

לולאות

- א. למדנו על לולאה מסוג While.
- ב. מטרת הלולאה היא לבצע מספר פעולות ברצף.
- ג. הלולאה מקבלת תנאי שרק אם הוא מתקיים הלולאה רצה.

```
While( תנאי )
{
    הקוד שירוצ כל התנאי מתקיים
}
```

- ד. הלולאה חוזרת על עצמה כל עוד התנאי נכון.
- ה. בכדי למנוע לולאה אין סופית (כלומר, לולאה שרצה בלי הפסקה) שעלולה לתקוע את הדפדפן, אנו נדאג לשים תנאי עצירה תקין.
- ו. בכדי להגדיר תנאי עצירה, אנו נגדיר משתנה מחוץ ללולאה ונקדם אותו בכל איטרציה פעם אחת.

```
var x = 0; // נגדיר משתנה ונאפס אותו

While(x < 50) // הלולאה תתקיים כל עוד המשתנה יהיה קטן מ-50
{
    console.log(x); // בכל איטרציה נכתוב את הערך של המשתנה לקונסול
    x++; // בכל איטרציה נקדם את המשתנה ב-1
}
```

- ז. לולאה מאוד יעילה בטיפול במערכים.
- ח. נגדיר מערך לדוגמא:

```
var colors = ["red", "green", "blue"];
```

- ט. ניצור לולאה שבכל איטרציה תרשום בקונסול את הערך של האיבר באינדקס הנוכחי.

```
var i = 0;

While(i < colors.length) // הלולאה תרוץ כל עוד המשתנה יהיה קטן מאורך המערך
{
    console.log(colors[i]);
    i++;
}
```

- י. ניתן לעצור לולאה ע"י שימוש ב-break; או לעצור את האיטרציה הנוכחית ע"י שימוש ב-continue;.

לולאה מסוג for:

- א. בהמשך ללולאה while שלמדנו בשיעור הקודם, למדנו על לולאה מסוג for.
- ב. כפי שלמדנו, רוב הלולאות מורכבות מ:
- איפוס המונה (משתנה).
 - תנאי עצירה.
 - קידום המשתנה (מונה).
- ג. בלולאה מסוג for כל הפרמטרים האלו נכנסים לסוגריים של הלולאה.

```
for(קידום המונה; תנאי עצירה; אתחול המונה)
{
    הקוד שירוצ כל התנאי מתקיים
}
```

- ד. נכתוב לולאה שרצה 10 פעמים, ומוסיפה את המספרים למערך.

```
var numbers = [];

for(var i = 0; i < 10; i++)
{
    numbers.push(i);
}
```