

# סיכום שיעור

מרצה: שלמה ספוזניקוב

15/08/2023



## חזרה קצרה על מה שלמדנו ומענה לשאלות על משימת הבית



# זמן שאלות

# Modules



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 }
```

# Modules

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

- מה הוא מודול?
- מה תפקידים 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.

 **Live Server** 17ms  
Launch a development local Server with  
Ritwick Dey

# 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

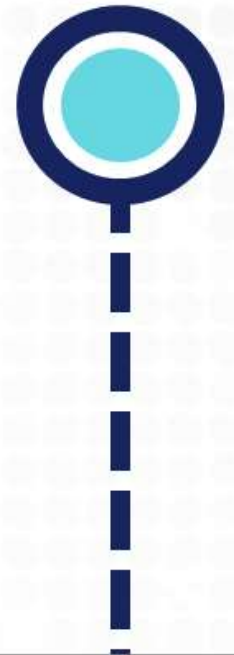


## תרגיל 4

- 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.



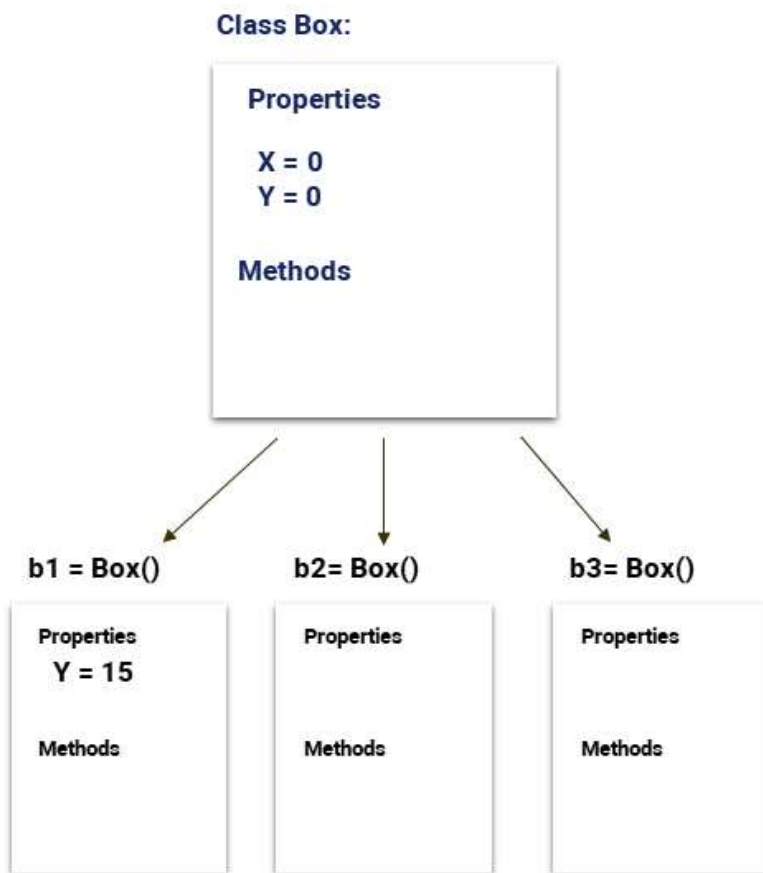
# מבוא לתכנות מונחה עצמים



# 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 יכול את כל המאפיינים והמתודות של המחלקה.
- **מאפיין** - משתנה פרטי של המחלקה / אובייקט.
- **מתודה** - פונקציה פרטית של המחלקה / אובייקט.
- **בנאי** - קטע קוד המבוצע אוטומטית בעת יצירת האובייקט.
- **מילת הייחוס this** - מתייחסת לאובייקט עליו מדובר.

# הכרות עם מחלקות ואובייקטים

JS\_OOP\_Class\_and\_Objects.html

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

```
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

- 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?



# 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.

## Person

first	"Gal"
last	"Lavi"
age	30

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

# מאפיינים של מחלקה ואובייקט







# הכרות עם מאפיינים

JS\_OOP\_Properties.html

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

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

מאפיינים של מחלקה

```
// create object from class
```

```
let box1 = new Box();  
console.log( box1.x );  
console.log( typeof box1.x );
```

פנייה למאפיין דרך  
האובייקט

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

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

שינוי של ערך  
המאפיין באובייקט

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

הוספה של מאפיין  
לאובייקט

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

מחיקה של מאפיין  
באובייקט

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

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

# מתודות של מחלקה ואובייקט



# הכרות עם מתודות

JS\_OOP\_Mehods.html

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

```
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);
  }
}
```

מתודה של המחלקה

מתודה של המחלקה

```
// 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
```

הפעלת מתודה לאחר  
יצירת האובייקט

הפעלת מתודה לאחר  
יצירת האובייקט

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

## Simple\_User

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

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



## 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

