

Typescript

A JAVASCRIPT SUPERSET



מה נלמד?

- מה זה Typescript?

- Types

- Union Types

- Casting

- Arrays

- Functions

Typescript

- שפת תכנות המהווה מעטפת לשפת javascript המוסיפה עליה הרחבות לתכנות מונחה עצמים
- כל קוד javascript הוא קוד typescript תקין, אך לא להיפך
- קבצים בסיומת .ts
- מפותחת ע"י Microsoft החל משנת 2012
- כל קובץ ts עובר הידור (קומפילציה) ל-javascript על מנת שהדפדפן יוכל להריץ אותו
- פיצ'רים מרכזיים: הגדרת סוג הנתונים, ממשקים, enums



Typescript

למה שנשתמש?

- ב-Javascript מעולם לא הגדרנו את סוג הנתונים איתו אנחנו עובדים, רק שם משתנה וערך – מה שיכול לגרום לשרשרת של שגיאות שלא נהיה מודעים עליה עד לשלבים מתקדמים בפיתוח
- Typescript מאפשרת לגלות שגיאות בסוג הנתונים בזמן כתיבת הקוד ולא בזמן ריצת הקוד ע"י הצהרה מראש על סוג הנתונים עבור משתנה מסוים
- Typescript מרחיבה את הפונקציונליות של Javascript ע"י הוספה של מבני נתונים כמו enum, tuple, interface



Typescript

הורדת nodejs כדי להשתמש ב-npm

▪ <https://nodejs.org/en/download/>

התקנה בטרמינל

▪ `npm i -g typescript`

`tsc --init` - יצירת קובץ קונפיגורציה (`tsconfig.json`) עבור תהליך ההידור

`tsc filename.ts` - יצירת קובץ `filename.js` לאחר תרגום

*ניתן להוסיף `--watch` - לקימפול בכל שמירה

<> index.html

JS test.js

TS test.ts

TS tsconfig.json

TS test.ts > ...

```
1 let myname: string = "Mor Ashkenazi";  
2 console.log(myname);
```

Mor Ashkenazi

Types

- ניתן להגדיר משתנים מסוגים שונים ע"י הצהרה על סוג הנתונים

```
1 let myname: string = "Mor Ashkenazi";
2 let myage : number = 18;
3 let isLecturer : boolean = true;
4 let something : any = "6"
5
6 myname = 11;
7 isLecturer = "false"
8 something = 6
```

let myname: string

Type 'number' is not assignable to type 'string'. ts(2322)

[View Problem \(Alt+F8\)](#) No quick fixes available

let isLecturer: boolean

Type 'string' is not assignable to type 'boolean'. ts(2322)

[View Problem \(Alt+F8\)](#) No quick fixes available

Union Types

- ניתן להגדיר שמשתנה מסוים יכול להיות מסוגי נתונים שונים

```
let isActive : boolean | string = false  
isActive = "true"
```



```
type booleanOrString = string | boolean  
  
let isActive : booleanOrString = false  
isActive = "true"
```

- ניתן להגדיר זאת גם ע"י הצהרה על type חדש

Casting

- המרת נתון מסוג אחד לאחר ע"י המילה השמורה as
- ההמרה תמיד תהיה מהכלל אל הפרט

סוג נתונים כמו any
רק שאי אפשר
להשים (השמה) אותו
לסוג ספציפי מלבדו

```
let id: unknown = "3256";  
console.log(id.length);
```

```
let id: unknown = "3256";  
console.log((id as string).length);
```


Array

- ניתן להגדיר מערך וסוג נתוניו

```
let names : string[] = ["Yoav", "Guy", "Shira"];  
let prices : number[] = [250, 145, 50];  
let departments: Array<string> = ["HR", "IT", "Software"];  
  
names.push("Aviv")  
prices.push("5");
```

Argument of type 'string' is not assignable to parameter of type 'number'. ts(2345)


[View Problem \(Alt+F8\)](#) No quick fixes available

Functions

- ניתן להגדיר פונקציות עם סוגי נתונים עבור הפרמטרים וערך מוחזר

```
function lengthBiggerThan4(str: string): boolean {  
    return str.length > 4;  
}  
console.log(lengthBiggerThan4("Hello"));  
console.log(lengthBiggerThan4("You"));
```

```
true  
false
```



```
function print(message: string): void {  
    console.log(message);  
}
```

- אם פונקציה לא מחזירה כלום נגדיר

Functions

- ניתן להגדיר ערכי ברירת מחדל לפרמטרים

```
function printMe(message: string = "Welcome"): void {  
  console.log(message);  
}  
  
printMe();  
printMe("Good Morning");
```

```
Welcome  
Good Morning
```

- ניתן להגדיר את סוג הפונקציה בנפרד

```
type MathFunc = (value: number) => number;  
  
let doubleX: MathFunc = (x) => x * 2;  
  
console.log(doubleX(6));
```

```
12
```