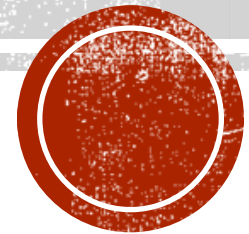


# PROGRAMMING

## The basics – part 2



# DOCUMENT OBJECT

- אובייקט שהדפדפן מספק לנו
- מאפשר גישה לדף **html** על ידי אובייקט זה
- כל שינוי באובייקט יגרור שינוי מיידי בדף

```
document.querySelector('.name-of-class')
```

# CONSOLE OBJECT

- מסופק לנו על ידי הדפדפן
- מאפשר כתיבת הודעות לתוך console במצב של developer בדפדפן

```
console.log('developer message')
```

# COMMENTS

`// One line comment`      ■ הערה של שורה אחת:

■ הערה של מספר שורות:

```
/*  
    multiline comment  
    cant have a lot of line  
*/
```

# JAVASCRIPT NAMING RULES

- שם משתנה יכול להכיל מספרים, אותיות, קו תחתי ואת סימן הדולר.
- שם משתנה לא יכול להתחיל במספר.

# OBJECT

- אובייקט הוא טיפוס מורכב.
- יכול להכיל בתוכו את כל סוגי הערכים הפרימיטיבים וגם אובייקטים אחרים
- מוגדר על ידי סוגריים מסולסלות

```
var person = {  
  age: 21,  
  name: 'student'  
}
```

# OBJECT — OTHER WAYS

■ יש כמה שיטות להגדיר מאפיינים של אובייקט

```
var person = {  
  age: 21,  
  name: 'student'  
}
```

```
var person = {};  
person.age = 21;  
person.name = 'student';
```

```
var person = {};  
person['age'] = 21;  
person['name'] = 'student';
```

# שמות משתנים

- מוסכמות לגבי שמות משתנים

- משתנה קבוע יהיה אותיות גדולות בלבד, אם יש כמה מילים קו תחתי יפריד ביניהם

```
var MAX_STUDENTS_PER_ROOM = 20;
```

- משתנה רגיל יהיה באותיות קטנות, במקרה של ריבוי מילים המילה השנייה ואילך יתחילו באות גדולה

```
var currentStudentsPerRoom = 15;
```



# LOOPS - WHILE

- במידה ויש לנו פעולה שאנחנו רוצים לבצע מספר רב של פעמים
- הלולאה תתבצע כל זמן שהתנאי הוא `true`
- כל זמן שהתנאי מתקיימת תתבצע כניסה ללולאה – איטרציה
- יש להיזהר מלולאות אינסופיות

```
while (condition) {  
    //some code  
}
```

# LOOPS — WHILE EXAMPLE

```
var NUM = 10;  
var currentNum = 1;  
var result = 0;  
  
// Calculate the sum of 1 to 10  
while (currentNum < NUM) {  
    result = result + currentNum;  
    currentNum++;  
}
```

# LOOPS - FOR

- לולאת **for** מורכבת מ-3 חלקים.
- חלק ראשון מגדיר אתחול שמתבצע לפני הכניסה ללולאה
- חלק שני מגדיר את התנאי לביצוע איטרציה בלולאה
- חלק שלישי מגדיר מה יתבצע בכל פעם שהלולאה מסיימת איטרציה
- בין החלקים יפריד נקודה-פסיק

```
for (initialize; condition; afterIteration) {  
    //some code  
}
```

# LOOPS — FOR EXAMPLE

```
var NUM = 10;  
var result = 0;  
for (var currentNum = 1; currentNum < NUM; currentNum++) {  
    result += currentNum;  
}
```

# FIZZBUZZ VOLUME 2

Remember *FizzBuzz* from Volume 1? Let's give a more complex version a try. ■

Print all numbers from 1 to 100 with three exceptions: ■

If the number is divisible by 3, print fizz ■

If the number is divisible by 5, print buzz ■

If the number is divisible by 3 AND 5, print fizzbuzz ■

```
var NUM = 10;
var result = 0;
for (var currentNum = 1; currentNum < NUM; currentNum++) {
    result += currentNum;
}
```

# FUNCTION

- נפגוש בכמה שמות: שגרה (באנגלית: Subroutine), פרוצדורה (באנגלית: Procedure) או שיטה (באנגלית: Method)
- נשתמש כאשר יש קטע קוד שאנחנו צריכים לבצע מספר פעמים בתכנית
- פונקציה תבצע לוגיקה אחת בלבד, לדוגמה, אם יש לנו מחשבון לא נכתוב פונקציה שגם מחשבת כפל וגם חילוק, נבנה 2 פונקציות אחת לכפל ואחת לחילוק.
- פונקציה יכולה לקבל ערכים לתוך משתנים על ידי שימוש בסוגריים מעוגלות
- פונקציה יכולה להחזיר תשובה על ידי `return`

# FUNCTION - EXAMPLE

```
function multiplication(x, y) {  
    return x * y;  
}
```

```
function division(x, y) {  
    return x / y;  
}
```

```
function division(x, y) {  
    if (y == 0) {  
        console.log('y is 0');  
    }  
    return x / y;  
}
```

# ARRAY

■ אוסף של איברים מכל סוג



# ARRAY WITH LOOP