

סיכום שיעור

מרצה: שלמה ספוזניקוב

06/08/2023



חזרה קצרה על מה שלמדנו ומענה לשאלות על משימת הבית



זמן שאלות

אופרטור Spread





אופרטור Spread

es6_spread_params

בחלק זה נכיר אופרטור אשר מאפשר לבצע שימוש נוח ויעיל באובייקטים.
בסיום הנושא תוכלו לענות על השאלות הבאות:

```
let myArray1 = [11, 22, 33, 44, 55];  
function print(num1, ...myArr){  
  console.log(num1);  
  console.log(myArr)  
}
```

פונקציה המקבלת
rest parameter

```
print(100, 10, 20, 10, 30);
```

קריאה לפונקציה
rest parameter

```
print(100, myArray1);
```

פונקציה המקבלת
rest parameter

- מה תפקידו של אופרטור spread?
- איך מגדירים אופרטור spread (...)?
- אילו שימושים נפוצים אנו מכירים?
- מה זה rest parameter?
- האם משנה מיקומו של ה-rest parameter בסדר מיקומי הארגומנטים הנשלחים לפונקציה?
- מה הם היתרונות בשימוש?

תרגילון

- יש ליצור פונקציה שמוצאת את הפרמטר הגדול ביותר

פונקציות חץ - *arrow*





JS_Advanced_Arrow_Function.html

```
function plus1(num1, num2) {  
  return num1 + num2;  
}
```

```
let plus2 = (num1, num2) => num1 + num2;
```

```
console.log( plus1(10,20) ); // 30  
console.log( plus2(10,20) ); // 30
```

```
let sayHola = (name) => "Hola " + name;
```

```
console.log( sayHola("Gal") ); // "Hola Gal"
```

פונקציות חץ - Arrow

בחלק זה נלמד ליצור פונקציות באמצעות שיטה נוספת לשיטות שהכרנו קודם.
בסיום הנושא תוכלו לענות על השאלות הבאות:

- מה היא פונקציית חץ? (arrow function)
- איך נוכל להגדיר פונקציית חץ?
- מה היתרון בשימוש בפונקציות חץ?
- מה ההבדל בין פונקציית חץ לפונקציה רגילה?

Object/array destruction

es6_destruct

Object destruction

```
Let elm = document.querySelector("input");
```

```
Let {value} = elm
```

```
// same as
```

```
// let value = elm.value
```

- כאשר יש לנו את הצורך למשוך מאפיין מסויים לתוך משתנה כדי שיהיה יותר נוח. נשתמש ב-destruction
- צורה זאת יוצרת משתנה במקום שאנחנו ניצור משתנה חדש
- לדוגמה כאשר נירצה להשתמש במה שהקליד המשתמש בinput אנחנו יכולים להשתמש בdestruction כמו בדוגמה.

Object destruction

```
Let elm = document.querySelector("input");  
Let {value:myVal} = elm  
// same as  
// let myVal = elm.value
```

• אם קיים הצורך להשתמש בdestruct
אבל אנו רוצים לתת שם אחר אז
נכתוב כמו בדוגמה

Array destruction

```
Let arr = ["Kenny", "mc", "lenny"]
```

```
Let [name, last] = arr
```

```
//same as
```

```
// let name = arr[0]
```

```
// let last = arr[1]
```

- כמו באובייקטים קיימת אותה האפשרות במערכים
- אבל במערכים בdestruct פועל בהתאם לאינדקסים

שימוש ב `try & catch`





שימוש ב try & catch

JS_Advanced_Try_Catch.html

בחלק זה נלמד כיצד לטפל ולתפוס בעיות בקוד ולטפל בהן, מה שיאפשר לקוד להמשיך לרוץ. בסיום הנושא תוכלו לענות על השאלות הבאות:

- ממה יכולות להיגרם שגיאות ב-JS?
- אילו שגיאות יכולות להתרחש בזמן ריצת התוכנית שלא ניתן לצפות מראש?
- מה נוכל לעשות על מנת לטפל בשגיאות בזמן ריצה?
- מה הוא בלוק ה-try catch?
- מה נוכל לעשות במקרה שתפסנו שגיאה באמצעות בלוק try-catch?
- מה תפקידו של finally?

```
const text = "Hola Class";
```

```
// demo 1
```

```
try{
```

```
  console.log(someText);
```

מנסה לבצע את הפעולה

```
}catch( error ){
```

```
  console.log(error);
```

קוד ירוץ במידה ותהיה שגיאה

```
  console.log(error.message);
```

```
}finally{
```

```
  console.log("Finish 1");
```

קוד ירוץ בכל מקרה בסיום

```
}
```

```
// demo 2
```

```
console.log(someText);
```

דוגמה לקוד שיוציא שגיאה

```
console.log("Finish 2");
```

✖ ▶ Uncaught ReferenceError: someText is not defined
at JS_Advanced_Try_Catch.html:21:21

פונקציות Callback





JS_Advanced_Callback_Functions.html

```
// Demo 1
document.getElementById('myButton1').addEventListener("click", function () {
    alert('Hola Class From myButton')
});
document.getElementById('myButton2').addEventListener("click", function () {
    alert('Hola Class From myButton')
});
```

פונקציית callback אנונימית
שתופעל בעת הפעלת האירוע

```
// Demo 2
function sayHola(){
    alert('Hola Class From myButton')
}
document.getElementById('myButton3').addEventListener("click", sayHola );
document.getElementById('myButton4').addEventListener("click", sayHola );
```

פונקציית callback שתופעל
בעת הפעלת האירוע

פונקציות Callback

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

- מה היא פונקציית callback?
- לאיזה שימושים נועדה?
- באיזה מקרים נעזר בפונקציית callback?

```
<button id="myButton1">click me 1</button>
<button id="myButton2">click me 2</button>
<button id="myButton3">click me 3</button>
<button id="myButton4">click me 4</button>
```




JS_Advanced_Callback_Functions.html

```
function sayThankYou() {  
    alert("Thank You");  
    alert("Welcome back again");  
}
```

פרמטר myFunc שמיועד
לשם פונקציה שתופעל

```
function totalPrice(price, products, myFunc) {  
    alert(price * products);  
    myFunc()  
}
```

הפעלת myFunc
שהתקבל כפרמטר

```
totalPrice(10, 5, sayThankYou)
```

שם הפונקציה
שתופעל בסיום הקוד

פונקציות Callback

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

- מה היא פונקציית callback?
- לאיזה שימושים נועדה?
- באיזה מקרים נעזר בפונקציית callback?

תרגילון

- יש ליצור פונקציה שיודעת לקבל כפרמטר פונקציה אחרת
- הפונקציות האחרות הן אחד מ:
 - פונקציה שמדפיסה hello
 - פונקציה שמדפיסה world
 - פונקציה שמדפיסה dog

תרגילון 2

- יש ליצור 4 פונקציות חיבור, חיסור, כפל וחילוק
- יש ליצור פונקציית מחשבון שהמתכנת יבחר איזה פונקציה להפעיל ואיזה מספרים לחשב

תרגיל

- יש ליצור קובץ html אם שם משתמש וסיסמה
- יש ליצור קובץ js
- בעזרת הכלים לשמדנו היום יש ליבדוק האם:
- שם המשתמש הוא "lenny"
- הסיסמה היא "a1234"
- אם כן אז להציג alert אתה יכול להיכנס
- אחרת להציג אתה לא יכול להיכנס



סיכום שיעור



אז מה למדנו היום?

- Arrow function
- Destructuring
- Try&catch



שיעורי בית

- יש ליצור פונקציה שבתוכה יש אובייקט של משתמש
 - יש להוצא ממנו 2 מאפיינים ולשנות להם את השם של המשתנה
 - יש להדפיס אותם
- יש ליצור פונקציה שמקבלת מספרים ומוצאת את המספר הקטן ביותר
- יש ליצור אתר אם כפתור שכאשר לוחצים עליו הרקע של האתר ישתנה לירוק

תודה על ההקשבה

אני וצוות המכללה כאן עבורכם

לשלוח לי:

איך היה השיעור מ-1 עד 10

איך היה הקצב מ-1 עד 10

