שמירת מידע ב Dom Storge

מתי יש צורך לאחסן נתונים?

נניח קיים אתר המאפשר למשתמש לבחור צבע רקע אהוב לאתר. המשתמש בוחר צבע ומעתה ואילך הצבע נשמר עבורו. בכל כניסה לאתר, יהיה צבע הרקע של האתר, בצבע שבחר המשתמש.

היכן נשמור את הצבע הנבחר?

אם נשמור במשתנה, הוא יפוג מיד עם סגירת הדפדפן, ואנו רוצים שגם בגלישה הבאה נכיר את הצבע הנבחר.

עד היום, אחסון מידע בדפדפן (ללא גישה לשרת!) נעשה באמצעות עוגיות (Cookie) .

Cookie – קבצי טקסט קטנים שנשתלים במחשב הלקוח.

המתכנת שומר מידע בעוגיה. כיון שהעוגיה הינה קובץ פיזי שנשמר בדיסק הקשיח, הקובץ נשמר גם לאחר סגירת הדפדפן (ואף לאחר סגירת המחשב).

בגלישה חוזרת, המתכנת ישלוף את המידע השמור בעוגיה, וישתמש בו לצרכיו. בדוגמא הנ"ל, יישלף הצבע הנבחר, והוא יקבע כצבע הרקע של האתר.

אולם, לאחסון המידע בעוגיות יש מגבלות:

- גודל - 4 **kb** בלבד
- ניתן לשמור רק מחרוזת טקסט, ולא ניתן לשמור גם מספרים וערכים בוליאניים, אלא אם עברו המרה לטקסט. משמעות הדבר, שלא ניתן לשמור למשל ערך משתנה מספרי.
- שליפה ושמירה קשה יחסית מבחינה תכנותית.

אם כן, מהו התחליף לשמירת מידע באתר, גם בין גלישה לגלישה?

תקן HTML5 מאפשר שיטות חדשות לאחסון נתונים בצד הלקוח, שתיים מתוכן אמורות להחליף את העוגיות.

השיטות הן: localStorage ו- sessionStorage.

שיטה א - localStorage.

localStorage היא דרך לאחסן נתונים בפורמט של value<=key (מפתח וערך).

הנתונים נשמרים **גם לאחר סגירת הדפדפן**, אלא שהשליפה והשמירה קלים יותר מהשימוש בעוגיות, וגודל האחסון הוא עד **5mb**.

תחביר:

שמירת מידע	localStorage.setItem('KEY', 'VALUE');
אחזור מידע	var value = localStorage.getItem('KEY');
מחיקת פריט	localStorage.clear('KEY');

תחביר נוסף:

```
localStorage['KEY', 'VALUE'];
```
שמירת מידע

```
var value = localStorage['KEY'];
```
אחזור מידע

דוגמא:

```
localStorge.setItem("moneGolshim", counter);
```

בדיקה שהפריט אכן קיים

```
if (localStorage.getItem("moneGolshim") != null)
    var golshim = localStorage.getItem("moneGolshim");
```

או

```
localStorge["moneGolshim"] = counter;
```

בדיקה שהפריט אכן קיים

```
if (localStorage["moneGolshim"])
     var golshim = localStorage["moneGolshim"];
```

שיטה ב – sessionStorage

דומה כמעט לחלוטין ל-localStorage, אך המידע נשמר רק **עד סגירת הדפד**פן. בגלישה הבאה הנתונים לא ישמרו. שיטה זו מיועדת להעברת מידע בתוך הדפים באתר, ולא לשמירת נתונים לגלישה הבאה.

דוגמא:

```
sessionStorage.setItem('KEY', 'VALUE');
```
שמירת מידע

```
var value = sessionStorage.getItem('KEY');
```
אחזור מידע

```
sessionStorage.clear('KEY');
```
מחיקת פריט

תחביר נוסף:

```
sessionStorage ['KEY', 'VALUE'];
```
שמירת מידע

```
var value = sessionStorage ['KEY'];
```
אחזור מידע

דוגמא:

```
sessionStorage.setItem("myColor", clr);
```

בדיקה שהפריט אכן קיים

```
if (sessionStorage.getItem("myColor"))
    var selectedClr = sessionStorage.getItem("myColor");
```

הבדל בין sessionStorage ל localStorat:

ההבדל ביניהן הוא הבדל קטן אך מהותי – sessionStorage מתקיימת לכל אורך חיי חלון הדפדפן. ע**ם סגירת הדפדפן, המידע שנשמר ב-Session אינו קיים יותר.** בעוד המידע שנמצא ב-localSession **נשמר** גם לאחר סגירת החלון.

דוגמא להצגת מספר הפעמים שמשתמש ביקר בדף מסוים:

```
if (localStorage.clickcount)
{
localStorage.clickcount=Number(localStorage.clickcount)+1;
}
else
{
localStorage.clickcount=1;
}
document.getElementById("result").innerHTML="You have clicked the button " +
localStorage.clickcount + " time(s).";
```

תכונות:

שם התכונה	תאור
Length	מספר הפריטים השמורים.

פונקציות:

שם הפונקציה	פרמטרים	תאור
Clear	ללא	מחיקת כל הפריטים השמורים
remainingSpace	ללא	מחזירה את גודל המקום הפנוי (בבתים).
Key	מספר סידורי של פריט באוסף	החזרת שם פריט באוסף של פריט לפי מיקום.
getItem	שם של פריט באוסף	החזרת הערך של פריט באוסף לפי שמו.
setItem	שמירת פריט באוסף	מקבל שם וערך ושומר את הפריט.

דוגמא:

מעבר על כל הפריטים באוסף, שליפת שם פריט לפי מיקומו הסידורי, ושליפת הנתונים של הפריט לפי שמו.

```
for (var i = 0; i <localStorage.length; i++) {

var key =localStorage.key(i);

var data =localStorage.getItem(key);
```

}

אירועים:

שם האירוע	תאור
Onstorage	כשתוכן פריט משתנה
Onstoragecommit	כאשר ה localStorage או ה sessionStorage סיים לשמור את עצמו בדיסק (קבצי xml).

סיכום:

שמירת בתקן html5 מקילה מאוד על הכתיבה, מאפשרת שמירת נתונים מורכבים יותר

טבלת השוואה בין עוגיות לאחסון בתקן html5:

	DOM Storage	Cookie
גודל מקסימלי	10MB	4KB (IE8 - 10KB)
מוכר בשרת	לא מוכר	נשלחים עם כל בקשה
הרשאות	לפי דומיין בלבד	לפי דומיין ותיקייה
גישה	בעזרת אובייקטים ומתודות	חיפוש על מחרוזות
אירועים	בזמן שינוי והוספה ובזמן סיום הכתיבה	אין
מחיקה	ידנית או בזמן סגירת הדפדפן	לפי זמן
סוג מידע	מחרוזות, מספרים וערכים בוליאניים	מחרוזות

© א. קצנלנבוגן

תרגיל מס' 5

הוסיפי לאתר שיצרת בתרגיל 1 דף "הגדרות אישיות", הוסיפי קישור לדף זה.

בדף "הגדרות אישיות" הגולש יכול לבחור אפשרויות עיצוב שונות.

כל אפשרות עיצוב ניתן לבחור עבור בחירה זמנית ובחירה ברירת מחדל. עבור כל סוג שמירה יש להגדיר את כל אפשרויות העיצוב.

אפשרויות העיצוב:

- צבע רקע האתר (פקד input type=color)
- צבע הגופן של האותיות (כנ"ל)
- סוג גופן (פקד select עם אפשרויות בחירה של גופנים)
- גודל גופן (כנ"ל עם אפשרויות גודל גופן)

בלחיצה על "אישור" יש לשמור את כל ההגדרות ע"פ הבחירה המתאימה:

- הגדרות זמניות – שמרי ב sessionStorage
- הגדרות ברירת מחדל – שמרי ב localStorage

בכל דף, הוסיפי קוד js לשימוש בהגדרות האישיות.

עבור כל אפשרות עיצוב יש לבדוק אם היא קיימת ב sessionStorage ולהשתמש בה. אם אינה קיימת, יש לשלוף אותה מתוך ה localStorage.

דוגמא לבדיקת צבע רקע:

```
if (sessionStorage.getItem("backColor")!=null)

  document.body.style.backgroundColor = sessionStorage.getItem("backColor");

else

      document.body.style.backgroundColor = localStorage.getItem("backColor");
```

כדי להציב את הערכים הנבחרים, יש לגשת לתכונה המתאימה ב style. דוגמא:

```
Document.body.style.color = sessionStorage.getItem("fontColor");
```

ערכי עיצוב:

צבע רקע – style.backgroundColor

צבע גופן – style.color

סוג גופן – style.fontFamily

גודל גופן – style.fontSize

תרגיל אתגר

צרי את המשחק "בול פגיעה"

הוראות:

1. **דף #1 – בחירת צבעים**

.1.1 על הדף יוצגו 4 ריבועים של צבעים. מתחת יהיו 4 מלבנים אליהם ניתן יהיה לגרור את הצבעים.

בלחיצה על "אישור", התוכנית שומרת את הצבעים הנבחרים במערך, והמשתמש מועבר לדף #2.

יש לשמור את מערך הצבעים ב- sessionStorage.

לצורך הדוגמא, נקרא למערך זה בשם arrSelect (מערך בחירה).

.1.2 רשות: בעת גרירת צבע, יוצב במלבן העתק הצבע (צבע הרקע יוגדר כמו הצבע הנגרר), כדי שהצבע המקורי ישאר, וניתן יהיה לבחור כמה פעמים צבע אחד.

2. **דף #2 - משחק**

.2.1 על הדף יוצגו 4 ריבועים של צבעים. בשורה מתחת יוצבו 4 מלבנים אליהם ניתן יהיה לגרור את הצבעים.

המשתמש מנסה לנחש את הצבעים והסדר הנכון, ע"י גרירת הצבעים למלבנים.

באירוע drop – סיום הגרירה, יש לשמור את הצבע הנגרר במערך.

לצורך הדוגמא, נקרא למערך זה בשם arrGuess (מערך ניחוש).

.2.2 בלחיצה על "המשך", התוכנית בודקת את נכונות הצבעים:

יש לחלץ מה- sessionStore את המערך arrSelect, ולהשוות אותו למערך arrGuess.

התוכנית מדפיסה על המסך סדרה של בולים ופגיעות בהתאם.

הדפסת בולים ופגיעות: ציור על קנבס של סדרת מלבנים צבועים. שחור לבול, אפור לפגיעה.

יש לשמור את מספר הניחושים ב localStorage .

.2.3 בעת ניצחון, יש להשמיע צליל נצחון.

2.4. **רשות:** בדף הכניסה למשחק תהיה הזדהות עם שם. בסיום המשחק ישמר מס' הניחושים למשתמש זה. בכניסה הבאה למשחק של המשתמש, יוצגו מס' הניחושים של המשחק האחרון.

דוגמא:

דף 1:

דף 2:

Notification

נספח לתרגיל אתגר

הוסיפי את שורות הקוד הבאות למשחק בול פגיעה.

```
function RequestPermission(callback) {
    window.webkitNotifications.requestPermission(callback);
  }

  function showNotification() {
    if (window.webkitNotifications.checkPermission() > 0) {
      RequestPermission(showNotification);
    }
    else {
      window.webkitNotifications.createNotification("כתבי כאן נתיב של תמונה", " כתבי כאן
כותרת", "כתבי כאן את מספר הניחושים השמורים ב localStorge").show();
    }
  }
```

א. זמני את הפונקציה showNotification בסיום המשחק.

ב. הסבירי מהו notification?

© א. קצנלנבוגן