הנה סיכום מפורט של השיעור:

**שיעור 7 - ניהול אובייקטים ו-JavaScript DOM**

**יצירת אובייקטים ב-JavaScript**

1. **מהו אובייקט?**
   * אובייקט הוא מבנה נתונים המכיל זוגות של מפתחות וערכים (key-value pairs).
   * ניתן להגדיר אובייקטים באמצעות {} ולשייך להם תכונות (properties).
2. **דרכים ליצירת אובייקט**
   * ניתן ליצור אובייקט ישירות על ידי הצהרה:
   * const user = {
   * name: "John",
   * age: 25
   * };
   * ניתן להוסיף תכונות לאובייקט לאחר יצירתו:
   * const user = {};
   * user.name = "John";
   * user.age = 25;
3. **שימוש ב-JSON (JavaScript Object Notation)**
   * JSON הוא פורמט טקסטואלי לאחסון וניוד מידע.
   * ניתן להמיר אובייקט לפורמט JSON באמצעות JSON.stringify().
   * ניתן להמיר JSON חזרה לאובייקט באמצעות JSON.parse().
   * const jsonString = JSON.stringify(user);
   * console.log(jsonString); // הופך לאובייקט JSON
   * const userObj = JSON.parse(jsonString);
   * console.log(userObj); // הופך חזרה לאובייקט רגיל
4. **שימוש במערך של אובייקטים**
   * כאשר רוצים לנהל מספר משתמשים, ניתן לאחסן אותם במערך.
   * const users = [];
   * users.push({ name: "Alice", id: 1 });
   * users.push({ name: "Bob", id: 2 });
5. **מניעת דריסה של אובייקט (Reference vs Copy)**
   * כאשר מוסיפים אובייקט למערך, יש לוודא שיוצרים אובייקט חדש כל פעם כדי למנוע שינוי של הנתונים הקודמים.
   * const user1 = { name: "Alice" };
   * users.push({ ...user1 }); // יצירת עותק חדש במקום להפנות לאותו האובייקט

**בדיקות ופעולות על נתונים**

1. **בדיקת סוג הנתונים עם typeof**
   * typeof מחזיר את סוג הנתונים של משתנה:
   * console.log(typeof "hello"); // "string"
   * console.log(typeof 42); // "number"
   * console.log(typeof true); // "boolean"
   * console.log(typeof {}); // "object"
   * console.log(typeof []); // "object" (מערך הוא אובייקט)

**שימוש ב-DOM (Document Object Model)**

1. **מהו ה-DOM?**
   * ה-DOM הוא המודל המאפשר גישה ושינוי של אלמנטים בדף ה-HTML באמצעות JavaScript.
2. **יצירת אלמנטים וניהול אירועים ב-DOM**
   * אפשר לגשת לכל אלמנט בדף דרך document.getElementById() או document.querySelector().
   * ניתן להוסיף אירועים כמו onclick כדי לבצע פעולות בעת לחיצה על כפתור.
3. **יצירת כפתור שמוסיף משתמש ומדפיס אותו ב-console**
4. function handleClick() {
5. const user = {};
6. user.name = document.getElementById("userName").value;
7. user.id = document.getElementById("id").value;
8. user.phone = document.getElementById("phone").value;
9. console.log("Create new user: ", user);
10. }

**תרגיל - יצירת מערכת לניהול משתמשים**

**דרישות התרגיל:**

1. יצירת טופס לקליטת פרטי משתמש כולל:
   * שם פרטי (firstName)
   * שם משפחה (lastName)
   * מספר מזהה (id)
   * תאריך לידה (dob)
   * כתובת (כתובת פנימית הכוללת עיר, מיקוד, ומספר בית)
2. הוספת המשתמש למערך users.
3. יצירת כפתור שמציג את רשימת המשתמשים בקונסול (console.table()).

**מימוש התרגיל**

const users = [];

function createUser() {

// יצירת אובייקט משתמש

const user = {};

user.firstName = document.getElementById("firstName").value;

user.lastName = document.getElementById("lastName").value;

user.id = document.getElementById("id").value;

user.dob = document.getElementById("dob").value;

// יצירת אובייקט כתובת

const address = {};

address.city = document.getElementById("city").value;

address.zip = document.getElementById("zip").value;

address.number = document.getElementById("number").value;

user.address = address;

// הוספת המשתמש למערך

users.push(user);

// ניקוי שדות הטופס לאחר הוספת משתמש

document.querySelectorAll('input').forEach(input => input.value = "");

alert("User added successfully!");

}

function showUsers() {

console.table(users);

}

**שימוש ב-Local Storage לשמירת נתונים**

1. **מהו Local Storage?**
   * Local Storage מאפשר שמירת נתונים בדפדפן באופן מתמיד (גם לאחר רענון הדף).
   * ניתן לאחסן ולהוציא מידע באמצעות localStorage.setItem() ו- localStorage.getItem().
2. **שמירת נתונים ב-Local Storage**
3. localStorage.setItem("users", JSON.stringify(users));
4. **שליפת נתונים מ-Local Storage**
5. const storedUsers = JSON.parse(localStorage.getItem("users")) || [];
6. console.log(storedUsers);
7. **מחיקת נתונים מ-Local Storage**
8. localStorage.removeItem("users");

**Session Storage לעומת Local Storage**

| **תכונה** | **Local Storage** | **Session Storage** |
| --- | --- | --- |
| משך הזמן | נשמר לצמיתות | נמחק בעת סגירת הדפדפן |
| שימושים | שמירת נתוני משתמשים, הגדרות | שמירת מידע זמני במהלך הסשן |
| גישה לנתונים | localStorage.getItem() | sessionStorage.getItem() |

**שימוש ב-appendChild() להוספת אלמנטים**

* כדי להוסיף אלמנטים חדשים לדף נשתמש ב- document.createElement() ו- appendChild():
* const newDiv = document.createElement("div");
* newDiv.textContent = "Hello, World!";
* document.body.appendChild(newDiv);

**סיכום עיקרי**

* **יצירת אובייקטים ומערכים ב-JavaScript**
* **שימוש ב-JSON להמרת אובייקטים למחרוזות ולהפך**
* **גישה וניהול אלמנטים בדף בעזרת ה-DOM**
* **יצירת טפסים ואיסוף נתונים מהמשתמש**
* **שמירת נתונים ב-Local Storage ו-Session Storage**
* **יצירת כפתורים ואירועים דינמיים ב-JavaScript**

אם יש לך שאלות נוספות או אם אתה רוצה הרחבה על נושא מסוים, אני כאן לעזור! 😊