

Angular

מצגת 2

- א. ביטוי TypeScript
- ב. משתנים
- ג. מערך
- ד. הצגת ערכי מערך בלולאה
- ה. פונקציות
- ו. מחלקה

א. ביטוי TypeScript

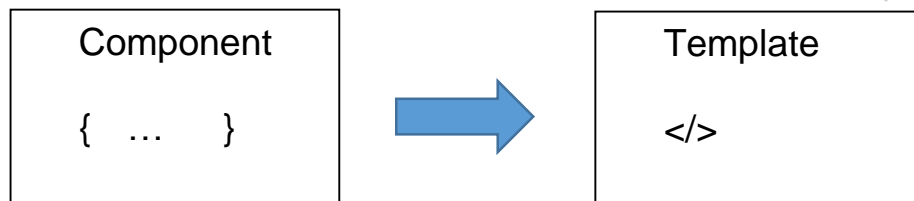
קובץ תבנית HTML - app.component.html

- קובץ זה מכיל קוד HTML רגיל עם תחביר נוסף ספציפי ל-Angular - לקבלת נתונים ממחלקת AppComponent.
- string interpolation – כותבים ביטוי בתוך 2 סוגריים מסולסלים.
- עדכן קובץ תבנית (HTML):

```
<p>{{1}}</p>  
<p>{{2}}</p>  
<p>{{1+2}}</p>
```

ב. משתנים

- מחלקת AppComponent - היא מחלקת TypeScript הכוללת משתנים ופעולות של הקומפוננטה.
- המחלקה ממוקמת בקובץ עם סיומת component.ts.
- מקושר עם קובץ תבנית HTML (binding):



דוגמה 1

- הוספת 2 משתנים:
 - x - מטיפוס number.
 - y - מטיפוס number.
- הפעולה constructor מאתחלת תוכן משתנים – מופעל בהפעלת התכנית.
- מילת המפתח "export" ב-Angular משמשת להפיכת מחלקות לזמינים לשימוש בקבצים אחרים.
- @Component
- מספק נתונים לרכיב ליצור מופע ולהשתמש ברכיב בזמן ריצה.

- עדכן AppComponent בקובץ מחלקת הקומפוננטה (.component.ts):

```
...

export class AppComponent
{
    x: number;
    y: number;

    constructor()
    {
        this.x = 2;
        this.y = 3;
    }
}
```

- עדכן קובץ תבנית (.HTML):

```
<p> x = {{ x }} </p>
<p> y = {{ y }} </p>
<p> x + y = {{ x + y }} </p>
```

תרגיל 1

- הוסף לדוגמה פעולות כפל חיסור וחילוק x, y.

* בהגשה יש לצרף את תיקיית app

דוגמה 2

- הוספת 2 משתנים:
 - title - מטיפוס string.
 - myHero - מטיפוס string.
- הפעולה constructor מאתחלת תוכן משתנים – מופעל בהפעלת התכנית.
- עדכן AppComponent בקובץ מחלקת הקומפוננטה (.component.ts):

```
...
export class AppComponent
{
    hero1: string;
    hero2: string;

    constructor()
    {
        this.hero1= 'Batman';
        this.hero2= 'Spiderman';
    }
}
```

- עדכן קובץ תבנית (HTML):

```
<p> {{ hero1 }} </p>
<p> {{ hero2 }} </p>
```

תרגיל 2

- הצג 5 פירות ממערך.
- הנתונים צריכים להלקח מקובץ ה-TypeScript של הקומפוננטה.

* בהגשה יש לצרף את תיקיית app

ג. מערך

הקצאת מערך

- עדכן AppComponent בקובץ מחלקת הקומפוננטה (.component.ts):

```
...

export class AppComponent
{
    title: string;
    heroes: string[];
    myHero: string;

    constructor()
    {
        this.title = 'My Heroes';
        this.heroes = ['Spiderman', 'Batman', 'Thor', 'Superman'];
        this.myHero = this.heroes[1];
    }
}
```

- עדכן קובץ תבנית (.HTML):

```
<h1> {{ title }} </h1>
<h2> Heroes: </h2>
<p> {{ heroes[0] }} </p>
<p> {{ heroes[1] }} </p>
<p> {{ heroes[2] }} </p>
<p> {{ heroes[3] }} </p>
<h2> My favorite hero is: {{ myHero }} </h2>
```

תרגיל 3

- הצג את 5 בפירות מהמערך.
- לדוגמה:

```
apple  
orange  
...
```

- הנתונים צריכים להילקח מקובץ ה-TypeScript של הקומפוננטה.

* בהגשה יש לצרף את תיקיית app

נכתב ע"י ניצן דולינסקי ©

ד. הצגת ערכי מערך בלולאה

- הפקודה *ngFor חוזרת על אוסף נתונים (מערך, רשימה וכו'), ויוצרת אלמנט HTML עבור כל אחד מהפריטים מתבנית HTML.

- - *ngFor = "..."

- תכונה בתגית המשמשת לשכפול התגית בלולאה.
- מאפשר לגשת לערכי המערך.

- - *ngFor = " let hero of heroes "

- let hero - הגדרת משתנה hero המקבל בכל איטרציה את הערך מתוך מערך heroes.

- עדכן קובץ תבנית HTML (ללא שינוי בקובץ מחלקת הקומפוננטה):

```
<h1>{{ title }}</h1>
<h2>Heroes:</h2>
<p *ngFor = "let hero of heroes"> {{ hero }} </p>
```

תרגיל 4

- הצג בעזרת לולאה 5 פירות ממערך. לדוגמה:

```
apple
orange
...
```

- הנתונים צריכים להלקח מקובץ ה-TypeScript של הקומפוננטה.

* בהגשה יש לצרף את תיקיית app

הצגת ערך אינדקס בלולאה

- `*ngFor = "let hero of heroes ; let i = index"` – הגדרת משתנה `i` המקבל את מספר האיטרציה.

- עדכן קובץ תבנית HTML (ללא שינוי בקובץ מחלקת הקומפוננטה):

```
<h2>Heroes:</h2>

<p *ngFor = "let hero of heroes ; let i = index"> {{ i }}: {{ hero }} </p>
```

תרגיל 5

- הצג רשימה לולאה מספר סידורי ו-5 פירות ממערך. לדוגמה:

```
0: apple
1: orange
...
```

- הנתונים צריכים להלקח מקובץ ה-TypeScript של הקומפוננטה.

* בהגשה יש לצרף את תיקיית app

ה. פונקציות

- הגדרת פעולה (פונקציה) במחלקת הקומפוננטה –

functionName(parameters)

דוגמה 1 – פונקציה ללא פרמטר

- עדכן AppComponent בקובץ מחלקת הקומפוננטה (.component.ts):

```
...

export class AppComponent
{
    str: string;

    constructor() {
        this.str=""
    }

    func() {
        this.str="hello world";
    }
}
```

- באמצעות שימוש ב-click listener:
(click)="func()" תקרא לפונקציה func().

- עדכן קובץ תבנית (HTML):

```
<button (click)="func()"> press button </button>
<p>str value = {{str}}</p>
```

דוגמה 2 – פונקציה ללא פרמטר

- עדכן AppComponent בקובץ מחלקת הקומפוננטה (.component.ts):

```
...

export class AppComponent
{
    count: number;

    constructor() {
        this.count = 0;
    }

    increment() {
        this.count++;
    }

    decrement() {
        this.count--;
    }
}
```

- התכונה בכפתור "increment (click) = " – קוראת לפעולה increment () בלחיצה על הכפתור.
- התכונה בכפתור "decrement (click) = " – קוראת לפעולה decrement () בלחיצה על הכפתור.
- עדכן קובץ תבנית (HTML):

```
<p>count: {{ count }}</p>
<button (click)="increment()"> Increment </button>
<button (click)="decrement()"> decrement </button>
```

תרגיל 6

- צור פונקציה switchFruitFunc() אשר נקראת מכפתור.
- בכל לחיצה תעדכן משתנה myFruit לערך אחר מתוך המערך.
- בכל לחיצה תעדכן משתנה myFruit לערך אחר מתוך המערך.

* בהגשה יש לצרף את תיקיית app

דוגמה 3 – פונקציה עם פרמטר

- עדכן AppComponent בקובץ מחלקת הקומפוננטה (.component.ts):

```
...

export class AppComponent
{
    str: string;

    constructor() {
        this.str=""
    }

    func(index: number) {
        this.str = "button " + index + " pressed";
    }
}
```

- עדכן קובץ תבנית (HTML):

```
<table style="width:100%">
  <tr>
    <td><button (click)="func(1)" style="width:100%"> 1 </button></td>
    <td><button (click)="func(2)" style="width:100%"> 2 </button></td>
    <td><button (click)="func(3)" style="width:100%"> 3 </button></td>
  </tr>
</table>

<p>str value = {{str}}</p>
```

תרגיל 7

- עדכן דוגמה בקובץ התבנית, AppComponent כך שתוצג תוצאת החיבור בין כל מספרים 1 עד 3 כאשר התוצאה תתעדכן בכל לחיצה.
- הוסף כפתור C אשר יאפס את התוצאה.
- בהגשה יש לצרף את תיקיית app.

* בהגשה יש לצרף את תיקיית app

דוגמה 4 – פונקציה עם פרמטר - אינדקס

- נאתחל מערך prices אשר ייצג בהתאמה את מחיר מערך heroes.
- עדכן AppComponent בקובץ מחלקת הקומפוננטה (.component.ts):

```
...

export class AppComponent
{
    heroes: string[];
    prices: number[];
    sum: number;

    getIndex(index: number)
    {
        this.sum+=this.prices[index];
    }

    constructor()
    {
        this.sum=0;
        this.heroes = ['Spiderman', 'Batman', 'Thor', 'Superman'];
        this.prices = [20,40,10,15];
    }
}
```

- עדכן קובץ תבנית (HTML):

```
<table style="width:100%">
  <tr>
    <td><button (click)="getIndex(i)" *ngFor = "let hero of heroes ; let i = index"> {{hero}} </button></td>
  </tr>
</table>
```

```
<p>sum is: {{sum}}</p>
```

תרגיל 8

- הצג את 5 הפירות כפתורים.
- בלחיצה על כל כפתור יתעדכן ויוצג הסכום.
- הוסף כפתור C אשר יאפס את התוצאה.
- בהגשה יש לצרף את תיקיית app.

* בהגשה יש לצרף את תיקיית app

תרגיל 9

- בנה מחשבון עם לחצנים 0 עד 2
- 1. מבנה קובץ תבנית (HTML):

```
<table style="width:100%">
  <tr>
    <td><button style="width:100%"> 0 </button></td>
    <td><button style="width:100%"> 1 </button></td>
    <td><button style="width:100%"> 2 </button></td>
  </tr>
  <tr>
    <td><button style="width:100%"> c </button></td>
    <td><button style="width:100%"> + </button></td>
    <td><button style="width:100%"> = </button></td>
  </tr>
</table>

<p>output: {{ output }}</p>
<p>result: {{ result }}</p>
```

* בהגשה יש לצרף את תיקיית app

1. הגדרת מחלקה

1. ליצירת מחלקה בצע פקודה (cmd):

```
ng g class Hero
```

- נוסף קובץ hero.ts. עדכן את הקובץ:

```
...

export class Hero
{
    id: number;
    name: string;

    constructor( id: number, name: string )
    {
        this.id = id;
        this.name = name;
    }
}

...
```


- עדכן AppComponent בקובץ מחלקת הקומפוננטה (.component.ts):
יש לייבא מחלקת Hero.

```
import { Hero } from './hero';
```

- עדכן AppComponent בקובץ מחלקת הקומפוננטה (.component.ts):

```
...

export class AppComponent
{
    title: string;
    heroes: Hero[];

    constructor()
    {
        this.title = 'My Heroes';
        this.heroes =
        [
            new Hero(1, 'Spiderman'),
            new Hero(13, 'Batman'),
            new Hero(15, 'Thor'),
            new Hero(20, 'Superman')    ];
    }
}
```

- עדכן קובץ תבנית HTML:

```
<p>{{ heroes[1].name }}</p>
<p>{{ myHero.name }}</p>
```

תרגיל 10

- הצג בעזרת לולאה 5 רשימת פירות הכוללת שם מוצר, מחיר.
- הפירות מאורגנים במערך מחלקות Fruit.
- לדוגמה:

```
name: apple, price: 12
name: orange, price: 16
...
```

* בהגשה יש לצרף את תיקיית app