# **Class Notes**

**String Interpolation**

מחרוזת מיוחדת הבנויה מ-Back Ticks (``) המאפשרת לבצע שני דברים:

1. לשבץ ערכים בתוך המחרוזת.
2. לרדת שורה באמצע המחרוזת בקוד.

דוגמה לשיבוץ מידע:

var name = “Moishe”, age = 20;

var message = `The name is: ${name}, the age is: ${age}`;

דוגמה לירידת שורה באמצע המחרוזת:

var longText = `This is a very

long text … `

**for-of**

לולאה המאפשרת לרוץ על מערכים:

for(var item of myArray) {

}

בתוך הלולאה המשתנה item הינו הפריט עצמו ולא האינדקס של הפריט.

לכן באיטרציה הראשונה, item יהיה הפריט הראשון, באיטרציה השנייה, item יהיה הפריט השני וכן הלאה.

**אופרטורי קיצור**

אלו אופרטורים המקצרים פעולה מתמטית / מחרוזתית.

|  |  |  |
| --- | --- | --- |
| אופרטור | שימוש | משמעות |
| += | a += b | a = a + b |
| -= | a -= b | a = a - b |
| \*= | a \*= b | a = a \* b |
| /= | a /= b | a = a / b |
| %= | a %= b | a = a % b |

קיימת עוד קבוצה של אופרטורי קיצור אחרים:

|  |  |  |
| --- | --- | --- |
| אופרטור | שימוש | משמעות |
| ++ עבור קידום מאוחר | a++ | a = a + 1 |
| ++ עבור קידום מוקדם | ++a | a = a + 1 |
| ++ עבור קידום מאוחר | b = a++ | b = a  a = a + 1 |
| ++ עבור קידום מוקדם | b = ++a | a = a + 1  b = a |
| -- עבור חיסור מאוחר | a-- | a = a - 1 |
| -- עבור חיסור מוקדם | --a | a = a - 1 |
| -- עבור חיסור מאוחר | b = a-- | b = a  a = a - 1 |
| -- עבור חיסור מוקדם | b = --a | a = a - 1  b = a |

מטריצות (מטריצה = Matrix)

מטריצה הינה רצף דו-מימדי של מספרים (נקרא מערך דו-מימדי)

(מערך הינו רצף חד-מימדי של מספרים).

דוגמה למטריצה בגודל 3 על 4 (3 שורות ו-4 עמודות פר שורה, נקראת N x M):

40 80 12 35

89 23 17 43

83 24 55 11

דוגמה למטריצה בעלת גדלים שונים של שורות:

12 23 34 45 56 57

32 44 55 67

23 34

56 5 234 45 5 6 67 5 45 34

בג'אווה סקריפט מטריצה היא מערך של מערכים.

מערך יכול להכיל בכל תא שלו מספר או מחרוזת או אובייקט או מערך וכו'.

אז אם מערך מכיל בכל תא שלו מערך אחר – קיבלנו מטריצה.

**Types**

Type זה סוג המשתנה.

בג'אווה סקריפט יש מספר סוגים של משתנים:

var a = 100; // type = number

var b = “Hello”; // type = string

var c = true; // type = Boolean

var d = { name: “Mitsi”, age: 4 }; // type = object

var e = document.write; // type = function

var f; // type = undefined

var g = Symbol(“Moishe”); type = symbol

**Truthy:**

כל ערך שמשפט תנאי מחשיב כ-true.

כך ערך שאינו Falsy יהיה Truthy.

**Falsy:**

כל ערך שמשפט תנאי מחשיב כ-false.

יש רק שישה ערכים ששווים ל-false. כל היתר יהיו true.

הערכים הם:

0, null, undefined, "", NaN, false

**Abstract Comparison**

השוואה ע"י ==

אי שוויון ע"י !=

השווה כזו לא משווה את ה-Types אלא רק את ה-Values ומכילה אי דיוקים בתוכה. לפעמים תחזיר true כאשר ערכים אינם זהים. לדוגמה השוואה כזו תחזיר true עבור “” == 0

**Strict Comparison**

השוואה ע"י ===

אי שוויון ע"י !==

השוואה כזו משווה ראשית את ה-Types ורק אם שווים תבצע השוואה של ה-Values.

השוואה כזו לא תכיל אי דיוקים. לדוגמה “” === 0 יחזיר false

אנו תמיד נשתמש ב- === או ב- ==!