Lägg till en statisk variabel som beskriver origo (0,0)
- Skapa två punkter (två instanser av klassen punkt) t.ex.

```
var p1 = new Point(1,2);
```

- Skapa en metod i klassen Punkt som adderar två punkter

Anrop av metoden kan se ut så här:

```
var p3 = p1.add(p2);
```

- Skapa en statisk metod i klassen Punkt som adderar två punkter.

Anrop av metoden kan se ut så här:

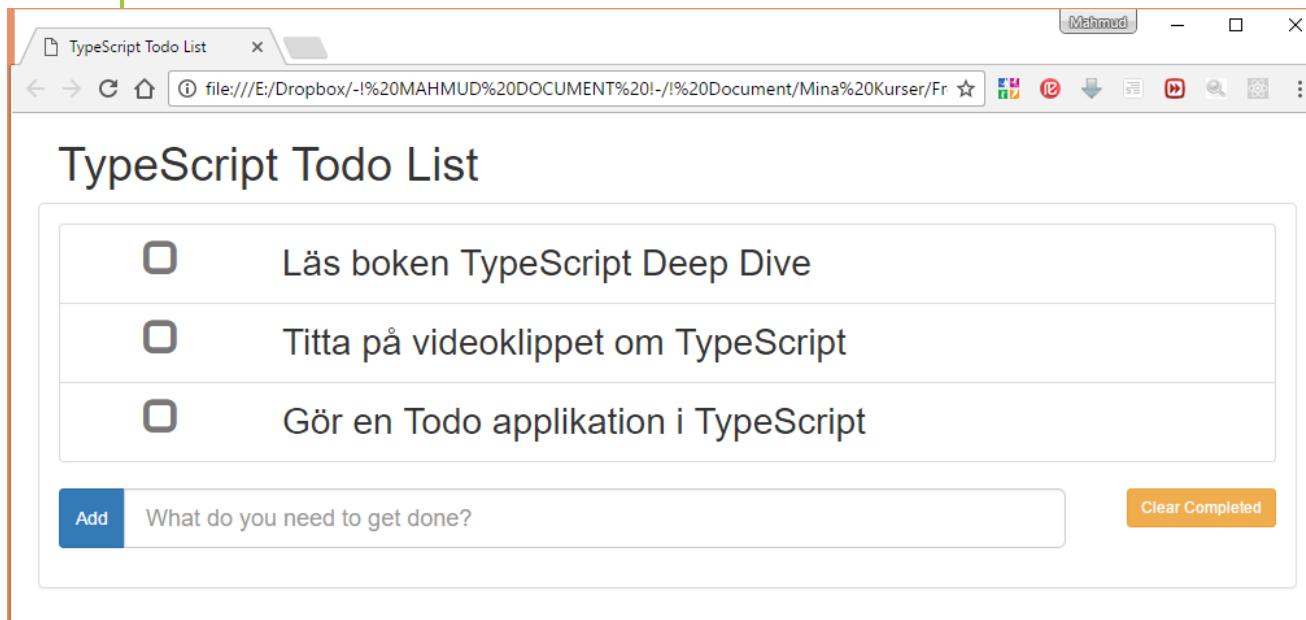
```
var p4 = Point.add(p1,p2);
```

```
class Point {  
    static origin = new Point(0, 0); //origo är alltid samma  
    static add(p1: Point, p2: Point) {  
        return p1.add(p2);  
    }  
    constructor(private xValue: number, private yValue: number) {}  
    get x() { return this.xValue; }  
    get y() { return this.yValue; }  
    add(point: Point) {  
        return new Point(point.x + this.x, point.y + this.y);  
    }  
    toString() { return `(${this.x},${this.y})`; }  
}
```

```
var p1 = new Point(2, 1);
var p2 = new Point(5, 2);
var p3 = p1.add(p2);
var p4 = Point.add(p1, p2);

window.onload = () => {
    var el = document.getElementById('content');
    el.innerHTML = `(${p3.x},${p3.y})<br>`;
    el.innerHTML += p4;
};
```

ÖVNING – SKAPA EN TODO-APPLIKATION



A screenshot of a GitHub repository page for "TypeScriptIntro/TodoService". The code editor shows a file named "TodosService.js" with the following content:

```
1 TodosService.___id = 0;
2
3 function TodosService() {
4
5     this.todos = [
6         { id: TodosService.___id++, name: 'clean cave', completed: false },
7         { id: TodosService.___id++, name: 'Dryclean cape', completed: true },
8         { id: TodosService.___id++, name: 'Save gotham', completed: false },
9     ];
10 }
11
12 TodosService.prototype.add = function add(todo) {
13
14     // Expect/accept 'todo' parameter as either a string...
15     var newTodo = { name: todo };
16
17     // or a Todo object
18     if(typeof todo === 'object') {
19         newTodo = todo;
20     }
21
22     newTodo.id = TodosService.___id++;
23     newTodo.completed = false;
24
25     this.todos.push(newTodo);
26
27 }
28
29 TodosService.prototype.clearCompleted = function clearCompleted() {
30     var completed = this.todos.filter(function (x) { return x.completed; });
31
32     completed.forEach(function (x) { _this.remove(x); });
33
34 }
35
36 var _this = this;
37 // Save a reference to "this" for use in the closure
```

The video player at the bottom of the GitHub page shows a video titled "Introduction to TypeScript" by Jess Chadwick, currently at 31:45 of 1:40:15.

Tips. Titta på denna video "Introduction to TypeScript"
av Jess Chadwick (rekommenderas varmt)
<https://www.youtube.com/watch?v=qRD7bkK7m10>

TIPS: ANDERS HEJLSBERG: INTRODUCING TYPESCRIPT

[HTTPS://WWW.YOUTUBE.COM/WATCH?V=EX2PXJJ-KDK](https://www.youtube.com/watch?v=EX2PXJJ-KDK)



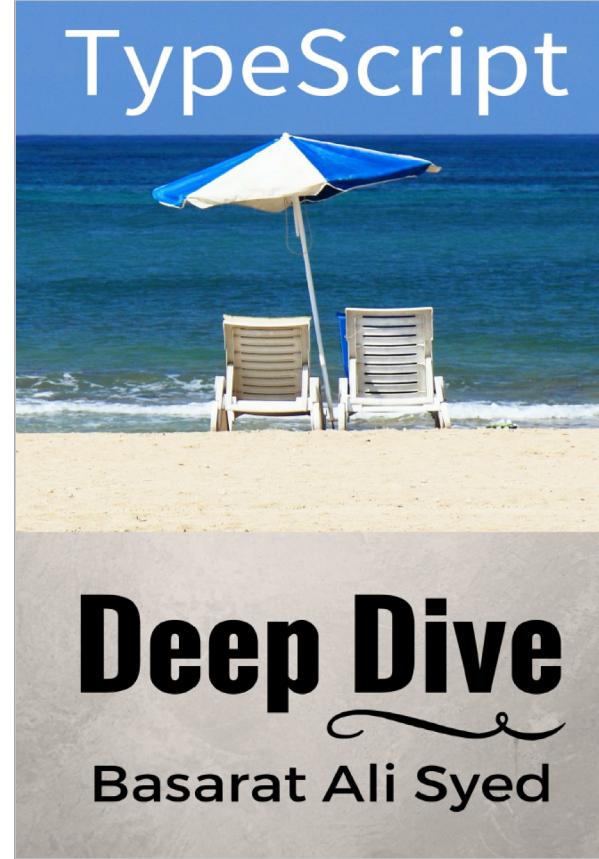
TIPS PÅ BRA BÖCKER

**Essential
TypeScript**



<https://leanpub.com/essentialtypescript/read>

TypeScript



<https://www.gitbook.com/book/basarat/typescript/details>