

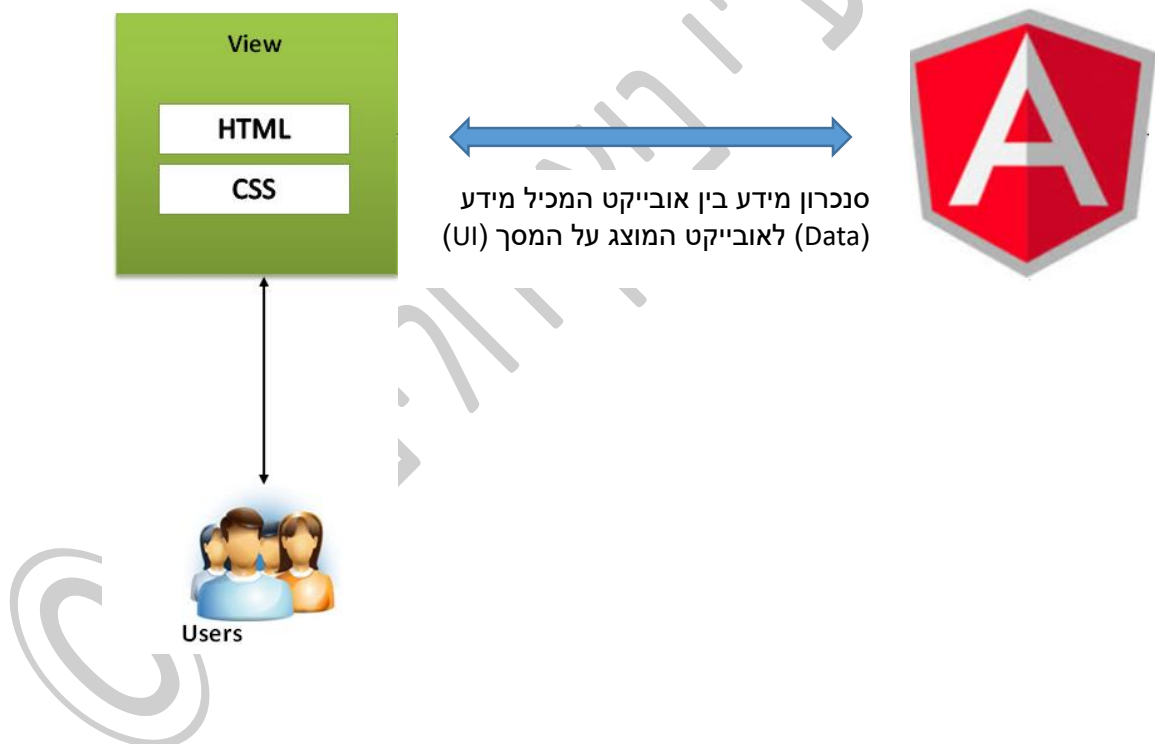
# Angular

## מצגת 1

- א. מבוא
- ב. התקנות
- ג. יצירת פרוייקט
- ד. הרצת תכנית
- ה. מבנה קומפוננטה ראשית
- ו. יצירת ועריכת קומפוננטות
- ז. הגדרות רכיב Component
- ח. הגדרות רכיב NgModule

## א. מבוא

- אנגולר הוא framework (בסיס לפיתוח אתר אינטרנט שנותן את כלים למתכנת לעבודה יעילה) מבוסס Typescript ליצירת User Interface דינאמי.
- אנגולר מאפשר מודולריות - הפרדת הפונקציונליות של תוכנה למודולים עצמאיים.
- באמצעות שימוש חוזר במודולים, ניתן להיפטר מכתובת אותו קוד שוב ושוב.
- אנגולר מאפשר בניית SPAs (single page applications).



## ב. התקנות

### Node.js התקנת

- Node.js היא פלטפורמת פיתוח קוד פתוח המבוססת על JavaScript.
1. להוריד לינק ולהתקין:

```
https://nodejs.org/en/download
```

### CLI התקנת

- CLI מספקת למשתמש אפשרות להקיש פקודות טקסטואליות.
2. הפעל cmd
3. הרץ קוד (cmd):

```
npm install -g @angular/cli
```

## ג. יצירת פרויקט

1. ליצירת פרוייקט חדש בשם hello-world בצע פקודה (cmd):

```
ng new hello-world
```

2. בשאלה Would you like to add Angular routing? - הקש n.

3. הקש Enter (שימוש ב-CSS).

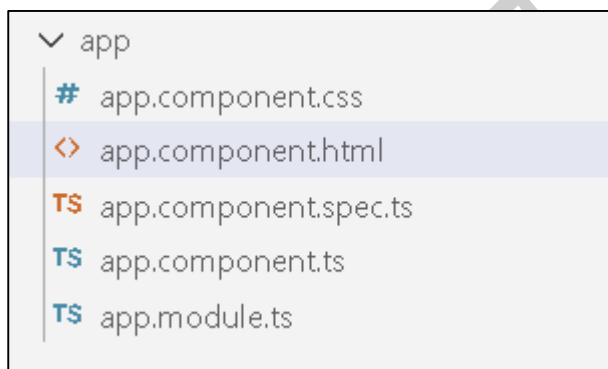
4. ב-Visual studio code יש לבחור בסרגל עליון

5. File -> add folder to workspace...

6. לבחור:

```
C:\Users\<Username>\hello-world\src\app
```

• נקבל:



## ד. הרצת תכנית

1. יש להגיע ב-cmd לתוך תיקיית הפרוייקט (cmd):

```
C:\Users\<Username>\hello-world
```

2. בצע (cmd):

```
ng serve
```

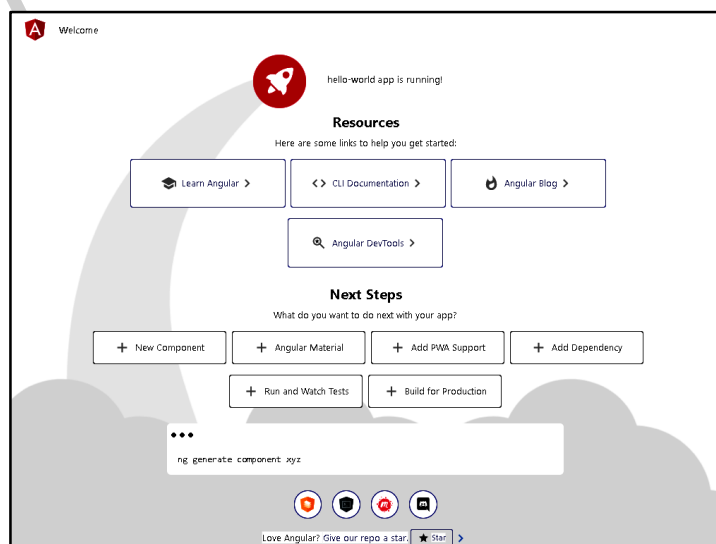
3. הקש

- הפקודה ng serve מפעילה השרת ווב לפיתוח המאפשר לצפות ולערוך את האפליקציה.

4. הפעל את chrome בכתובת זו (הכתובת השרת רץ על המחשב):

```
http://localhost:4200/
```

- נקבל דף ברירת מחדל:



## ה. מבנה קומפוננטה ראשית

- הקומפוננטה הראשית היא הרכיב הבסיסי של ממשק המשתמש באפליקציית Angular.
- מכילה הוראות ונתונים המשויכים לתבנית ה-HTML.
- ב-Visual studio code:

File -> add folder to workspace -> ...\Hello-world\src\app

### קובץ html לתבנית הקומפוננטה - app.component.html:

1. פתח קובץ תבנית (HTML):

...\Hello-world\src\app\app.component.html



2. מחק את הקוד והוסף:

<p>hello world</p>

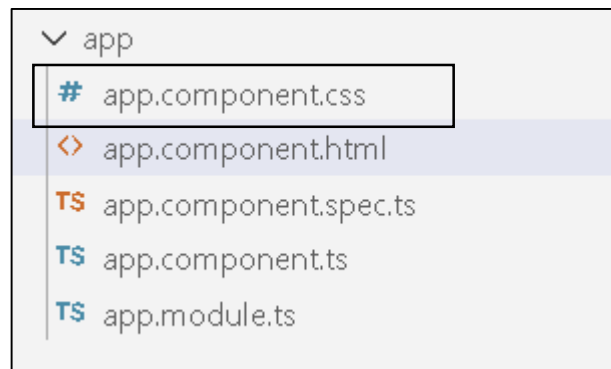
3. שמור (ביצוע קומפילציה וטעינת הדף).

4. הפעל את chrome בכדי לראות את עדכון הדף הראשי.

## קובץ הגדרות CSS של הקומפוננטה - app.component.css:

1. פתח קובץ עיצוב (CSS):

```
...\app\app.component.css
```



2. הוסף בקובץ:

```
p{
    font-size:40px;
}
```

## הוספת תמונות לקומפוננטה

• ב-VISUAL STUDIO CODE:

```
File -> add folder to workspace -> ...\\Hello-world\\src\\assets
```

1. יש למקם תמונות בתיקייה assets:

2. עדכן קובץ תבנית (HTML):

```

```

## תרגיל 1

- יש לבצע את התרגיל ממצגת CSS – ולהוסיף לאפליקציית האנגולר.

\* בהגשה יש לצרף את תיקיית app

נכתב ע"י ניצן דולינסקי ©



## 1. יצירת ועריכת קומפוננטות נוספות

- סיבות לשימוש בקומפוננטות:
  - הארכיטקטורה מבוססת רכיבים.
  - שימוש חוזר ברכיבים.
  - פיתוח מבוסס רכיבים משפר את ביצועי האפליקציה הכוללים.

1. יש ליצור פרוייקט חדש hello-world2 ולהגיע ל:

```
C:\Users\<Username>\hello-world2
```

2. יש למחוק את הקוד ב:

```
...\app\app.component.html
```

## יצירת קומפוננטה

- יצירת קומפוננטה באמצעות הפקודה:

```
ng generate component <component name>
```

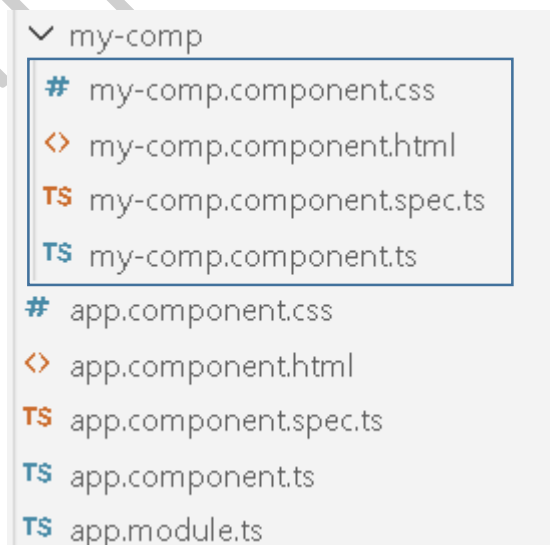
1. יש ליצור קומפוננטה חדשה ב-cmd באמצעות פקודה:

```
ng generate component my-comp
```

- ניתן לרשום גם בקיצור:

```
ng g c my-comp
```

- ניתן לראות כי נוצרת ספרייה בשם הקומפוננטה בתוך תיקיית app עם 4 קבצים.



## הצגת תוכן דף ה- html של הקומפוננטה מתוך קומפוננטה ראשית

1. הפעל אפליקציה – בצע פקודה:

```
ng serve
```

2. הפעל את הדפדפן בכתובת:

```
http://localhost:4200/
```

3. פתח את קובץ התבנית של הקומפוננטה:

```
...\Hello-world2\src\app\my-component\ my-comp.component.html
```

• התוכן אותו נרצה להציג בדף הראשי:

```
<p>my-comp works!</p>
```

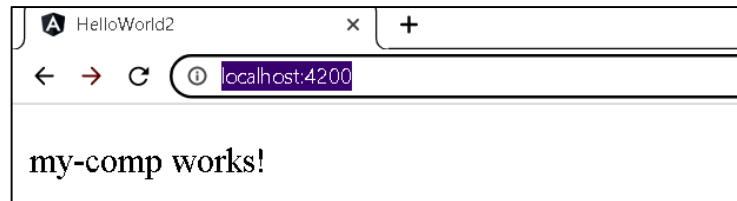
4. פתח באמצעות notepad++ את קובץ דף ה-html של הקומפוננטה הראשית של האפליקציה app.component.html במיקום:

```
...\Hello-world2\src\app\ app.component.html
```

5. הוסף את התגית הבאה לקובץ דף ה-html הראשי בכדי לקרוא לדף ה-html של הקומפוננטה:

```
<app-my-comp></app-my-comp>
```

- נקבל:



- התגית מוגדרת בקובץ ה- my-comp.component.ts במיקום:

```
...\app\my-comp\ my-comp.component.ts
```

- זהו קובץ typescript בו נמצאות הגדרות שונות של הקומפוננטה:

```
@Component({  
  selector: 'app-my-comp',  
  templateUrl: './my-comp.component.html',  
  styleUrls: ['./my-comp.component.css']  
})
```

## תרגיל 2

- צור פרויקט שיכיל 3 קומפוננטות.
- כאשר נריץ את הדפדפן יוצג:

```
Hello world from main app
Hello world from component 1
Hello world from component 2
Hello world from component 3
```

- שורה 1 תקרא מתוך דף ה-html הראשי של האפליקציה.
- שורה 2 תקרא מתוך דף ה-html של הקומפוננטה הראשונה.
- שורה 3 תקרא מתוך דף ה-html של הקומפוננטה השנייה.
- שורה 4 תקרא מתוך דף ה-html של הקומפוננטה השלישית.

\* בהגשה יש לצרף את תיקיית app

## תרגיל 3

- צור פרויקט שיכיל 4 קומפוננטות.
- יש לבצע את תרגיל 1 כאשר כל נושא ייקרא מקומפוננטה שונה.

\* בהגשה יש לצרף את תיקיית app

## ז. הגדרות רכיב Component

### מהו decorator

- decorator – מיועד לתאר לקומפיילר כיצד להדר את התכנית.
- מיוצג באמצעות @ על מודול מיובא.
- מגדיר את תכונות הרכיב באופן הבא:

```
{
  מודול: @
  תכונה 1: _____,
  תכונה 2: _____,
  ...
}
```

### הגדרת Component כרכיב Angular באמצעות decorator

- מיוצג בקובץ component.ts.
- אוסף את נתוני הקומפוננטה.
- ייבוא מודול Component מליבת אנגולר:

```
import { Component } from '@angular/core';
```

- שימוש ב-decorator:

```
@Component({
  selector: '_____',
  templateUrl: './_____',
  styleUrls: ['./_____']
})
```

- תכונות:

- selector – שם התגית
- templateUrl – מיקום דף html שתוכנו יוצג בקריאה לתגית
- styleUrls – מיקום css של דף ה-html שתוכנו יוצג בקריאה לתגית.

## ח. הגדרות רכיב NgModule

- יש ליצור 3 קומפוננטות בשם:

- my-comp1
- my-comp2
- my-comp3

### הגדרת NgModule כרכיב Angular באמצעות decorator

- מיוצג בקובץ app.module.ts.
- מחלקת **NgModule** – אוסף את נתוני הקומפוננטות הקשורות לקומפוננטה הראשית ושירותים שונים.
- ייבוא מודול NgModule מליבת אנגולר:

```
import { NgModule } from '@angular/core';
```

- ייבוא קומפוננטות:

```
import { AppComponent } from './app.component';
import { MyComp1Component } from './my-comp1/my-comp1.component';
import { MyComp2Component } from './my-comp2/my-comp2.component';
import { MyComp3Component } from './my-comp3/my-comp3.component';
```

- שימוש ב-decorator:

```
import { NgModule } from '@angular/core';
...

'מיקום קומפוננטה 1' from { מודול קומפוננטה 1 } Import
'מיקום קומפוננטה 2' from { מודול קומפוננטה 2 } Import
'מיקום קומפוננטה 3' from { מודול קומפוננטה 3 } Import
...

@NgModule({
  declarations: [ קומפוננטה 1 ,
                  קומפוננטה 2 ,
                  קומפוננטה 3
                ],
  ...
})
```

- תכונות:

- declarations – מערך הקומפוננטות ביישום