

JS OOP

09/01/2025

JavaScript OOP

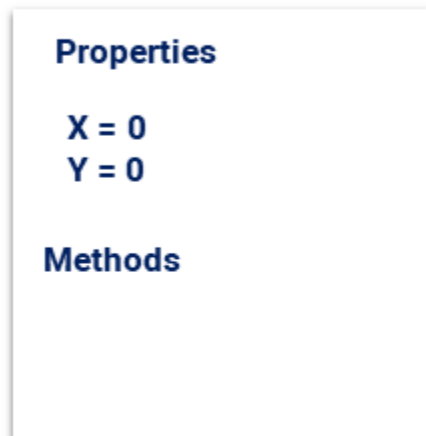
במסגרת הנושא נלמד את עקרונות תכנות מונחה עצמים, נתנסה בבניית מחלקות ויצירת אובייקטים.



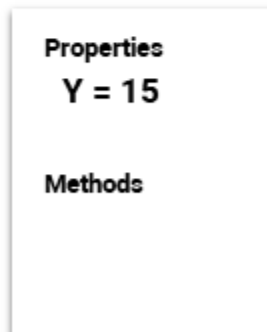
מבוא לתכנות מונחה עצמים



Class Box:



b1 = Box()



b2 = Box()



b3 = Box()



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

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

הגדרה של מחלקה

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

יצירת אובייקט מהמחלקה

```
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
```

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 Person 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



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



JS_OOP_Mehods.html

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

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.

Simple_User



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

תרגיל	תיאור המשימה
Ex-1	צרו את המחלקה Simple_User 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	sayWelcome הוסיפו מתודה שאומרת "Welcome" + first + last 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.

הכרות עם constructor



הכרות עם constructor

JS_OOP_Constructor.html

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

```
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);
  }
}
```

הוספת constructor למחלקה

```
// 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
```

הפעלת constructor בעת יצירת האובייקט

- 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?

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" + שם פרטי + שם משפחה. 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.

תודה על ההקשבה

אני וצוות המכללה כאן עבורכם