

TypeScript

Generics וחומרים נוספים

interface

```
interface IPoint {
  x: number;
  y: number;
  print: () => void;
}

class Point implements IPoint {
  constructor(public x: number, public y: number) { }

  public print(): void {
    console.log(`x: ${this.x}, y: ${this.y}`);
  }
}

const point = new Point(10, 20);
point.print();
```

- מהו Interface (ממשק, חוזה מימוש)?
- מה Interface יכול להכיל?
- מי יכול לממש interface?
- כיצד נוכל להשתמש ב-Interface כ-type?
- איך מטמיעים ומממשים interface במחלקה?
- מה זה בא לפתור לנו?

Interface - תרגול

- צרו Interface בשם IRectangle.
- תנו לו את המאפיינים:
 - height: number
 - width: number
 - area: number
 - printlnInfo: () => void
 - draw: (bgColor:string, color:string) => void
- צרו את מחלקת Rectangle אשר תממש ותטמיע את הממשק IRectangle.
- הבנאי של המחלקה ידרוש את הפרמטרים height ו-width.
- מאפיין ה-area יחושב בבנאי לאחר השמת הגובה והרוחב.
- המתודה printlnInfo תדפיס את פרטי המופע לקונסול בצורה מאורגנת.
- המתודה draw תוסיף למסמך ה-HTML המקושר אלמנט div בעל רקע וצבע רקע בהתאם לפרמטר שנשלח אליה, ובעל מימדים בפיקסלים לפי פרטי המופע, במרכז המלבן יופיע השטח המחושב.

Generics

```
interface IList<T> {
    add(item: T): void;
    show(): void;
}

class List<T> implements IList<T> {
    private list: T[] = [];

    public add(item: T): void {
        this.list.push(item);
    }

    public show(): void {
        this.list.forEach(item => console.log(item));
    }
}

const list = new List<number>();
list.add(123);
list.add(52);
list.add(3345);
list.show();

const list2 = new List<string>();
list2.add('!!!!');
list2.add('Hello');
list2.add('World');
list2.add('!!!!');
list2.show();
```

- מהי גנריות ב-TypeScript?
- למה כדאי להשתמש בגנריות?
- מהו טיפוס גנרי (Generic Type)?
- כיצד מגדירים מחלקות גנריות (Generic Classes)?
- איך משתמשים בפרמטרים גנריים במחלקות?
- מה זה בא לפתור לנו?
- כיצד מגדירים ממשקים גנריים (Generic Interfaces)?
- איך משתמשים בפרמטרים גנריים בממשקים?

Generics - תרגול

- צרו Interface גנרי בשם IList.
- תנו לו את המאפיינים:
`add(item: T): void`
`show(): void`
- צרו את מחלקת List אשר תממש ותטמיע את הממשק IList.
- תנו למחלקה מאפיין פרטי בשם list אשר יהיה מערך של T.
- המתודה add תוסיף איבר למערך ה-list.
- המתודה תדפיס לקונסול את הרשימה בצורה מאורגנת.
- נסו ליצור מופעים שונים של מחלקת List על ידי שימוש בטיפוס שונה בכל פעם.
- הוסיפו איברים לרשימות של המופעים.
- הדפיסו את הרשימות לקונסול.
- צרו מחלקה חדשה לבחירתכם וצרו 3 מופעים של המחלקה.
- צרו רשימה של המופעים באמצעות שימוש במחלקה List והדפיסו לקונסול.



סיכום שיעור

תודה על ההקשבה

אני וצוות המכללה כאן עבורכם