JS OOP

13/01/2025

מתודות ומאפיינים סטטיים

כל הזכויות שמורות ©

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?

כל הזכויות שמורות ©

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

כל הזכויות שמורות ©

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

כל הזכויות שמורות ©

ירושה של מחלקות ואובייקטים

כל הזכויות שמורות ©

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

JS_OOP_Inherit.html

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

```
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

כל הזכויות שמורות ©

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

JS_OOP_Inherit_Constructor.html

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

```
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

כל הזכויות שמורות ©

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.

Person

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

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.

כל הזכויות שמורות ©

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
constructor(user, pass)	
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 תדפיס רק שם משתמש והאם מחובר).

תרגיל פילוח ציוני סטודנטים

כל הזכויות שמורות ©

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 **StudentsClass** אשר תכלול מאפיין אחד אשר יכיל מערך של מספרים המשקפים ציוני סטודנטים בבחינה, הוסיפו constructor אשר מקבל מערך ומאתחל את הציונים באובייקט.
printInfo	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()	

כל הזכויות שמורות ©

תרגיל פילוח ציוני סטודנטים

המשיכו עם הקובץ הקודם 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()	

כל הזכויות שמורות ©

תרגיל פילוח ציוני סטודנטים

המשיכו עם הקובץ הקודם 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 מספר המכיל ציון. המתודה מחזירה true/false במידה הציון קיים במערך.

StudentsClass

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

כל הזכויות שמורות ©

תודה על ההקשבה

אני וצוות המכללה כאן עבורכם