סיכום שיעור

מרצה: שלמה ספוזניקוב

15/08/2023

חזרה קצרה על מה שלמדנו
ומענה לשאלות על משימת הבית

זמן שאלות

Modules

Modules

בחלק זה נלמד מגוון נושאים.
בסיום הנושא תוכלו לענות על השאלות הבאות:

- מה הוא מודול?
- מה תפקידם של מודולים ב-JavaScript?
- כיצד נוכל להגדיר מסמך js כמודול?
- מדוע שנרצה לעשות זאת?
- כיצד נוכל לייבא\לייצא מודול?
- למה נשתמש ב-export default?
- מה היתרון של השיטה?
- דוגמאות ושימושים נפוצים.

JS_Advanced_Modules1.html

JS_Advanced_Modules2.html

JS_Advanced_Modules.js

הגדרת הסקריפט כמודול

ייבוא מידע

```
<script type="module">

    import { name, sayHola, changeName } from './script/JS_Advanced_Modules.js';

    alert(name);

    sayHola();

    changeName("Gal Lavi");

    alert(name);

</script>
```

קובץ JS הכולל **ייצא** מידע

```
let name = "Israel Israeli";

function sayHola(){
    alert("Hola class");
}

function changeName(newName) {
    name = newName;
    alert("name changed successfully");
}


export { name, sayHola, changeName }
```

תרגיל 1

- יש ליצור קובץ JS ובו פונקציה שמדפיסה את התוצאה של 5*5
- יש ליצור קובץ JS נוסף שתפעיל את הפונקציה הקודמת

תרגיל 2

- יש ליצור 2 inputs וכפתור בקובץ html
- יש ליצור קובץ js שבו יהיה כל הקוד שקשור ל-jsdom כמו לדוגמה addEventListener
- יש ליצור קובץ js נוסף שבוא פונקצייה שמחברת 2 מספרים
- לאחר פעולת החיבור יש להציג את התוצאה בדף ה-html

תרגיל 3

- יש ליצור 2 inputs וכפתור בקובץ html
- יש ליצור קובץ js שבו יהיה כל הקוד שקשור ל-jsdom כמו לדוגמה addEventListener
- יש ליצור קובץ js נוסף שבוא פונקצייה שמחסר 2 מספרים
- לאחר פעולת החיבור יש להציג את התוצאה בדף ה-html

תרגיל 4

- יש ליצור 2 inputs ו 4 כפתור בקובץ html
- יש ליצור קובץ js שבו יהיה כל הקוד שקשור ל-jsdom כמו לדוגמה addEventListener
- יש ליצור קובץ js נוסף שבוא פונקציות לחיבור/חיסור/כפל/חילוק 2 מספרים
- לאחר פעולת החיבור יש להציג את התוצאה בדף ה-html
- * יש ליצור פרוייקט חדש אם XO
- הפרוייקט כולל אפשרות לשחק נגד המחשב
- יש לחלק את הפרוייקט לקבצים
- לדוגמה הקוד שקשור למשחק נגד המחשב בקובץ אחד
- כל מה שקשור ל-dom לקובץ ניפרד וכו

JavaScript OOP

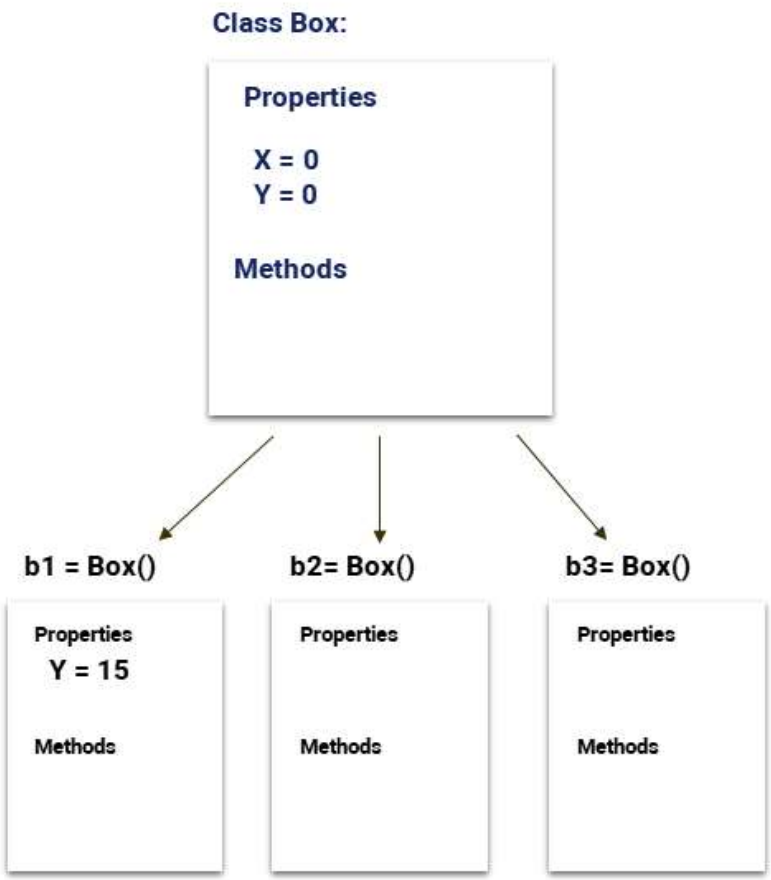
במסגרת הנושא נלמד את עקרונות תכנות מונחה עצמים, נתנסה בבניית מחלקות ויצירת אובייקטים.

מבוא לתכנות מונחה עצמים

מבוא

המחלקה היא למעשה התוכנית הבסיסית לאובייקטים שייווצרו ממנה האובייקט שנוצר מהמחלקה מקבל את מאפייני המחלקה.

- מהי מחלקה וכיצד ניתן להגדיר מחלקה?
- מהם אובייקטים וכיצד ניתן ליצר אובייקטים?
- מהם מאפיינים?
- מהם מתודות?



מילון מונחים

לפניכם מילון מנחים שיסייע לכם להבין את הנושאים בהמשך.

- **מחלקה** - מגדירה את המאפיינים מתודות (שיטות, תכונות) שיהיו לאובייקט.
- **אובייקט** –מופע פרטי של המחלקה, האובייקט יכיל את כל המאפיינים והמתודות של המחלקה.
- **מאפיין** - משתנה פרטי של המחלקה / אובייקט.
- **מתודה** - פונקציה פרטית של המחלקה / אובייקט.
- **בַּנאי** – קטע קוד המבוצע אוטומטית בעת יצירת האובייקט.
- **מילת הייחוס this** – מתייחסת לאובייקט עליו מדובר.

הכרות עם מחלקות ואובייקטים

המחלקה היא למעשה התוכנית הבסיסית לאובייקטים שייווצרו ממנה
האובייקט שנוצר מהמחלקה מקבל את מאפייני המחלקה אך יכול לדרוס אותם למשהו ספציפי יותר.

JS_OOP_Class_and_Objects.html

- מהי מחלקה וכיצד ניתן להגדיר מחלקה?
- מהם אובייקטים וכיצד ניתן ליצר אובייקטים?
- מהם מאפיינים של מחלקה?
- מהם מאפיינים של אובייקט?
- כיצד אובייקט יכול לדרוס (לא למחוק) מאפייני המחלקה?

```
class Person {
    first = "Israel";
    last = "Israeli";
    age = 0;
}

let p1 = new Person();
console.log( p1.first , p1.last); // Israel Israeli

p1.first = "Gal";
p1.last = "Lavi";
console.log( p1.first , p1.last); // Gal Lavi

let p2 = new Person();
p2.first = "Dana";
console.log( p2.first , p2.last); // Dana Israeli
```

הגדרה של מחלקה

יצירת אובייקט מהמחלקה

יצירת אובייקט נוסף מהמחלקה

תרגול מחלקות ואובייקטים

צרו קובץ חדש בשם **oop_class_person** לטובת הנושא ופתרו את התרגילים לפי הסדר.
חשוב להדפיס הודעות הצלחה ושגיאה למשתמש וכמובן לבדוק כל תרגיל בסיום.

תרגיל	תיאור המשימה
Ex-1	צרו את המחלקה **Person** הכוללת 3 מאפיינים: [first] [last] [age] והגדירו בהם ערכים. צרו משתנה חדש בשם p1 ושימו בתוכו מופע של המחלקת Person הדפיסו בקונסולה את סוג המשתנה p1 הדפיסו בקונסולה את ערכי המאפיינים של האובייקט p1
Ex-2	שנו את ערכי המאפיינים [first] [last] של p1 והגדירו בהם ערכים חדשים. הדפיסו בקונסולה את ערכי המאפיינים של האובייקט p1
Ex-3	צרו משתנה חדש בשם p2 ושימו בתוכו מופע של המחלקת Person הדפיסו בקונסולה את ערכי המאפיינים של האובייקט p2
Ex-4	שנו את ערכי המאפיינים [first] [last] [age] של p2 והגדירו בהם ערכים חדשים. הדפיסו בקונסולה את ערכי המאפיינים של האובייקט p2
Ex-5	צרו משתנה חדש בשם p3 ושימו בתוכו מופע של המחלקת Person שנו את ערכי המאפיינים [first] [last] [age] של p3 והגדירו בהם ערכים חדשים. הדפיסו בקונסולה את ערכי המאפיינים של האובייקט p3

Person

first	"Gal"
last	"Lavi"
age	30

מאפיינים של מחלקה ואובייקט

הכרות עם מאפיינים

לאחר יצירת האובייקט מהמחלקה ניתן ליצור מאפיינים ייחודיים לאובייקט אשר ידרסו את מאפייני המחלקה.
כמו כן ניתן להוסיף מאפיינים לגשת למאפיינים ולמחוק מאפיינים מאובייקט.

- כיצד לגשת למאפייני האובייקט?
- כיצד ליצר מאפיינים ייחודיים לאובייקט (ע"י דריסה)?
- כיצד להוסיף מאפיינים ייחודיים לאובייקט?
- כיצד ניתן למחוק מאפיינים ייחודיים לאובייקט?

JS_OOP_Properties.html

```
class Box {
    x = 0;
    y = 0;
}

// create object from class
let box1 = new Box();
console.log( box1.x );
console.log( typeof box1.x );

const num1 = box1.x; // get info;
console.log(num1); // 0

box1.y = 15; // set info;
console.log(box1.y); // 15

box1.z = 10; // add info;
console.log(box1.z); // 10

delete box1.z; // del info;
console.log(box1.z); // undefined

delete box1.y; // del info;
console.log(box1.y); // undefined
```

מאפיינים של מחלקה

פנייה למאפיין דרך האובייקט

שינוי של ערך המאפיין באובייקט

הוספה של מאפיין לאובייקט

מחיקה של מאפיין באובייקט

מתודות של מחלקה ואובייקט

הכרות עם מתודות

ניתן להגדיר במחלקה מתודות (פונקציות) אשר יוכלו לתפעל את האובייקט שייווצר מהמחלקה.
השימוש המילה this מתאר את האובייקט שייווצר מהמחלקה.

- מהן מתודות?
- כיצד ניתן להגדיר מתודות?
- מהי המילה this ולמה מתייחסת?
- כיצד האובייקט יכול להפעיל מתודות?

JS_OOP_Mehods.html

```
class Box {

    width = 100;
    height = 200;

    setInfo(w,h){
        this.width = w;
        this.height = h;
    }
    printInfo(){
        alert("Box size: width=" + this.width + " height=" + this.height);
    }

}

// create object from class
let box1 = new Box();
box1.printInfo(); // Box size: width=100 height=200

box1.width = 300;
box1.printInfo(); // Box size: width=300 height=200

box1.setInfo(150,250);
box1.printInfo(); // Box size: width=150 height=250
```

מתודה של המחלקה

מתודה של המחלקה

הפעלת מתודה לאחר יצירת האובייקט

הפעלת מתודה לאחר יצירת האובייקט

תרגול מתודות

צרו קובץ חדש בשם **oop_simple_user** לטובת הנושא ופתרו את התרגילים לפי הסדר חשוב להדפיס הודעות הצלחה ושגיאה למשתמש וכמובן לבדוק כל תרגיל בסיום.

תרגיל	תיאור המשימה
Ex-1	צרו את המחלקה **Simple_User** הכוללת3 מאפיינים: [first] [last] [age] והגדירו בהם ערכים. צרו משתנה חדש בשם p1 ושימו בתוכו מופע של המחלקח Simple_User הדפיסו בקונסולה את סוג המשתנה p1 הדפיסו בקונסולה את ערכי המאפיינים של האובייקט p1
Ex-2	**sayWelcome** הוסיפו מתודה שאומרת "Welcome" + full_name בסיום צרו אובייקטים חדשים מהמחלקה ובדקו שהמתודה עובדת.
Ex-3	**changeName** הוסיפו מתודה שמבקשת שם פרטי ושם משפחה ומעדכנת את המידע באובייקט. בסיום צרו אובייקטים חדשים ובדקו שהמתודה עובדת.

Simple_User

first	"Gal"
last	"Lavi"
age	30
sayWelcome()	
changeName(first,last)	

הכרות עם constructor

הכרות עם constructor

ה constructor היא מתודה שנקראת לפעולה אוטומטית בעת יצירת האובייקט.
ה constructor מאפשר לנו לאתחל את נתוני האובייקט בעת היצירה.

- מהו constructor ?
- מתי מופעל ה constructor?
- כמה בנאים (constructor) ניתן להגדיר למחלקה?
- למה משמש ה constructor ולמה הוא חשוב?
- כיצד ניתן להגדיר constructor למחלקה?
- כיצד מופעל ה constructor ביצירת האובייקט?

JS_OOP_Constructor.html

```
class Box {

    constructor(w,h){
        this.width = w;
        this.height = h;
    }

    setInfo(w,h){
        this.width = w;
        this.height = h;
    }

    printInfo(){
        alert("Box size: width=" + this.width + " height=" + this.height);
    }

}

// create object from class
let box1 = new Box(200, 300);
box1.printInfo(); // Box size: width=200 height=300

box1.width = 300;
box1.printInfo(); // Box size: width=300 height=300
```

הוספת constructor למחלקה

הפעלת ה constructor בעת יצירת האובייקט

תרגול constructor

צרו קובץ חדש בשם **oop_basic_user** לטובת הנושא ופתרו את התרגילים לפי הסדר חשוב להדפיס הודעות הצלחה ושגיאה למשתמש וכמובן לבדוק כל תרגיל בסיום.

BasicUser

first	"Gal"
last	"Lavi"
username	"gallavi"
password	"gal123"
Is_login	False
sayWelcome()	
changeName(first,last)	
login(u, p)	
Logout()	
changePassword(old,new)	

תרגיל	תיאור המשימה
Ex-1	צרו את המחלקה **BasicUser** הכוללת את מאפיינים: [first] [last][username][password] [is_login] הגדירו **constructor** המחייב לקבל [first] [last] [username] [password] צרו משתנה חדש בשם p1 ושימו בתוכו מופע של המחלקת Person תוך שימוש בהפעלת ה constructor
Ex-2	**sayWelcome** הוסיפו מתודה שאומרת "שלום" + full_name בסיום צרו אובייקטים חדשים מהמחלקה ובדקו שהמתודה עובדת.
Ex-3	**changeName** הוסיפו מתודה שמבקשת שם פרטי ושם משפחה ומעדכנת את פרטי הפרטים באובייקט. בסיום צרו אובייקטים חדשים ובדקו שהמתודה עובדת.
Ex-4	**login** הוסיפו מתודה המבקשת שם משתמש וסיסמה, במידה ושם המשתמש וסיסמה תואמים את username ו password, יש לשנות את מצב is_login ל – True
Ex-5	**logout** הוסיפו מתודה המשנה את מצב is_login ל-False
Ex-6	**change_password** הוסיפו מתודה המבקשת סיסמה ישנה (לזיהוי) וסיסמה חדשה (לשינוי) במידה וסיסמת הזיהוי תואמת את password, יש לשנות את password לסיסמה החדשה.

מתודות ומאפיינים סטטיים

הכרות מאפיינים סטטיים

השפה מאפשרת לנו ליצור מאפיינים ומתודות שהגישה אליהם ניתנת ישירות מהחלקה.
הגישה למאפיינים ומתודות סטטיות מתאפשרת ללא צורך ליצור אובייקט חדש.

JS_OOP_Static_Properties.html

```
class User {
    static age = 18;
    static country = "Israel";
    mail = "mail@gmail.com";
    password = "123456"
}

alert(User.country); // Israel
alert(User.password); // undefined

// create new static property
User.city = "Tel-Aviv";
alert(User.city)

// delete static property
alert(User.age); // 18
delete User.age;
alert(User.age); // undefined
```

יצירת מאפיין סטטי

פנייה למאפיין סטטי ישירות מהמחלקה

הוספת מאפיין סטטי למחלקה (מחוץ לבלוק המחלקה)

מחיקת מאפיין סטטי למחלקה (מחוץ לבלוק המחלקה)

- מהו מאפיין סטטי?
- מה ההבדל בין מאפיין סטטי למאפיין פרטי?
- מה היתרון של מאפיין סטטי?
- מתי נשתמש במאפיין סטטי?
- כיצד לגשת למאפיינים סטטיים?
- כיצד ניתן ליצור מאפיינים סטטיים מחוץ לאזור המחלקה?
- כיצד ניתן למחוק מאפיינים סטטיים מחוץ לאזור המחלקה?

הכרות עם מתודות סטטיות

השפה מאפשרת לנו ליצור מאפיינים ומתודות שהגישה אליהם ניתנת ישירות מהחלקה.
הגישה למאפיינים ומתודות סטטיות מתאפשרת ללא צורך ליצור אובייקט חדש.

JS_OOP_Static_Methods.html

- מהן מתודות סטטיות?
- מה ההבדל בין מתודות סטטיות מתודות פרטיות?
- מה הייתרון של מתודות סטטיות?
- מתי נשתמש במתודות סטטיות?
- כיצד לגשת למתודות סטטיות?

```
class Box {

    width = 100;
    height = 200;

    setInfo(w,h){
        this.width = w;
        this.height = h;
    }
    getInfo(){
        alert("Box size: width=" + this.width + " height=" + this.height);
    }

    static printInfo(){
        alert("default Box size: width=100 height=200");
    }

}

Box.printInfo();

let box1 = new Box();
box1.setInfo(500, 400)
box1.getInfo();
```

יצירת מתודה סטטית

הפעלת מתודה סטטית ישירות מהמחלקה

מתודות ומאפיינים פרטיים

הכרות מאפיינים פרטיים

השפה מאפשרת לנו ליצור מאפיינים ומתודות שהגישה אליהם ניתנת רק בתוך המחלקה.
הגישה למאפיינים ומתודות פרטיות שמורה ומאפשרות להגן על מידע.

- מהו מאפיין פרטי?
- מה ההבדל בין מאפיין פרטי למאפיין ציבורי?
- מה הייתרון של מאפיין פרטי?
- מתי נשתמש במאפיין פרטי?
- כיצד לגשת למאפיינים פרטיים?
- כיצד ניתן ליצור מאפיינים פרטיים?
- מהו המונח "כימוס" וכיצד הוא קשור לנושא?

JS_OOP_Static_Properties.html

```
class User {

    #username;
    #password;

    constructor( uName, uPass) {
        this.#username = uName;
        this.#password = uPass;
        this.#printInfo()
    }
    changePassword(oldPass, newPass){
        if( this.#password == oldPass ){
            this.#password = newPass;
            alert("Changed successfully");
            return false;
        }else{
            alert("Changed NOT successfully")
            return false;
        }
        this.#printInfo()
    }
    #printInfo() {
        alert("username = " + this.#username);
        alert("password = " + this.#password);
    }
}

let u1 = new User("Gallavi","123456");
u1.changePassword("11111", "22222");
u1.changePassword("123456", "78910");

console.log(u1.username) // undefined
//console.log(u1.#username); // Error: Private field
```

יצירת מאפיין פרטי

יצירת מתודה פרטית

לא ניתן לגשת למאפיין פרטי מחוץ למחלקה

ירושה של מחלקות ואובייקטים

הכרות עם ירושת מחלקות

נלמד שניתן ליצר מחלקות יורשות ממחלקות אחרות ובכך ליצר סדר בקוד ולמנוע קוד ספגטי.
האובייקט שייווצר מהמחלקה היורשת מקבלת את המתודות והמאפיינים של 2 המחלקות

- נלמד מהי מחלקה יורשת?
- נלמד למה משמשת מחלקה יורשת?
- נלמד כיצד להגדיר מחלקה יורשת?

JS_OOP_Inherit.html

```
class Person {

    setName(first, last){
        this.first = first;
        this.last = last;
    }

    printName(){
        alert(this.first + " " + this.last);
    }

}

class User extends Person{

    setInfo(username, password){
        this.username = username;
        this.password = password;
    }

    printInfo(){
        alert( "USER: " + this.username + ", PASSWORD: " + this.password);
    }

}

let u1 = new User("Gal", "Lavi",  )
u1.setName("Gal", "Lavi");
u1.setInfo("gall","123456");
u1.printName();
u1.printInfo();
```

מחלקת Person

מחלקת User יורשת את מחלקת Person

הכרות עם ירושת מחלקות עם constructor

נלמד שניתן ליצר מחלקות יורשות ממחלקות אחרות ובכך ליצר סדר בקוד ולמנוע קוד ספגטי.
האובייקט שייווצר מהמחלקה היורשת מקבלת את המתודות והמאפיינים של 2 המחלקות

- נלמד להפעיל את ה constructor של מחלקת האב במחלקה היורשת
- נלמד למה משמשת המילה super

JS_OOP_Inherit_Constructor.html

```
class Person {

    constructor(first, last){
        this.first = first;
        this.last = last;
    }

    printName(){
        alert(this.first + " " + this.last);
    }

}

class User extends Person{

    constructor(first, last, username, password){
        super(first, last)
        this.username = username;
        this.password = password;
    }

    printInfo(){
        alert( "USER: " + this.username + ", PASSWORD: " + this.password);
        alert( "USER: " + this.username + ", PASSWORD: " + this.password);
    }

}

let u1 = new User("Gal", "Lavi", "gall","123456" )
u1.printName();
u1.printInfo();
```

מחלקת User יורשת את מחלקת Person

הפעלת ה constructor של מחלקת האב

תרגול ירושת מחלקות עם constructor

צרו קובץ חדש בשם **oop_Inherit_Class** לטובת הנושא ופתרו על התרגילים לפי הסדר חשוב להדפיס הודעות הצלחה ושגיאה למשתמש וכמובן לבדוק כל תרגיל בסיום.

תרגיל	תיאור המשימה
Ex-1	צרו את המחלקה **Person** הכוללת את מאפיינים: [first] [last] [age] [city]
Ex-2	הגדירו **constructor** המחייב [first] [last] [age] [city] צרו משתנה חדש בשם p1 ושימו בתוכו מופע של המחלקת Person תוך שימוש בהפעלת ה constructor
Ex-3	**print** הוסיפו מתודה שתדפיס בקונסולה את נתוני האובייקט. (המתודה תשמש אותנו לצורכי בדיקה)
Ex-4	**sayWelcome** הוסיפו מתודה שאומרת "שלום" + full_name בסיום צרו אובייקטים חדשים מהמחלקה ובדקו שהמתודה עובדת.
Ex-5	**setName** הוסיפו מתודה שמבקשת שם פרטי ושם משפחה ומעדכנת את פרטי הפרטים באובייקט. בסיום צרו אובייקטים חדשים ובדקו שהמתודה עובדת.

Person

first	"Gal"
last	"Lavi"
age	30
city	"Tel-Aviv"
constructor()	
print()	
sayWelcome()	
setName(first,last)	

תרגול ירושת מחלקות עם constructor

באותו הקובץ שפתחנו **oop_Inherit_Class** לטובת הנושא יש להמשיך לענות על התרגילים חשוב להדפיס הודעות הצלחה ושגיאה למשתמש וכמובן לבדוק כל תרגיל בסיום.

User

Person	
username	"gallavi"
password	"gal123"
Is_login	False
print()	
login(u, p)	
logout()	
setPassword(old,new)	

תרגיל	תיאור המשימה
Ex-1	צרו את המחלקה **User** הכוללת את מאפיינים:[username] [password] [is_login] אשר יורשת ממחלקת Person
Ex-2	הגדירו **constructor** המחייב לקבל [username] [password] ומאתחלת את נתוני האובייקט. העזרו ב super כדיי להפעיל את ה constructor של מחלקת Person.
Ex-3	צרו משתנה חדש בשם u1 ושימו בתוכו מופע של המחלקת User תוך שימוש בהפעלת ה constructor נסו להפעיל את המתודה sayWelcome של מחלקת Person ובדקו את תגובת האובייקט.
Ex-4	**login** הוסיפו מתודה המבקשת שם משתמש וסיסמה, במידה ושם המשתמש וסיסמה תואמים את username ו password, יש לשנות את מצב is_login ל – True
Ex-5	**logout** הוסיפו מתודה המשנה את מצב is_login ל-False
Ex-6	**setPassword** הוסיפו מתודה המבקשת סיסמה ישנה (לזיהוי) וסיסמה חדשה (לשינוי) במידה וסיסמת הזיהוי תואמת את password, יש לשנות את password לסיסמה החדשה.
Ex-7	**print** הוסיפו מתודה אש תדפיס בקונסולה את נתוני האובייקט – המתודה **דורסת** את מתודה print של Person. (המתודה תשמש אותנו לצורכי בדיקה).

תרגילים נוספים

תרגיל פילוח ציוני סטודנטים

תרגיל פילוח ציוני סטודנטים

צרו קובץ חדש בשם **oop_students** לטובת הנושא ופתרו את התרגילים לפי הסדר חשוב להדפיס הודעות הצלחה ושגיאה למשתמש וכמובן לבדוק כל תרגיל בסיום.

תרגיל	תיאור המשימה
Class	הגדירו מחלקה חדשה בשם **Class** אשר תכלול מאפיין אחד אשר יכיל מערך של מספרים המשקפים ציוני סטודנטים בבחינה, הוסיפו constructor אשר מקבל מערך ומאתחל את הציונים באובייקט.
print	הגדירו מתודה חדשה אשר תדפיס את המידע הבא: תכולת המערך, אורך המערך, סכום הציונים במערך, וממוצע הציונים במערך.
add	הגדירו מתודה חדשה המתודה תבקש מהמשתמש להקיש את כמות הציונים החדשים שצריך להוסיף. (לדוגמה 5 סטודנטים) עבור כל סטודנט המשתמש יתבקש להקיש ציון, כל הציונים התווספו למערך.
between	הגדירו מתודה חדשה המתודה תבקש מהמשתמש להקיש ציון מינימלי (לדוגמה: 70) וציון מקסימלי (לדוגמה: 90) המתודה תחזיר את כמות הסטודנטים בתוך הטווח.
data	הגדירו מתודה חדשה המתודה תדפיס בקונסולה את פילוח הסטודנטים בכיתה: כמות הסטודנטים הנכשלים (ציון 69 ומטה), כמות סטודנטים בשכבת אמצע (ציון בין 70-90) וכמות הסטודנטים המצטיינים (ציון 91 עד 100)
Start	הגדירו מתודה חדשה המתודה מבקשת מהמשתמש להזין אפשרות ביצוע. 1 - הדפס | 2 – הוסף | 3 – בדוק כמות בטווח | 4 – הצג פילוח נתונים. הפעילו את הפונקציה בהתאם לבחירת המשתמש.

StudentsClass

students_grades[]	
print()	
add()	
between()	
data()	
start()	
average()	
min()	
max()	

תרגיל פילוח ציוני סטודנטים

צרו קובץ חדש בשם **oop_students** לטובת הנושא ופתרו את התרגילים לפי הסדר חשוב להדפיס הודעות הצלחה ושגיאה למשתמש וכמובן לבדוק כל תרגיל בסיום.

תרגיל	תיאור המשימה
average	הגדירו מתודה המקבלת מערך (הכולל ציוני סטודנטים). המתודה מחזירה את **ממוצע הציונים**.
max	הגדירו מתודה המקבלת מערך (הכולל ציוני סטודנטים). המתודה מחזירה את **הציון הגבוה** ביותר.
min	הגדירו מתודה המקבלת מערך (הכולל ציוני סטודנטים). המתודה מחזירה את **הציון הנמוך** ביותר.
gap	הגדירו מתודה המקבלת מערך (הכולל ציוני סטודנטים). המתודה מחזירה את **פער הציונים** בין הציון הנמוך ביותר לציון הגבוה ביותר.
endPoints	הגדירו מתודה המקבלת מערך (הכולל ציוני סטודנטים). המתודה מחזירה מערך הכולל את ה**ציון הנמוך** ביותר ואת **הציון הגבוה** ביותר.

StudentsClass

students_grades[]	
print()	
add()	
between()	
data()	
start()	
average()	
min()	
max()	

תרגיל פילוח ציוני סטודנטים

צרו קובץ חדש בשם **oop_students** לטובת הנושא ופתרו את התרגילים לפי הסדר חשוב להדפיס הודעות הצלחה ושגיאה למשתמש וכמובן לבדוק כל תרגיל בסיום.

תרגיל	תיאור המשימה
Tops	הגדירו מתודה המקבלת מערך (הכולל ציוני סטודנטים). המתודה מחזירה את **כמות המצטיינים**. (מצטיין מוגדר ציון 90 ומעלה)
fail	הגדירו מתודה המקבלת מערך (הכולל ציוני סטודנטים). המתודה מחזירה את **כמות הנכשלים**. (נכשל מוגדר ציון 69 ומטה)
Middle	הגדירו מתודה המקבלת מערך (הכולל ציוני סטודנטים). המתודה מחזירה את **כמות הסטודנטים בשכבת אמצע**. (שכבת אמצע מוגדר ציון 70-89)
data	הגדירו מתודה המקבלת מערך (הכולל ציוני סטודנטים). המתודה מחזירה מערך הכולל ניתוח את **פילוח התוצאות בכיתה** **מערך כולל 3 מספרים המציגים** את כמות הסטודנטים המצטיינים, בשכבת אמצע, נכשלים.
search	הגדירו מתודה המקבלת מערך (שמות) ומשתנה המכיל שם. המתודה מחזירה true/false במידה והשם קיים במערך

StudentsClass

students_grades[]	
print()	
add()	
between()	
data()	
start()	
average()	
min()	
max()	

תרגיל ניהול משתמשים

תרגיל ניהול מערכת משתמשים

צרו קובץ חדש בשם **oop_users** לטובת הנושא ולענות על התרגילים לפי הסדר
חשוב להדפיס הודעות הצלחה ושגיאה למשתמש וכמובן לבדוק כל תרגיל בסיום.

Users

users[BasicUser]	
print()	
add()	
isUser(u,p)	
del()	

BasicUser

first	"Gal"
last	"Lavi"
username	"gallavi"
password	"gal123"
is_login	False
sayWelcome()	
setName(first,last)	
login(u, p)	
Logout()	
changePassword(old,new)	

תרגיל ניהול הזמנה
מתפריט מסעדה פשוט

תרגיל ניהול הזמנה מתפריט מסעדה פשוט

צרו קובץ חדש בשם **oop_cart_simple_menu** לטובת הנושא ופתרו את התרגילים לפי הסדר חשוב להדפיס הודעות הצלחה ושגיאה למשתמש וכמובן לבדוק כל תרגיל בסיום.

Cart

products[**Product**]	
print()	
startOption()	
add()	
del()	
totalPrice()	

simple_menu[Product]

0	{ id:1, name: "Salad", price: 22.5 }
1	{ id:2, name: "Pasta", price: 35 }
2	{ id:3, name: "Pizza", price: 54 }
3	{ id:4, name: "Burger", price: 45 }

תרגיל ניהול הזמנה
מתפריט מסעדה חכם

תרגיל ניהול הזמנה מתפריט מסעדה חכם

צרו קובץ חדש בשם **oop_cart_menu** לטובת הנושא ופתרו את התרגילים לפי הסדר חשוב להדפיס הודעות הצלחה ושגיאה למשתמש וכמובן לבדוק כל תרגיל בסיום.

Cart

products[**Product**]	
print()	
startOption()	
add()	
del()	
totalPrice()	

Menu

products[**Product**]	
print()	
startOption()	
add(product)	
del(id_product)	
printMenu()	

Product

id	
Name	
Price	

תרגיל ניהול משלוח ממלאי מוצרים פשוט

תרגיל ניהול משלוח ממלאי מוצרים פשוט

צרו קובץ חדש בשם **oop_cart_simple_stock** לטובת הנושא ופתרו את התרגילים לפי הסדר חשוב להדפיס הודעות הצלחה ושגיאה למשתמש וכמובן לבדוק כל תרגיל בסיום.

Cart

products[]	
print()	
startOption()	
start()	
add()	
del()	
total()	

simple_stock[Product]

0	{CodeId:152254, name: "computer", price: 1500, weight: 1.2, inStock: 15, minStock: 5}
1	{CodeId:1364, name: "LG phone", price: 1080, weight: 0.2, inStock: 9, minStock: 10}
2	{CodeId:889254, name: "samsong phone", price: 980, weight: 0.3, inStock: 26, minStock: 10}
3	{CodeId:72138, name: "LG phone min", price: 980, weight: 0.1, inStock: 19, minStock: 10}

תרגיל ניהול משלוח ממלאי מוצרים חכם

תרגיל ניהול משלוח ממלאי מוצרים מתקדם

צרו קובץ חדש בשם **oop_cart_stock** לטובת הנושא ופתרו את התרגילים לפי הסדר חשוב להדפיס הודעות הצלחה ושגיאה למשתמש וכמובן לבדוק כל תרגיל בסיום.

Cart

products[]	
print()	
startOption()	
start()	
add()	
del()	
totalPrice()	
totalWeight()	

Stock

products[]	
print()	
startOption()	
printInStock()	
printOutStock()	
Add()	
Del()	

Product

CodeId	152254
Name	"computer"
Price	1500
Weight	1.2
inStock	15
minStock	5

מילון מונחים

לפניכם מילון מנחים שיסייע לכם להבין את הנושאים בהמשך.

- **יְרוּשָה** - מחלקה יכולה לרשת תכונות ויכולות ממלקה אחרת.
- **כימוס** - טכניקה המאפשרת להגן על מאפיינים ומתודות ויוצרת קוד פשוט והמונע טעויות.
- **פולימורפיזם** – (רב צורתיות) אובייקטים ומחלקות יורשות יכולים לשנות את עצמם בהתאם לצרכיהם האישיים ובכך מאפשרים התנהגות שונה.
- **הַפְשָטָה** – יכולות התכנות המונחה עצמים מאפשרות לשקף מודל מציאותי של עצמים מורכבים בצורה פשוטה.

סיכום שיעור

אז מה למדנו היום?

- GITHUB
- Js modular
- oop

שיעורי בית

- להכין טבלה שמסכמת את החומרים שלמדנו
- שקופיות 15,20

תודה על ההקשבה

אני וצוות המכללה כאן עבורכם

לשלוח לי:

איך היה השיעור מ-1 עד 10

איך היה הקצב מ-1 עד 10

😄