

# Introduction

---

UI-Angular



# מנהלות

---

- **שם המרצה:** מור אשכנזי
- **כתובת אימייל:** [mora@braude.ort.org.il](mailto:mora@braude.ort.org.il)
- **הקורס עוסק** בפיתוח ובניית אתרים (client side) בפלטפורמת Angular.
- **מרכיבי הציון:**
  - פרויקט: 90% מהציון הסופי
  - מעבדות/תרגילי בית: 10% מהציון הסופי
  - מעבר בחינה סופית הינו תנאי לקבלת ציון חיובי בקורס.
  - נוכחות חובה ב- 80% מההרצאות הינה תנאי להשתתפות בבחינות.

# מה נלמד היום?

---

- Introduction to Angular
- Angular Architecture & Concepts
- SPA (Single Page Application)
- Installing and Setting Up an Angular Development Environment
- How to Create a new project in Angular
- Bootstrapping in Angular



# למה Angular?

---

- אחת הפלטפורמות הפופולריות והמתקדמות כיום לבניית יישומי web בצד לקוח
- Angular היא framework של Typescript שמפותחת ע"י חברת Google
- הגרסה הראשונה של Angular שיצאה ב-2010 נקראת angular.js, כל הגרסאות שבאו אחריה נקראות angular



# למה Angular?

Angular Version	Date	Description
Angular 2	14.09.2016	Initial Version of Angular
Angular 4	23.03.2017	Version 4
Angular 5	11.11.2017	Version 5
Angular 6	03-05-2018	Version 6
Angular 7	18-10-2018	Version 7
Angular 8	25-08-2019	Version 8
Angular 9	06-02-2020	Version 9
Angular 10	24-06-2020	Version 10
Angular 10.0.12	24-08-2020	Version 10.0.12

# למה Angular?

---

- כתובה ב-TypeScript (הרחבה של JavaScript)
  - מקמפלים פעם אחת, ולאחר מכן כל שמירת קובץ היא קימפול אוטומטי.
- מיועדת לפיתוח אפליקציות מבוססות web, mobile, desktop
- משמשת ליצירת אפליקציות מבוססות דף יחיד, באנגלית SPA (Single Page Application)



# לינקים חשובים

---

- Angular docs:
  - <https://angular.io/docs>
- Typescript:
  - <https://www.typescriptlang.org/>

# SPA (Single Page Application)

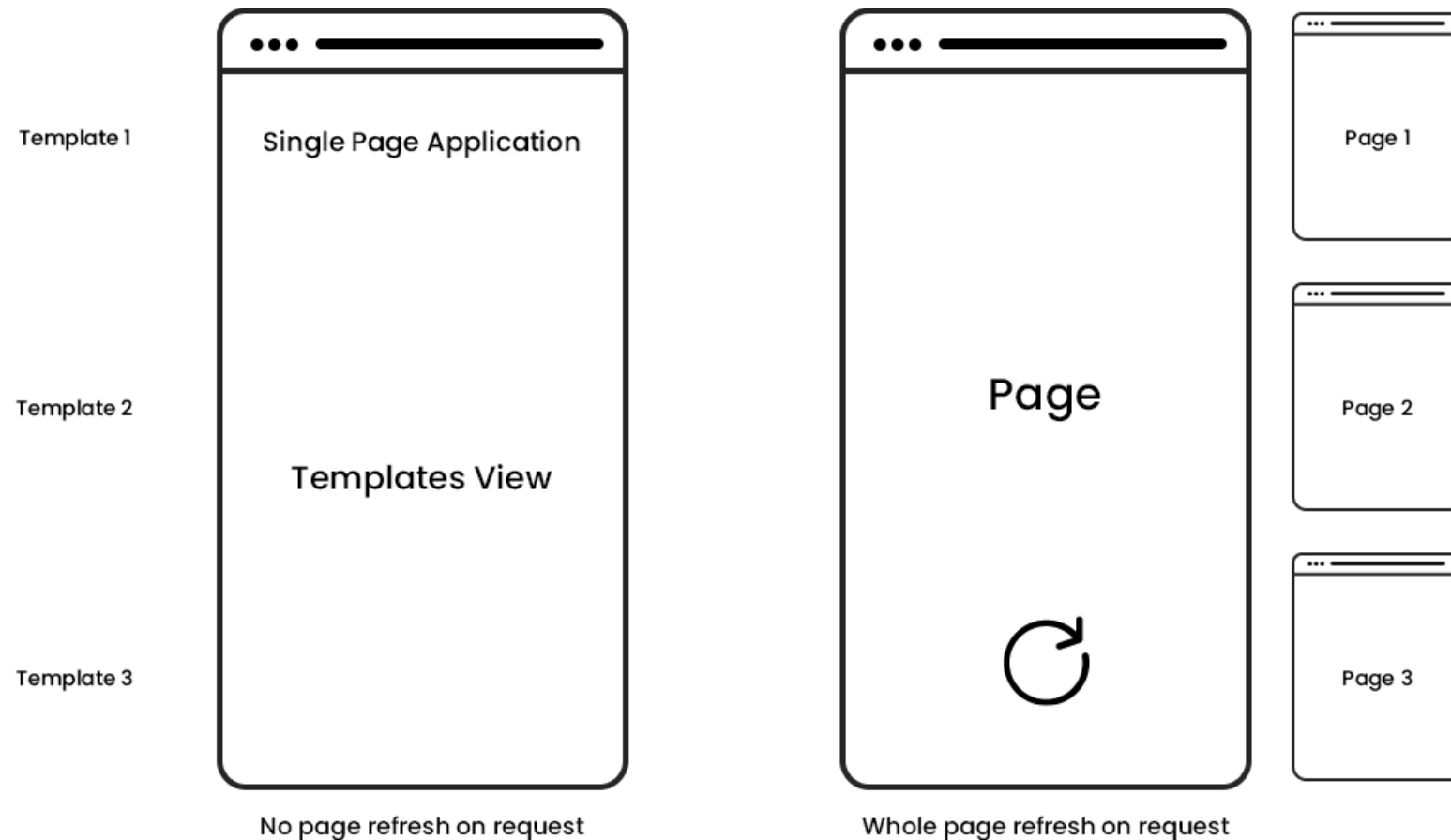
---

- טעינת קוד יחידה  
כל הקוד הדרוש (JavaScript, CSS, HTML) מגיע אל הדפדפן בטעינת דף אחת
- טעינה דינאמית  
משאבים נוספים נטענים בצורה דינמית
- תגובתיות  
הטעינה הדינאמית מתבצעת בעת פעולת משתמש



# SPA (Single Page Application)

---

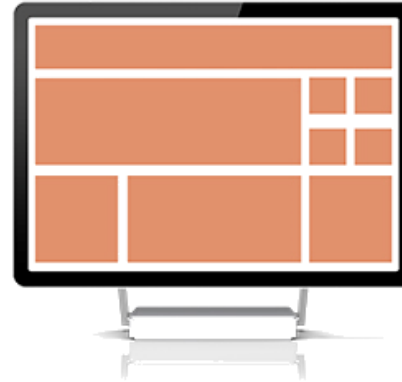


# SPA (Single Page Application)

---

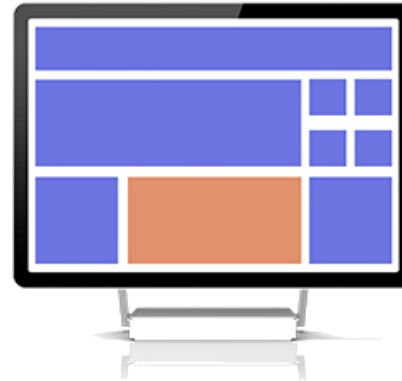
## Traditional

Every request for new information gives you a new version of the whole page.



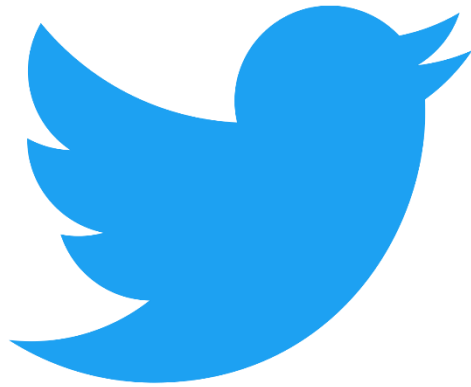
## Single Page Application

You request just the pieces you need.



# SPA (Single Page Application)

---



# SPA (Single Page Application)

## יתרונות

- מעניק חווית משתמש טובה יותר והאתר מרגיש למשתמש "מהיר"

- קל לתחזוקה מצד המתכנת, ניתן לבצע פיצול קוד

- פחות מידע מגיע מהשרת ← פחות עבודת שרת

- פחות לחכות למידע, תקורת תקשורת מופחתת



# הכנת סביבת העבודה

סביבת העבודה כוללת:

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows
(c) 2020 Microsoft Corporation. All rights reserved.
C:\Users\morac>
```

1. כלי שורת פקודות cmd (windows) או טרמינל (linux/mac)
  - קיים במערכת ההפעלה



Visual Studio Code

2. IDE (עורך קוד) – Visual Studio Code
  - הורדה: <https://code.visualstudio.com/>

# הכנת סביבת העבודה



סביבת העבודה כוללת:

## 3. התקנת node.js

- הורדה: <https://nodejs.org/en/download>
- כשמתקינים את node.js מקבלים את מתקין החבילות (node package manager) **npm** שבו ניעזר כדי להתקין את Angular ואת יתר החבילות שבהם תלוי הפרויקט שלנו.

## 4. התקנת Angular CLI (Command Line Interface)

- הקלדה בשורת הפקודה:

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

- מאפשר יצירה, הרצה, פיתוח ותחזוקת אפליקציות Angular ישירות משורת הפקודה.

# יצירה והרצה של אפליקציה

---

```
> ng new app-name  
> cd app-name  
> ng serve
```

- יצירת אפליקציית Angular
- כניסה לתיקיית האפליקציה
- הרצת השרת וקימפול מ-ts ל-js

- את האפליקציה ניתן לראות בדפדפן בכתובת:

<http://localhost:4200/>

- TCP-4200 הוא פורט התקשורת ברירת המחדל שאפליקציית Angular מאזינה לו.

# יצירה והרצה של אפליקציה

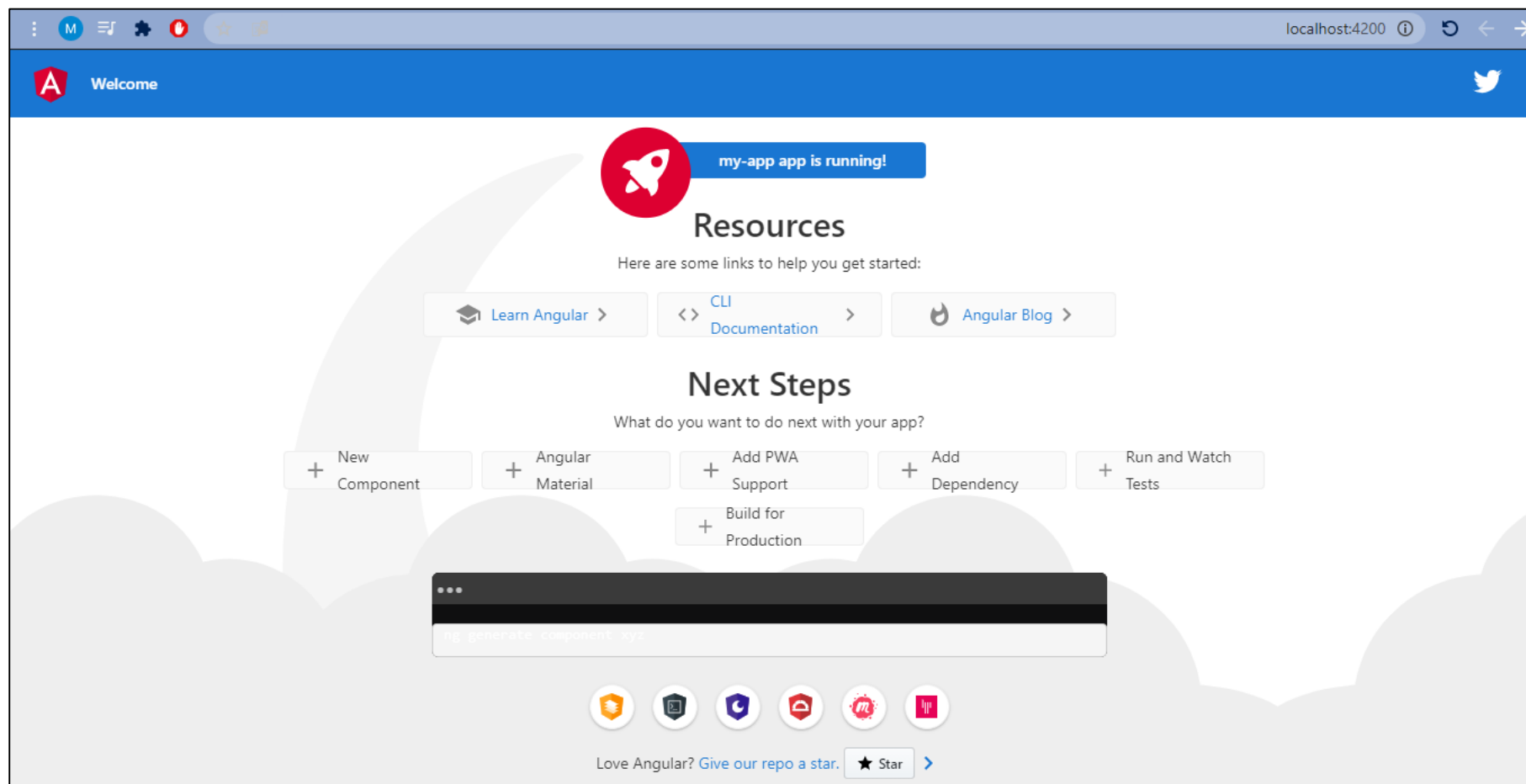
---

דוגמת יצירת אפליקציה משורת הפקודה:

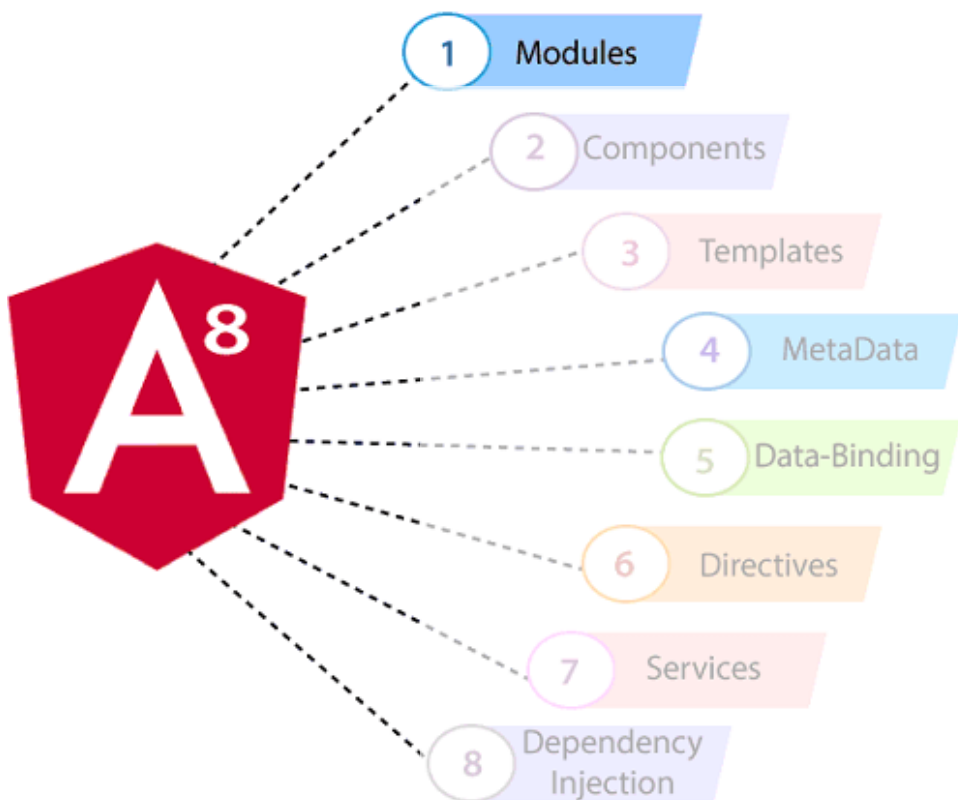
```
C:\Users\morac\angularProjects>ng new my-app
? Would you like to add Angular routing? No
? Which stylesheet format would you like to use? CSS
CREATE my-app/angular.json (3566 bytes)
CREATE my-app/package.json (1258 bytes)
CREATE my-app/README.md (1023 bytes)
CREATE my-app/tsconfig.base.json (458 bytes)
CREATE my-app/tsconfig.json (426 bytes)
CREATE my-app/tslint.json (3184 bytes)
CREATE my-app/.editorconfig (274 bytes)
CREATE my-app/.gitignore (631 bytes)
CREATE my-app/.browserslistrc (853 bytes)
CREATE my-app/karma.conf.js (1018 bytes)
CREATE my-app/tsconfig.app.json (292 bytes)
```



# יצירה והרצה של אפליקציה



# ארכיטקטורת יישום Angular



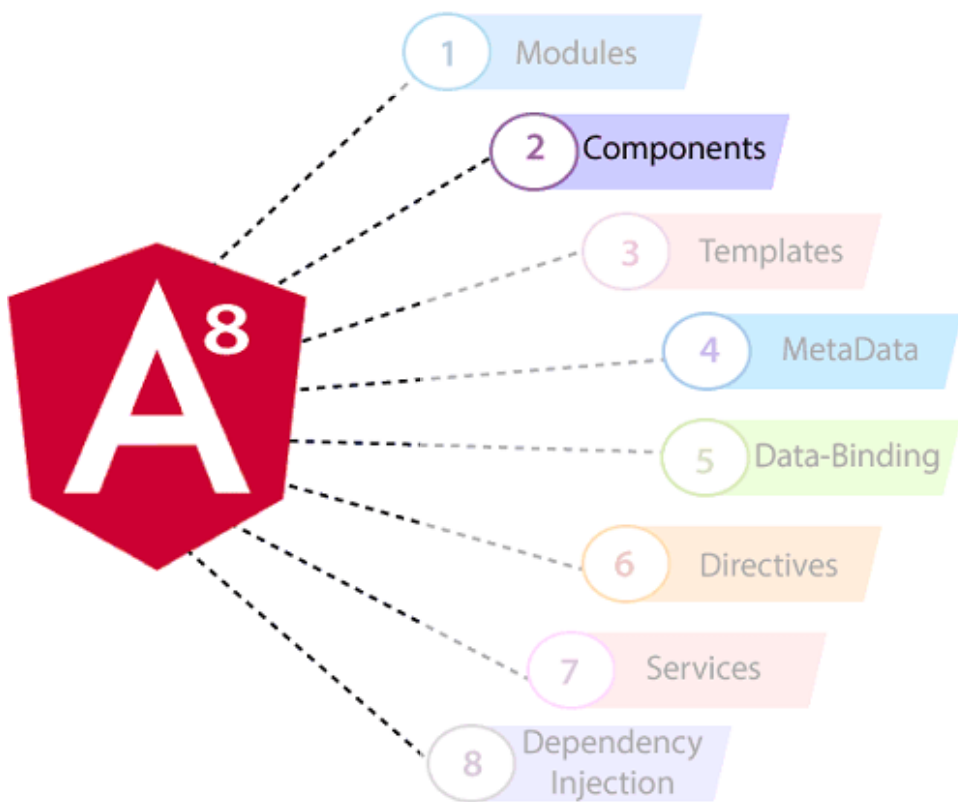
- Module הוא סט של רכיבים בסיסיים ב-angular

- הרכיבים יכולים להיות:  
components, directives, pipes ,services

- לכל אפליקציית angular יש את ה-module הראשי שלה

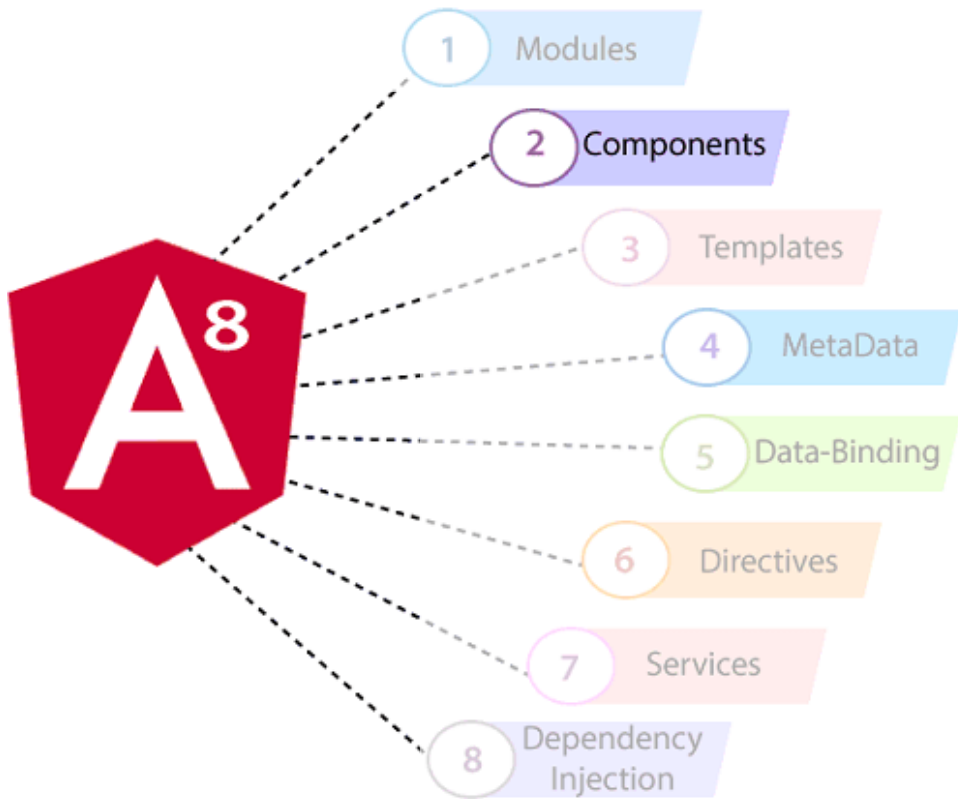
- ניתן להוסיף modules נוספים

# ארכיטקטורת יישום Angular

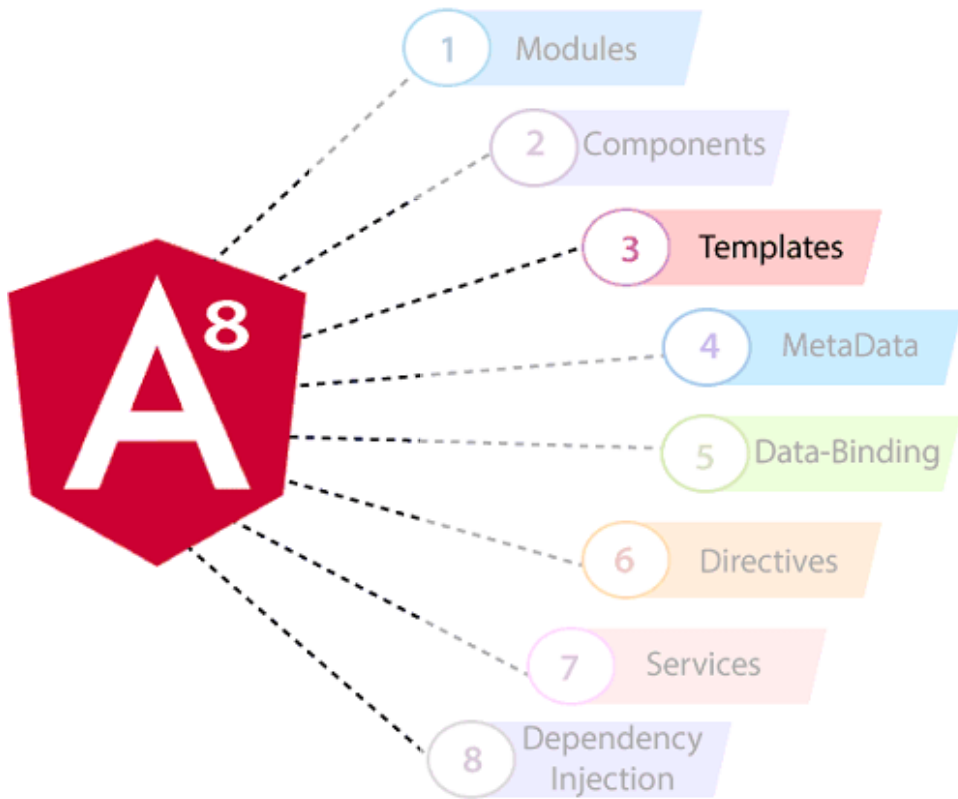


- Component היא אבן הבניין של אפליקציית אנגולר, חלק של ממשק המשתמש (UI)
- מורכב מקובץ Typescript, המכיל מחלקה ייעודית שמטפלת בקומפוננטה
- המתודות הרלוונטיות אליה יהיו ב-class שלה. למשל מתודות שמגיבות לאירועי קליק, או לשינוי תוכן בקומפוננטה וכדומה
- דוגמאות: קומפוננטה של סרגל ניווט, קומפוננטה של התוכן

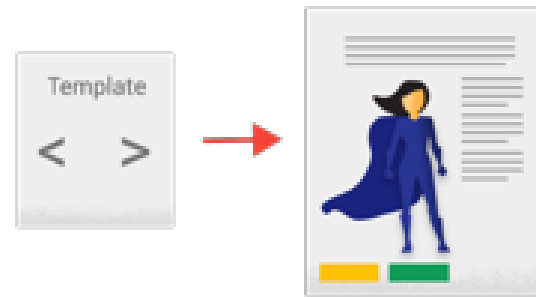
# ארכיטקטורת יישום Angular



# ארכיטקטורת יישום Angular



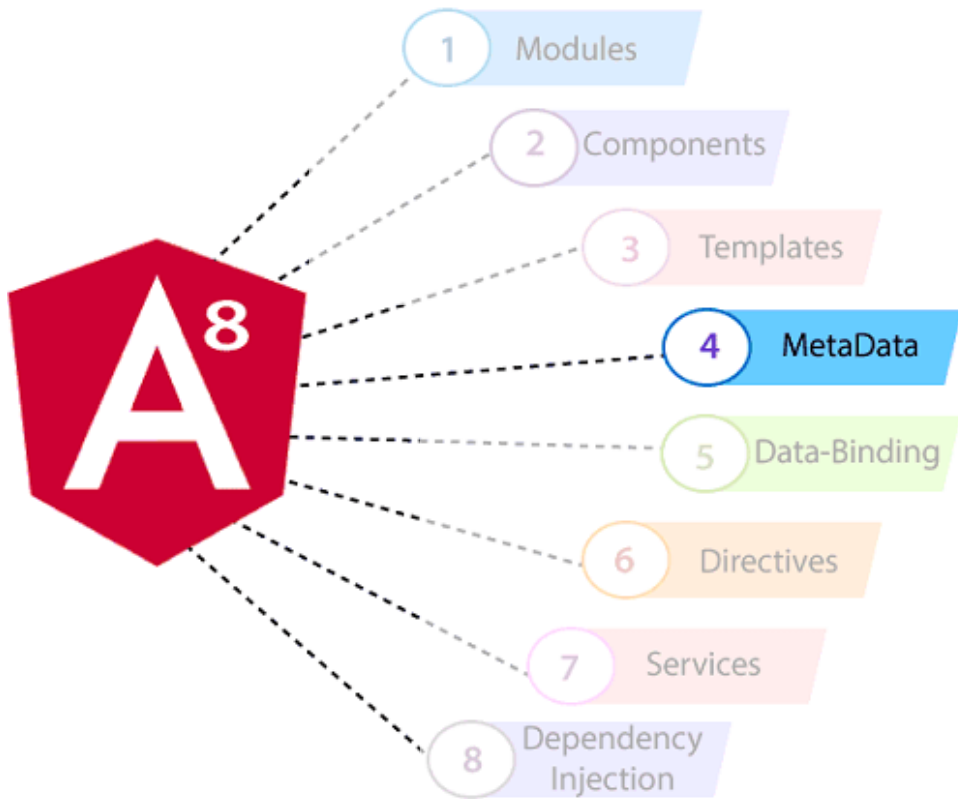
■ Template הוא למעשה קוד ה-html של הקומפוננטה. כלומר, איך הקומפוננטה תוצג למשתמש.



סוגי Templates:

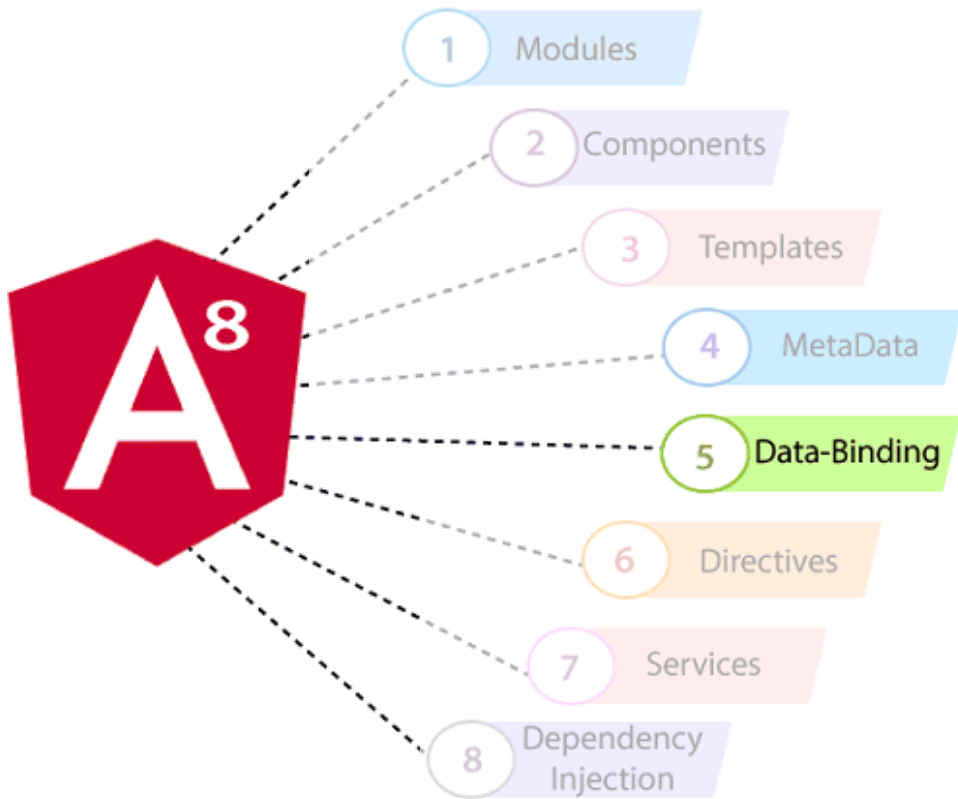
- Inline template
- External template (template file)

# ארכיטקטורת יישום Angular



- MetaData היא למעשה פיסת "מידע על המידע"
- מספק מידע נוסף על הקומפוננטה לאנגולר ומשמש לעיבוד מידע הנמצא במחלקת הקומפוננטה.
- דוגמה למטא-דטה הם השדות ב- **Decorator**
  - למשל השדות ב- `@Component`  
`selector, templates, styles...`

# ארכיטקטורת יישום Angular



- Data Binding מגדיר את הקשר בין הקומפוננטות ל-DOM\*. טכניקה לקישור בין המידע הקיים באפליקציה לבין התצוגה

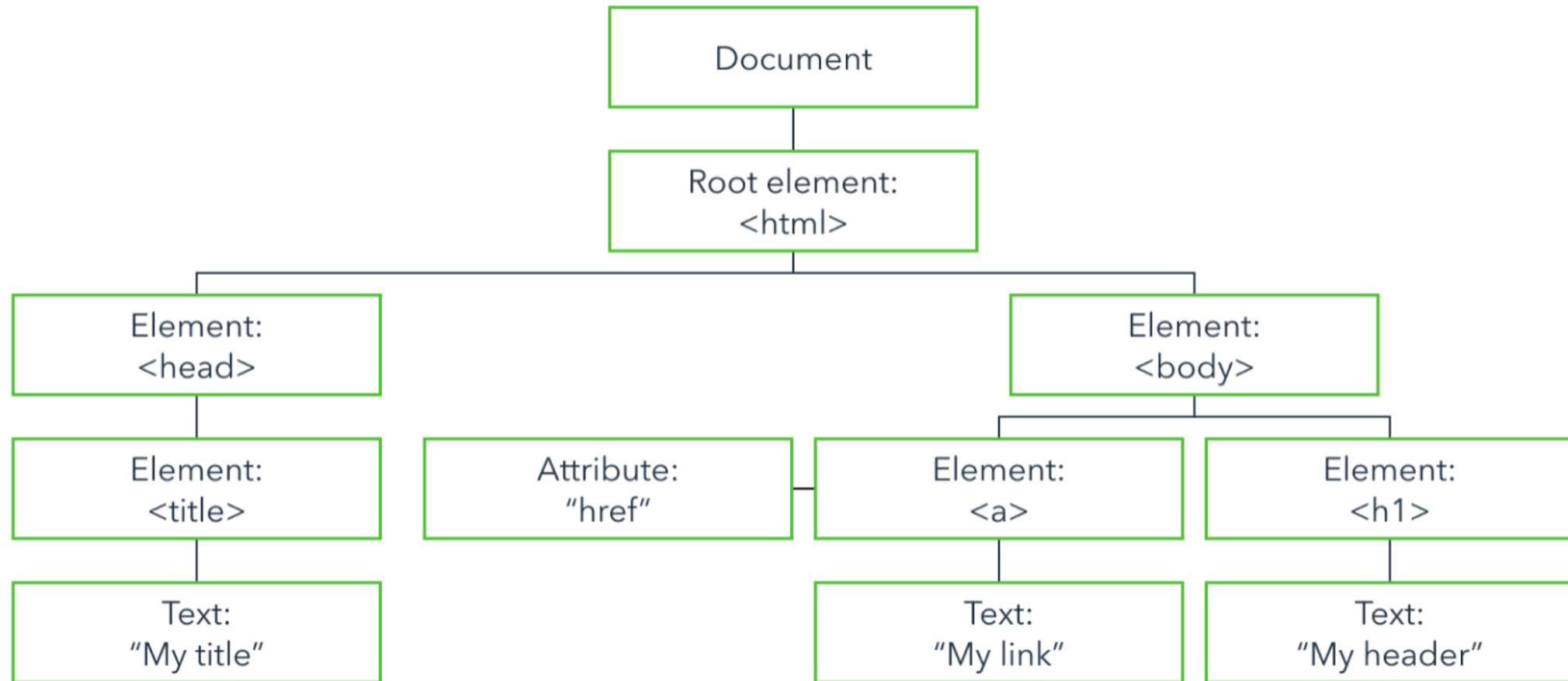
- במילים אחרות, הקשר בין קוד ה-ts של הקומפוננטה וה-template לבין מה שהמשתמש רואה.

- One-way Data Binding

- Two-way Data Binding

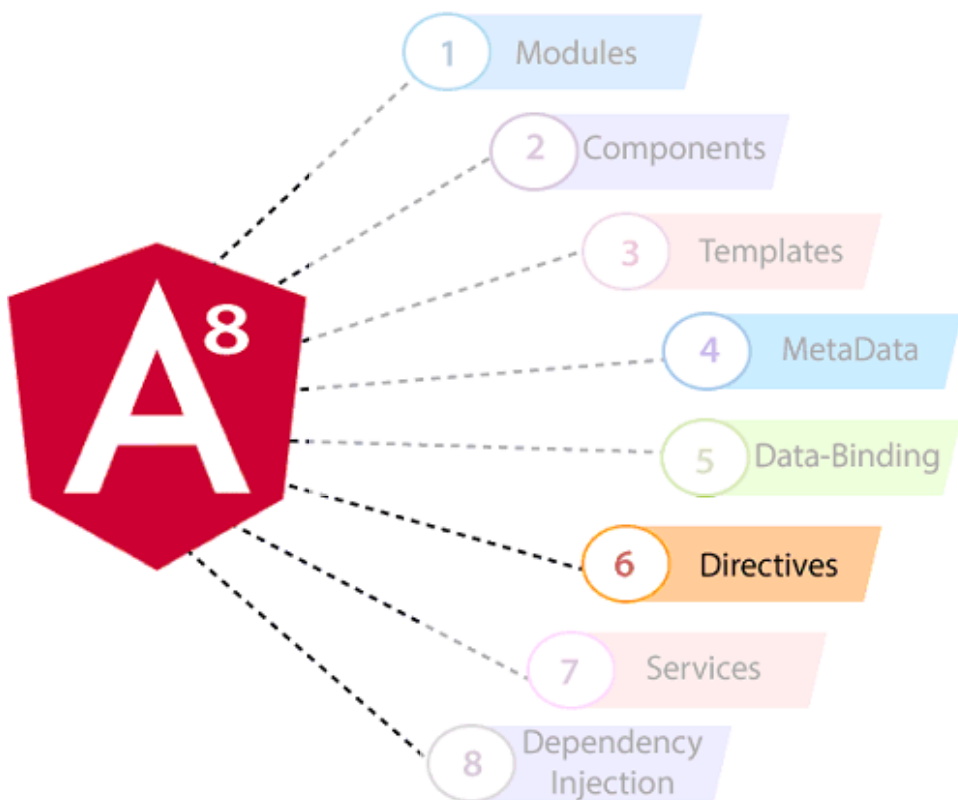
\*The Document Object Model (DOM) is a programming API for HTML and XML documents. It defines the logical structure of documents and the way a document is accessed and manipulated.

# DOM in Angular





# ארכיטקטורת יישום Angular



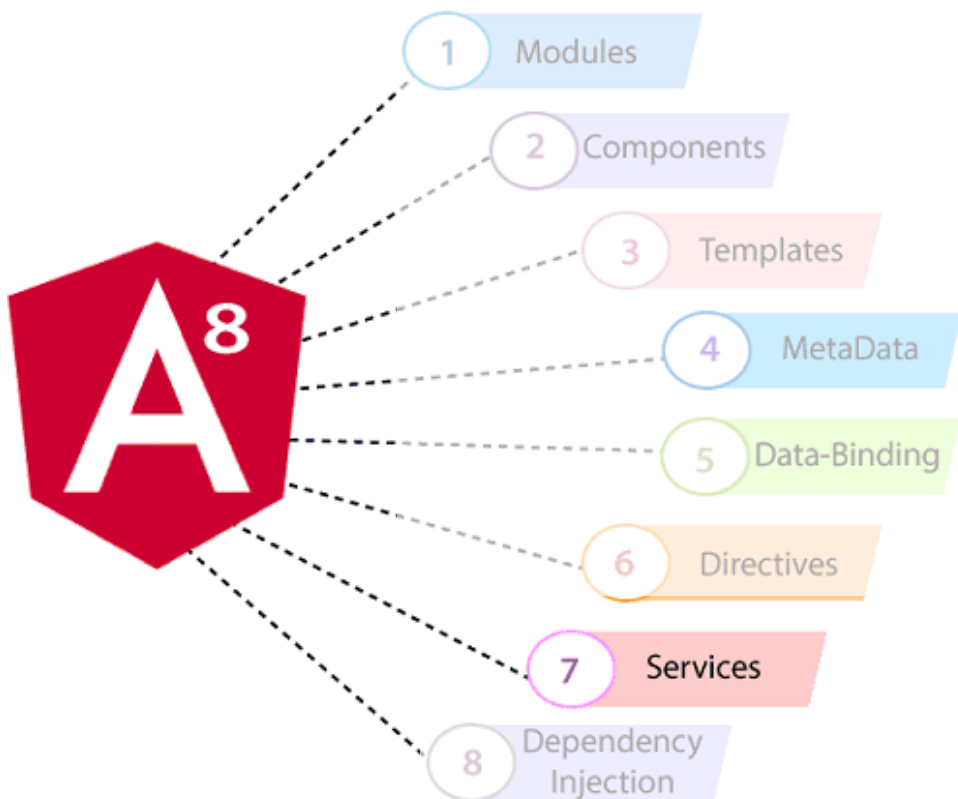
- הוראות Directives מאפשרות הרחבת אוצר המילים של קוד ה-HTML ע"י הוספה שלהם כשדות לתגיות.

- הוראות ה-Directives אומרות לאפליקציית angular מה עליה לעשות.

- דוגמאות:

`*ngIf`, `*ngSwitch`, `*ngFor`

# ארכיטקטורת יישום Angular

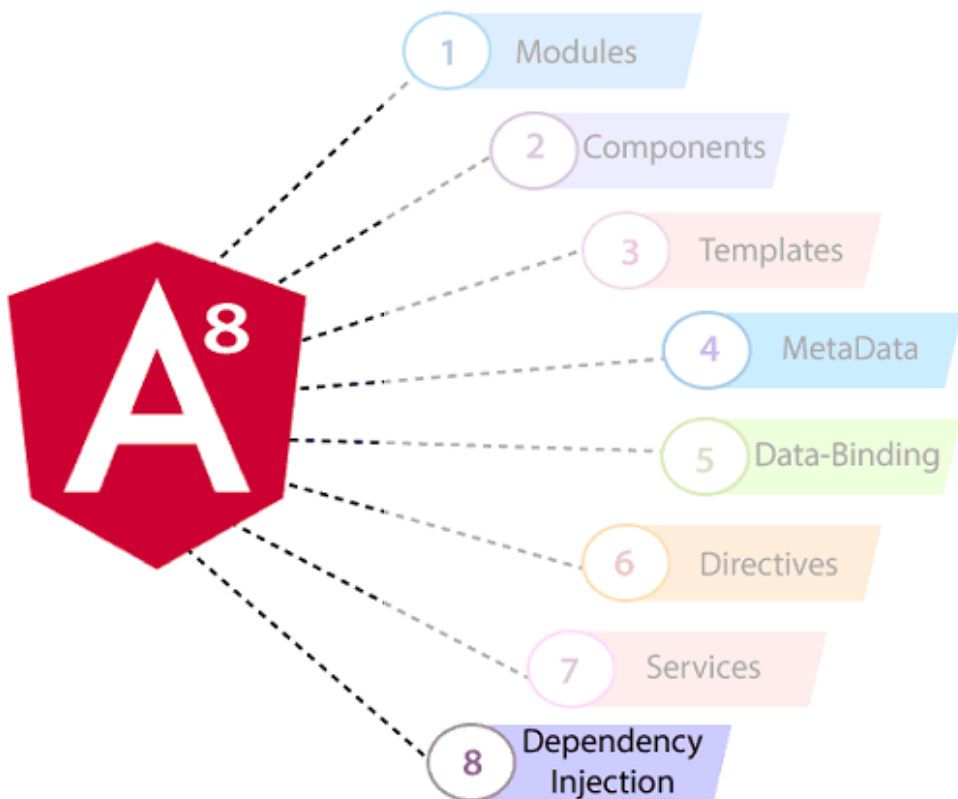


■ Service זו בדרך כלל מחלקה שמספקת שירות כלשהו המשותפת לכמה רכיבים באפליקציית angular.



■ המחלקה תכיל לוגיקה כלשהי המתאימה לקומפוננטה אחת או יותר.

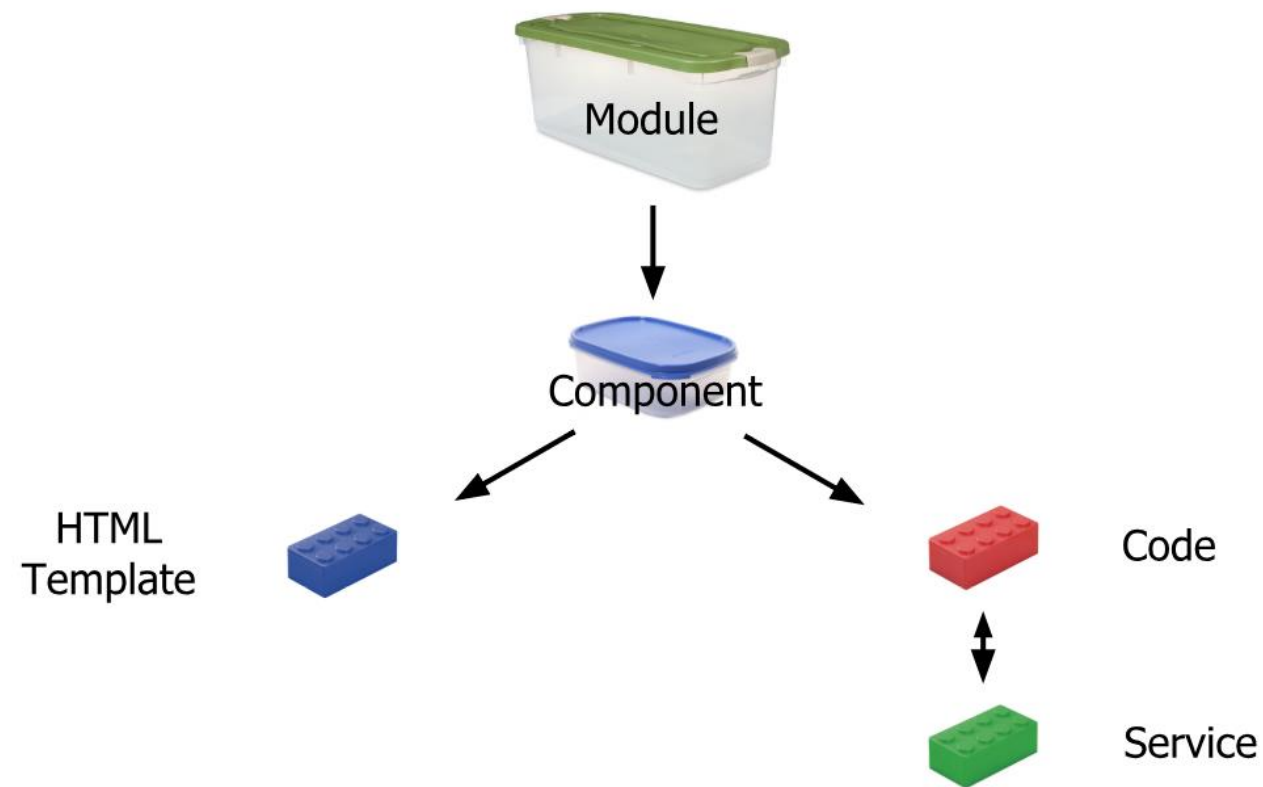
# ארכיטקטורת יישום Angular



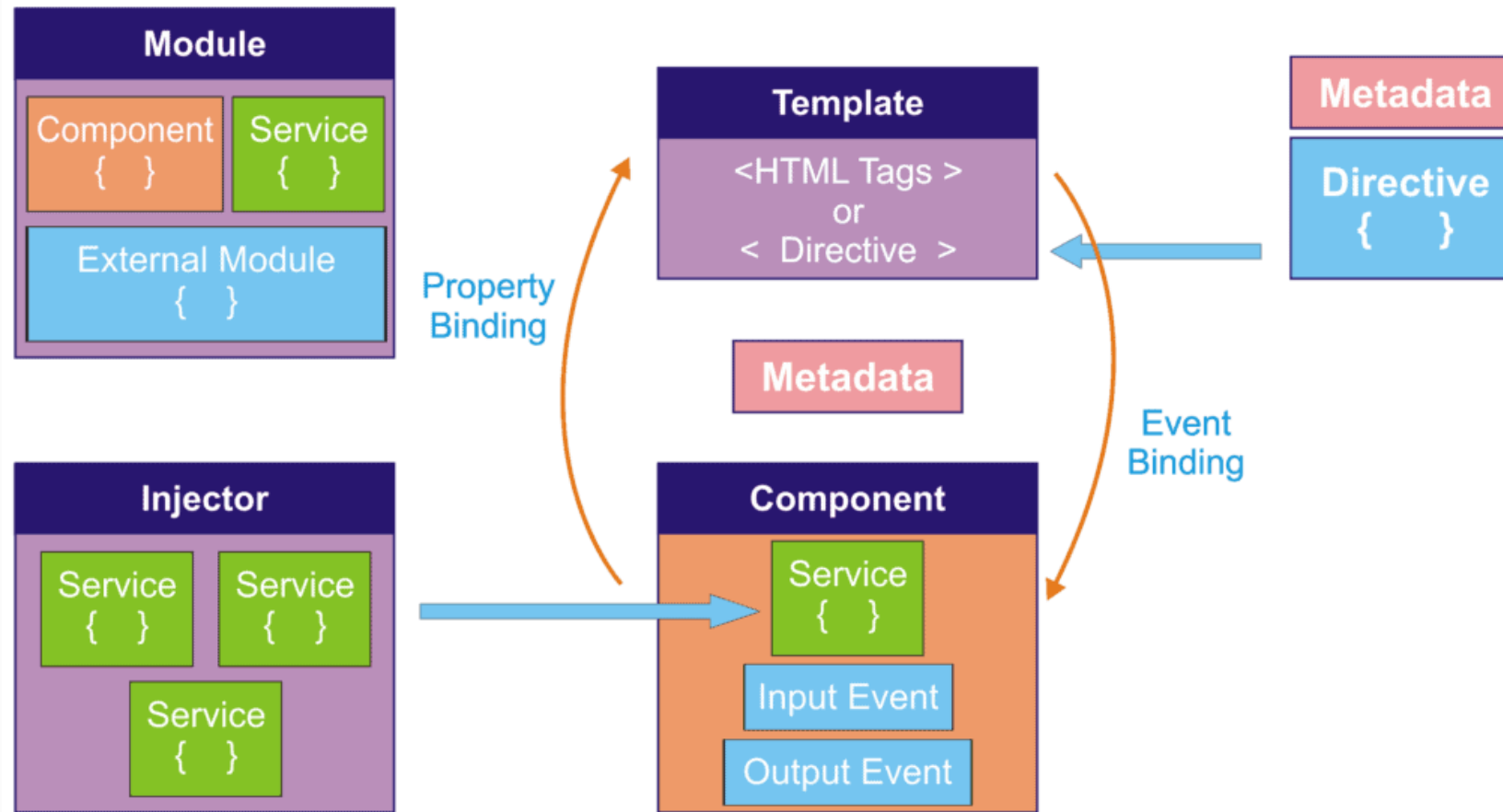
- Dependency Injection זוהי תבנית עיצוב המאפשרת להפוך את הקוד לדינאמי ויעיל.
- מאפשרת לבדוק את הקוד שלנו ביעילות רבה ולא מצריכה שינויים רבים בקוד כאשר מתבצע שינוי בשירות כלשהו.
- מאפשרת יכולת להוסיף את הפונקציונליות של הרכיבים בזמן הריצה.

# ארכיטקטורת יישום Angular

---



# ארכיטקטורת יישום Angular

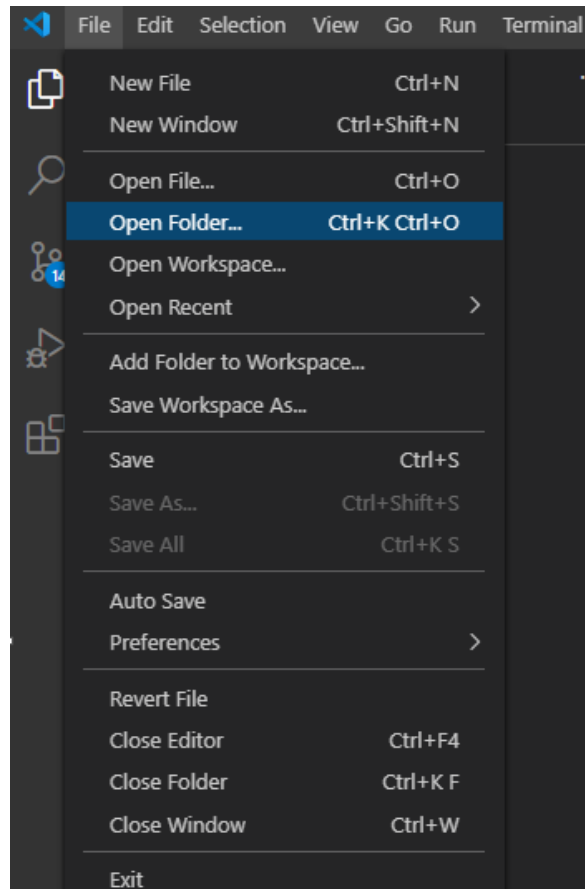


# יצירת פרויקט חדש

■ יצירה ב-cmd

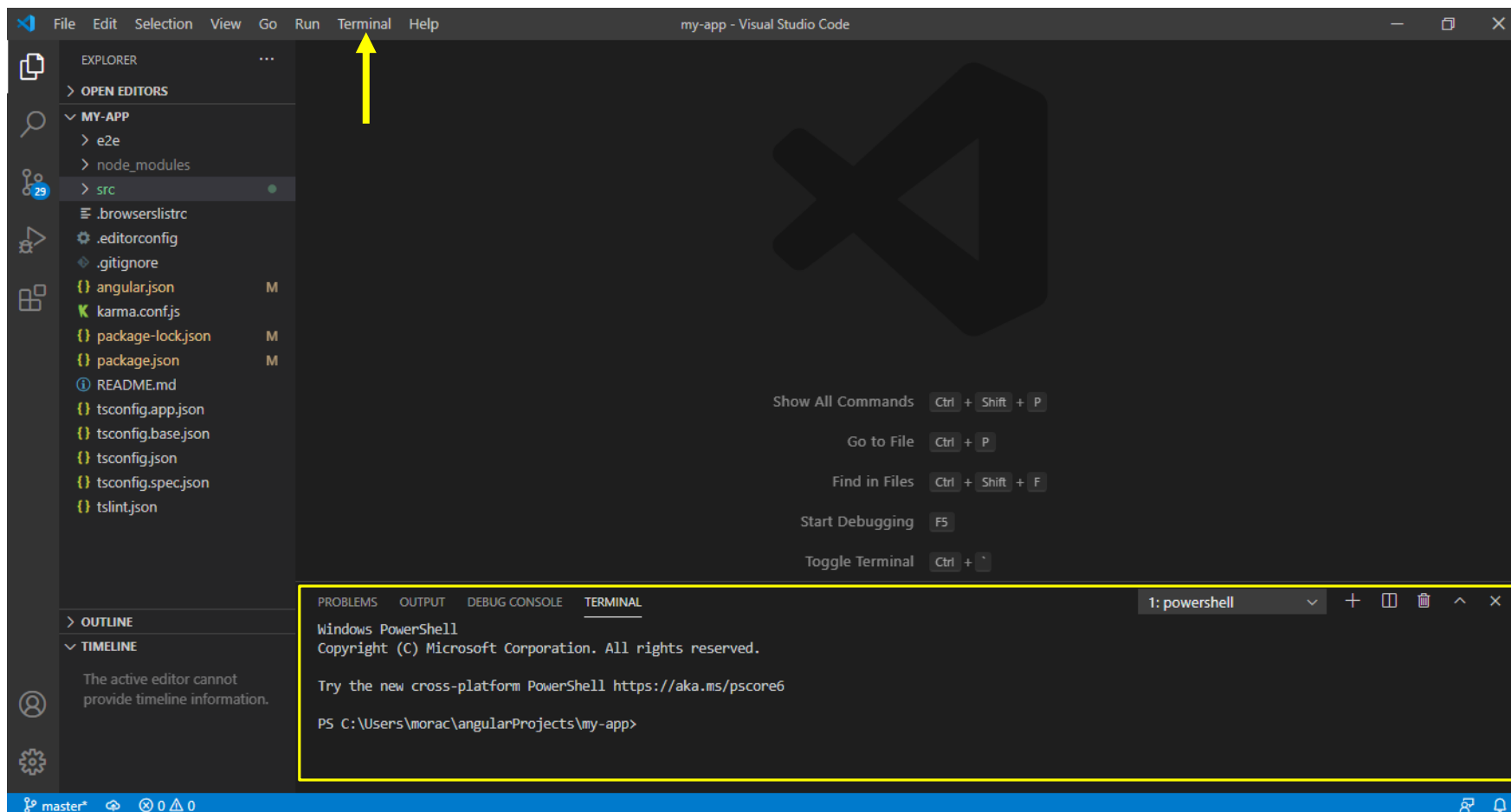
```
C:\Users\morac\angularProjects>ng new my-app
? Would you like to add Angular routing? No
? Which stylesheet format would you like to use? CSS
CREATE my-app/angular.json (3566 bytes)
CREATE my-app/package.json (1258 bytes)
CREATE my-app/README.md (1023 bytes)
CREATE my-app/tsconfig.base.json (458 bytes)
CREATE my-app/tsconfig.json (426 bytes)
CREATE my-app/tslint.json (3184 bytes)
CREATE my-app/.editorconfig (274 bytes)
CREATE my-app/.gitignore (631 bytes)
CREATE my-app/.browserslistrc (853 bytes)
CREATE my-app/karma.conf.js (1018 bytes)
CREATE my-app/tsconfig.app.json (292 bytes)
CREATE my-app/tsconfig.spec.json (272 bytes)
```

# יצירת פרויקט חדש



- פתיחה ב-VS code
- או `code` ב-cmd

# יצירת פרויקט חדש





# תהליך הפעלת האפליקציה

---

■ התהליך נקרא bootstrapping

1. Index.html loads
2. Angular, Third-party libraries & Application loads
3. Main.ts the application entry point
4. Root Module
5. Root Component
6. Template

# תהליך הפעלת האפליקציה

## 1. *Index.html loads*

```
<> index.html X
src > <> index.html > ...
1  <!doctype html>
2  <html lang="en">
3  <head>
4    <meta charset="utf-8">
5    <title>MyApp</title>
6    <base href="/">
7    <meta name="viewport" content="width=device-width, initial-scale=1">
8    <link rel="icon" type="image/x-icon" href="favicon.ico">
9  </head>
10 <body>
11   <app-root></app-root>
12 </body>
13 </html>
```

# תהליך הפעלת האפליקציה

## 2. *Angular, Third-party libraries & Application loads*

```
angular.json ×
angular.json > {} projects > {} my-app > {} architect > {} build > {} configurations > {} prod
1  {
2    "$schema": "./node_modules/@angular/cli/lib/config/schema.json",
3    "version": 1,
4    "newProjectRoot": "projects",
5    "projects": {
6      "my-app": {
7        "projectType": "application",
8        "schematics": {},
9        "root": "",
10       "sourceRoot": "src",
11       "prefix": "app",
12       "architect": {
13         "build": {
14           "builder": "@angular-devkit/build-angular:browser",
15           "options": {
16             "outputPath": "dist/my-app",
17             "index": "src/index.html",
18             "main": "src/main.ts",
19             "polyfills": "src/polyfills.ts",
```

# תהליך הפעלת האפליקציה

## 3. *main.ts* the application entry point

```
TS main.ts ×
src > TS main.ts > ...
1  import { enableProdMode } from '@angular/core';
2  import { platformBrowserDynamic } from '@angular/platform-browser-dynamic';
3
4  import { AppModule } from './app/app.module';
5  import { environment } from './environments/environment';
6
7  if (environment.production) {
8    enableProdMode();
9  }
10
11  platformBrowserDynamic().bootstrapModule(AppModule)
12  .catch(err => console.error(err));
```

The responsible module for loading the Angular application in the desktop browser.

Declaring the root module

# תהליך הפעלת האפליקציה

## 4. Root Module

```
TS app.module.ts •
src > app > TS app.module.ts > ...
1  import { BrowserModule } from '@angular/platform-browser';
2  import { NgModule } from '@angular/core';
3
4  import { AppComponent } from './app.component';
5
6  @NgModule({
7    declarations: [
8      AppComponent
9    ],
10   imports: [
11     BrowserModule,
12   ],
13   providers: [],
14   bootstrap: [AppComponent]
15 })
16 export class AppModule { }
```

### imports

All the external modules required including other Angular modules, that is used by this Angular Module

### Declarations

List of components, directives, & pipes that belong to this Angular Module.

### Providers

The Providers array listed the services we create. The Angular Dependency injection framework injects these services in components, directives. pipes and other services.

### Bootstrap

The component that angular should load, when this Angular Module loads. The component must be part of this module. We want AppComponent load when AppModule loads, hence we list it here.

# תהליך הפעלת האפליקציה

---

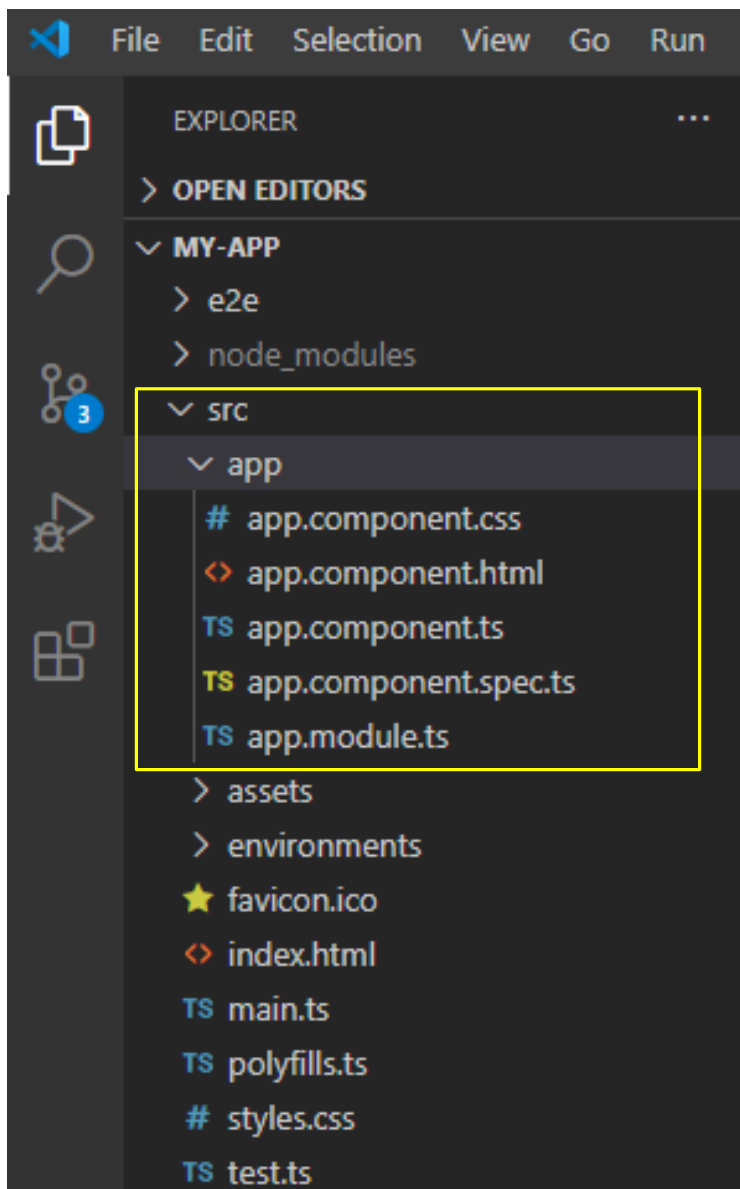
## 5. Root Component

```
TS app.component.ts ●
src > app > TS app.component.ts > ...
1  import { Component } from '@angular/core';
2
3  @Component({
4    selector: 'app-root',
5    templateUrl: './app.component.html',
6    styleUrls: ['./app.component.css']
7  })
8  export class AppComponent {
9    title = 'Mor Ashkenazi'
10 }
```

# תהליך הפעלת האפליקציה

## 6. Template

```
<> index.html X
src > <> index.html > ...
1  <!doctype html>
2  <html lang="en">
3  <head>
4    <meta charset="utf-8">
5    <title>MyApp</title>
6    <base href="/">
7    <meta name="viewport" content="width=device-width, initial-scale=1">
8    <link rel="icon" type="image/x-icon" href="favicon.ico">
9  </head>
10 <body>
11   <app-root></app-root>
12 </body>
13 </html>
```



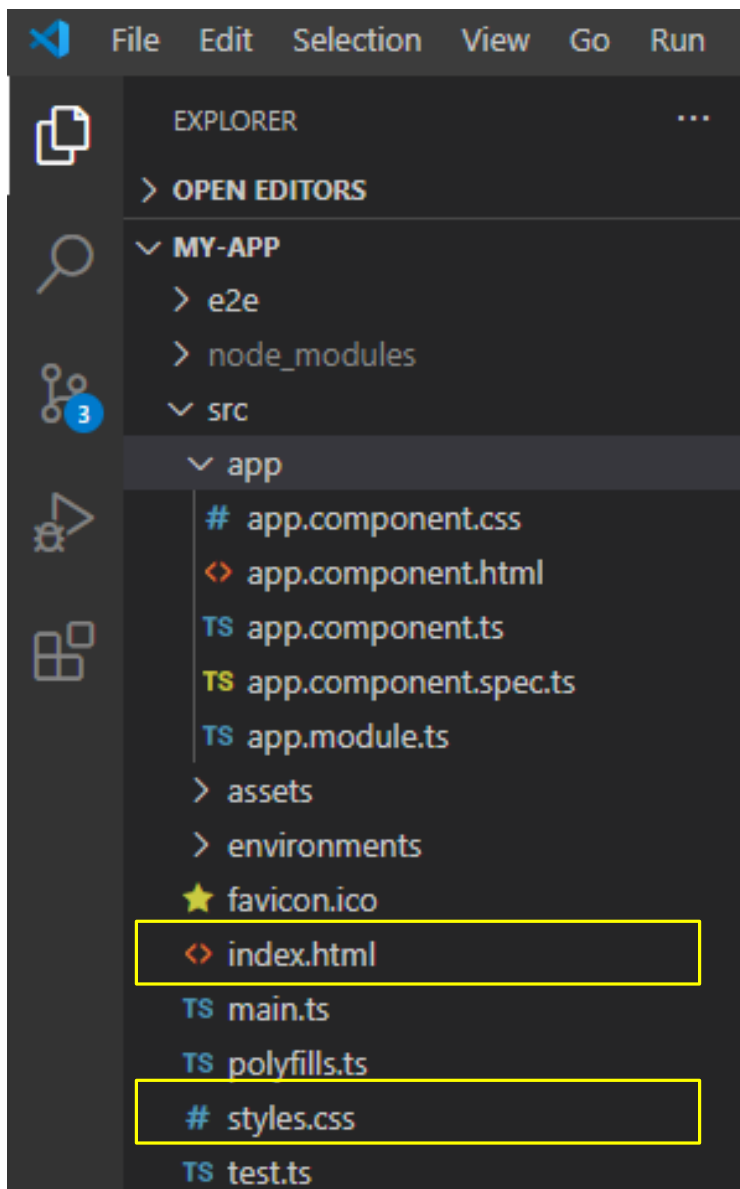
# התיקייה src/app

■ **src/app/app.module.ts**  
המודול המרכזי האפליקציה, ובתוכו נגדיר את המודולים והקומפוננטות בהם נשתמש.

■ **src/app/app.component.ts**  
מכיל את הקוד עבור הקומפוננטה המרכזית של האפליקציה.

■ **src/app/app.component.html**  
בתוכו ימצא ה-html של כל האפליקציה מכיוון שלתוכו נשבץ את ה-html של יתר הקומפוננטות בצורת תגיות שייצגו את הקומפוננטות השונות.





# התיקייה src/app

## ■ src/index.html

הוא קובץ ה-html המרכזי שמשותף לכל מרכיבי האפליקציה.

## ■ src/styles.css

הוא קובץ ה-CSS המשותף לכל מרכיבי האפליקציה.

# התיקייה src/app

## ■ src/angular.json

קובץ הגדרות וקונפיגורציות הרלוונטיות לסביבת העבודה ולפרויקט.

למשל, ניתן להגדיר שם אילו קבצים אחראים על עיצוב יישום האנגולר.

