# **Angular**

Framework מבית גוגל, הושק לראשונה ב 2010 בגרסה ראשונה שנקראה AngularJS.

בגרסה הראשונה שעבדה בצורה שונה ממה שהיום עבדו עם CDN, בדומה לספריות כמו jQuery...

מגרסה Angular 2.0 עשו שכתוב מחדש של כל ה Framework, וכיום הוא ממש SPA בקומפוננטות.

איתו אנו עובדים היום על שלל גרסאותיו, בגרסה זו ומעלה העבודה היא על תשתית NPM.

**התקנת Angular גלובלית:**

npm i –g @angular/cli

CLI = Command line interface

פקודות להרצה ב CMD.

**בדיקת גרסה:**

ng version

**יצירת פרויקט חדש:**

ng new <project-name>

**יצירת פרויקט חדש – ללא בדיקות:**

ng new <project-name> --skip-tests

**יצירת קומפוננט:**

ng g c <path + component-name>

**Decorator**

זו פקודה שאנו כותבים מעל אלמנט תכנותי כלשהי כגון data-member method, class וכו' המציינת עבור ה Framework משהו.

כל Decorator מתחיל ב @.

לדוגמא: @Component זהו Decorator המגדיר ל Angular שה Class הבא הינו קומפוננט.

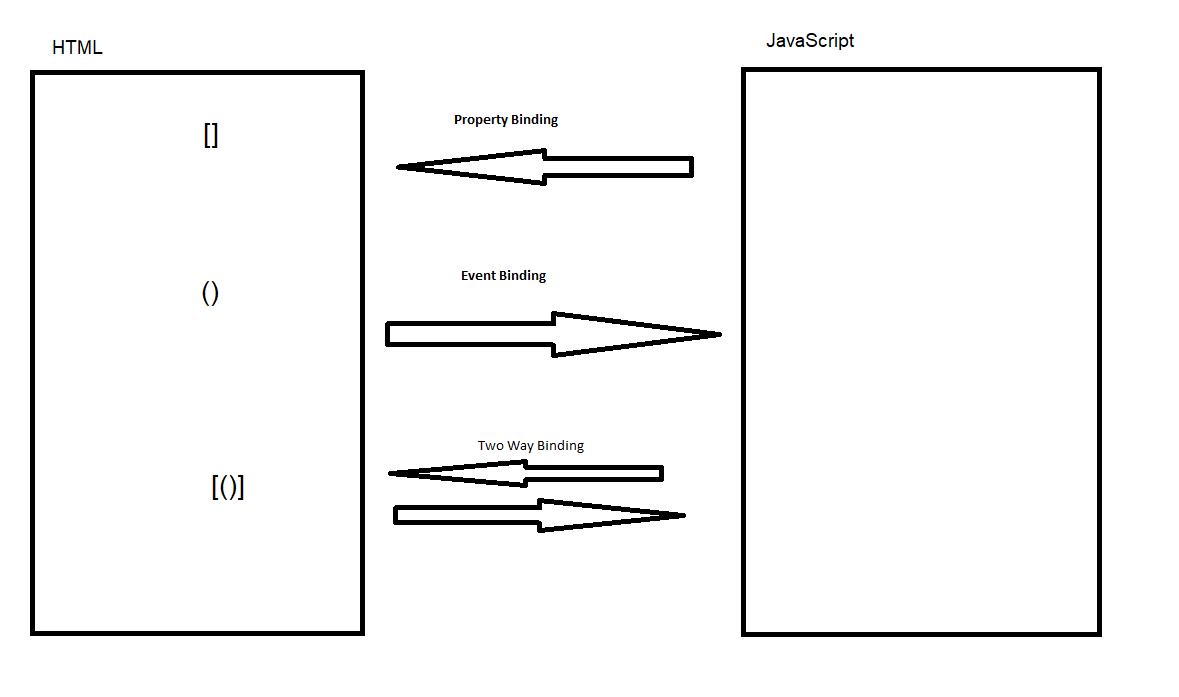
**Data binding**

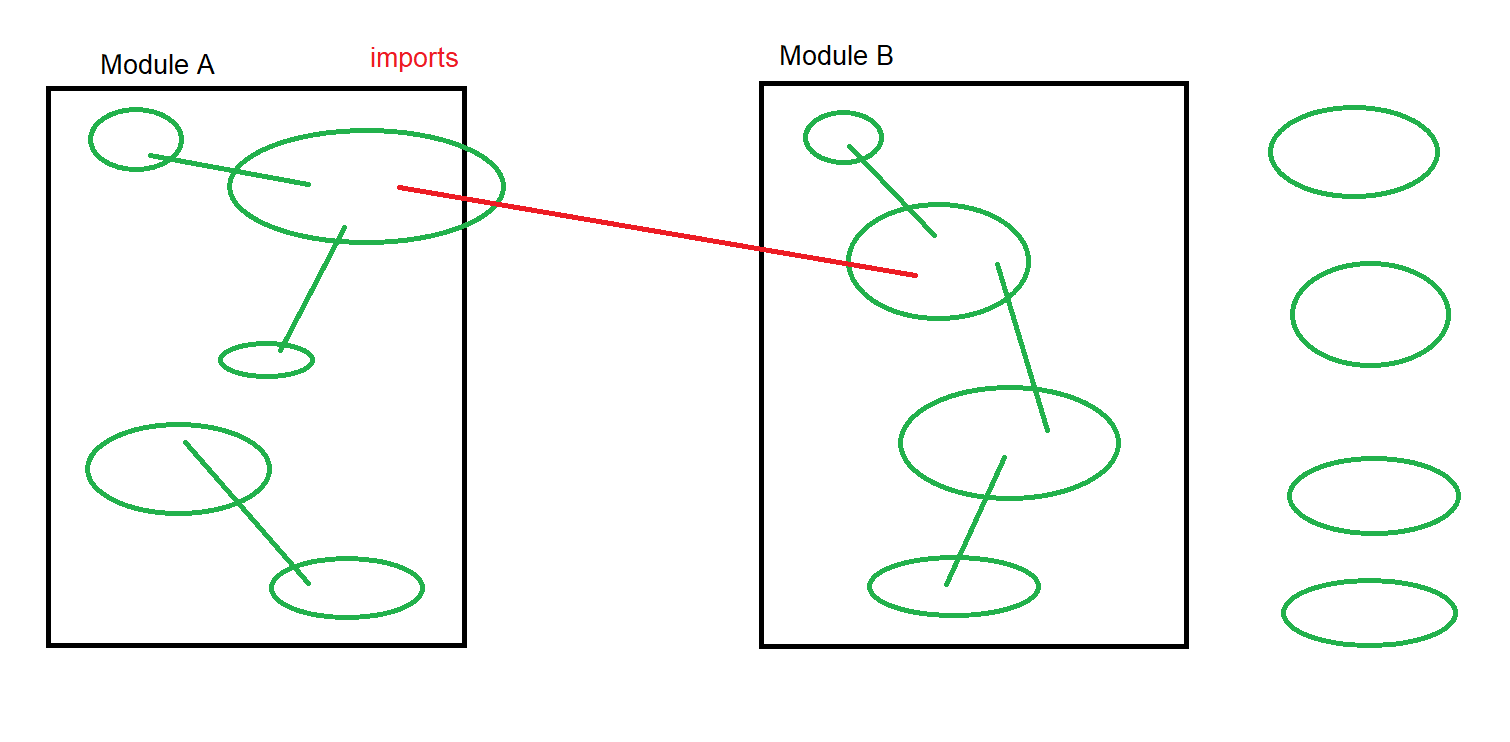
קישור בין מקור ליעד, שינוי של המקור משפיע על היעד.

קישור בין ערך הנמצא ב Class של הקומפוננט ל HTML שלה.

ישנם ארבעה סוגי Data Binding ב Angular.

1. **Property Binding** – קישור בין משתנה מחלקת הקומפוננט למאפיין ב HTML של הקומפוננט.
2. **Event Binding –**קישור בין אירוע ב HTML של הקומפוננט לבין פונקציה במחלקת הקומפונטה, האירוע מפעיל את הפונקציה.
3. **Two Way Binding** – קישור בין תיבת קלט למשתנה.   
   שינוי ערך התיבה יכניס מיידית את הערך למשתנה, שינוי המשתנה יציג אותו בתיבה
4. **Interpolation** – הצגת מידע הקיים במחלקת הקומפוננט ישירות בתוך ה HTML.





**Directive**

Custom HTML Attributes של Angular, הם מתנהגים כמו HTML Attributes.

ישנם 2 סוגים:

1. Attribute Directive – שינו כלשהו המבוצע על התגית, שינוי עיצוב, שינוי מבנה, שינוי התנהגות.
2. Structural Directive – מעיין Attribute הנותן הוראה כמה פעמים להכניס את התגית ל DOM.

כל Structural Directive מתחיל ב \*.

**Pipe**

שינוי המתבצע על ערך ב Interpolation.

בפועל זו פונקציה במקבלת את הערך משמאל כארגומנט.

בגלל שזו פונקציה ניתן להעביר אלו עוד ארגומנטים כקונפיגורציה.

**Template Reference variable**

מזהה ייחודי שאנו נותנים לתגיות.

ידוע שאין לתת מאפיין ID במערכות מבוססי קומפוננטות.

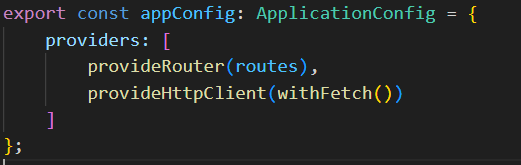
זו מילה הנכתבת עם # בתחילתה, מאחורי הקלעים אנגולר יוצרת אותו באובייקט רפרנס של התגית ב DOM.

ניתן לגשת לאלמנט זה מ HTML, מה TypeScript וניתן לשלוח אותו כארגומנט לפונוקציה.

**גלישה לשרת מרוחק**

באנגולר אנו משתמשים באובייקט בשם httpClient המגיע ממודול הנקרא httpClientModule אני צריכים לבקש מAngular להזריק לנו אותו לקלאס שנצרה להשתמש בו.

ע"מ לאפשר שימוש ב Module יש לעדכן את app.config.ts:



**Dependency injection – DI**

זהו שיטת עבודה נפוצה בשפות תכנות שונות, בה המערכת מנהלת את האובייקטים ומזריקה לנו אותם בעת הצורך לנקודה בה אנו רוצים לעשות בהם שימוש.

כאשר אנו רוצים להשתמש בשירות כלשהו אני מבקשים אותו לתוך פונקציית הבנאי במחלקה שאנו עובדים.

אנגולר בונה אובייקט שלו ומזריקה אותו.

**Observable**

זהו אובייקט המגיע מספרייה חיצונית הקיימת גם באנגולר בשם "rxjs".

אובייקט זה דומה ל Promise אך הוא יותר מאסיבי ובעל יכולות.

Promise מיועד להריץ קוד אסיכרוני ולדווח הצלחה או כשלון.

Observable מיועד להריץ קוד אסינכרוני ולדווח מידע לאורך זמן, או כשלון בודד.