

JS3

Document Object Model

DOM

- עברו כל דף HTML שהדף טוען הוא יוצר לפני אובייקט מודל הנקרא DOM -
Document Object Model
- דף HTML מתייחס לדף HTML נוסף סדור של – תגיות, שדות וטקסטים
בDOM כל אובייקט צזה נקרא `node`.
האלמנטים מוחזקים בעץ של `nodes`.
- Javascript יכולה לשנות את תוכן הדף על ידי גישה לאובייקטים שלו.
הDOM מאפשר גישה לאובייקטים ובכך מאפשר לחפש, לשנות ולהסר אלמנטים.

fonkziot DOM

`setAttribute`

`createElement`

`appendChild`

`insertBefore`

`removeChild`

setAttribute

element.setAttribute(name, value);

```
document.querySelector("div").setAttribute("class", "democlass");
```

```
let x = document.querySelector("div").getAttribute("class");
```

```
document.querySelector("div").removeAttribute("class");
```

setAttribute

```
myAnchor.setAttribute("href", "https://www.mysite.com");
```

```
document.getElementById("game-container").setAttribute("class", "game-con");
```

```
gameHeader.setAttribute("id", "game-header");
```

```
checkbox.setAttribute('checked', 'checked');
```

```
myImg.setAttribute('src', 'images/dog.jpg');
```

setAttribute

```
function change() {  
    document.querySelector("input").setAttribute("type", "button");  
}  
  
<body>  
    <div id="div1">This is static div.</div>  
    <input onclick="change()" value="Click me" />  
</body>
```

createElement method

```
let element = document.createElement(-----insert tag name---);  
Tag name - div, span, p, h1.....
```

```
const newDiv = document.createElement("div");  
newDiv.innerHTML = " Good Morning !";  
document.body.appendChild(newDiv);
```

תרגיל חימום

עם טיענת הדף צרו 3 DIVS, כאשר בכל DIV תופיע כותרת לפי התמונה מטה וצבע ה-DIV יהיה לפי התמונה:



*בתוך ה-body לא יהיה אף אלמנט בצורה סטטית,
אפשר לשים script ב-body

פתרונות

```
<script>
    const colors = ["green", "orange", "red"];

    for (let i = 0; i < 3; i++) {
        let div = document.createElement("div");
        let h = document.createElement("h1");
        div.append(h);
        h.innerHTML = "header " + (i + 1);
        div.style.backgroundColor = colors[i];

        document.body.append(div);
    }
</script>
```

הוספת אירוח בצורה דינמית

onclick .1

addEventListener .2

onclick

```
let el = document.getElementById("element");
```

```
el.onclick = function x() { console.log('Hi'); }
```

onclick

```
let el = document.getElementById("element");
el.onclick = function() { console.log('Hi'); }
```

IX

```
el.onclick = x;
function x() { console.log('Hi'); }
```

onclick

```
let el = document.getElementById("element");
```

מקבל תמיד את ה event

```
el.onclick = x;
```

```
function x() { console.log( event.currentTarget.innerText); }
```

מכיר את this

```
el.onclick = x;
```

```
function x() { console.log( this.innerText); }
```

onclick

```
document.getElementById("para").onclick = fun;  
  
function fun() {  
    document.getElementById("para").innerHTML = "Welcome to Client side class";  
    document.getElementById("para").style.color = "blue";  
    document.getElementById("para").style.backgroundColor = "yellow";  
    document.getElementById("para").style.fontSize = "25px";  
    document.getElementById("para").style.border = "4px solid red";  
}
```

onclick

```
<body>
<button id="myBtn">Try it</button>
<p id="para"></p>
<script>
document.getElementById("myBtn").onclick = fun;

function fun() {
    document.getElementById("para").innerHTML = "Welcome to Client side class";
    document.getElementById("para").style.color = "blue";
    document.getElementById("para").style.backgroundColor = "yellow";
    document.getElementById("para").style.fontSize = "25px";
    document.getElementById("para").style.border = "4px solid red";
}
</script>
</body>
```

onclick

In previous example - better use **this**

```
<body>
<button id="myBtn">Try it</button>
<p id="para"></p>
<script>
    document.getElementById("myBtn").onclick = function () { fun(this) };

    function fun(elem) {
        elem.innerHTML = "Welcome to Client side class";
        elem.style.color = "blue";
        elem.style.backgroundColor = "yellow";
        elem.style.fontSize = "25px";
        elem.style.border = "4px solid red";
    }
</script>
</body>
```

שימוש ב `addEventListener`

```
let el = document.getElementById("element");

el.addEventListener("click", function () { console.log("Hi");} );
```

addEventListener

```
element.addEventListener("click", myFunction);
```

```
function myFunction() {  
    document.getElementById("demo").innerHTML = "Hello World";  
}
```

שימוש קצר יותר:

```
element.addEventListener("click", function() {  
    document.getElementById("demo").innerHTML = "Hello World";  
});
```

addEventListener

ניתן להוסיף כמה אירועים לאותו אלמנט

```
<body>
  <button id="myBtn">Try it</button>
  <p id="demo"></p>
  <script>
    const element = document.getElementById("myBtn");
    element.addEventListener("click", myFunction1);
    element.addEventListener("click", myFunction2);
    function myFunction1() {
      document.getElementById("demo").innerHTML += "First function was executed! "
    }
    function myFunction2() {
      document.getElementById("demo").innerHTML += "Second function was executed!"
    }
  </script>
</body>
```

```
<body>
  <button id="myBtn">Try it</button>
  <p id="demo"></p>
  <script>
    const element = document.getElementById("myBtn");
    element.addEventListener("mouseover", myFunction);
    element.addEventListener("click", mySecondFunction);
    element.addEventListener("mouseout", myThirdFunction);
    function myFunction() {
      document.getElementById("demo").innerHTML += "Moused over!<br>"
    }
    function mySecondFunction() {
      document.getElementById("demo").innerHTML += "Clicked!<br>"
    }
    function myThirdFunction() {
      document.getElementById("demo").innerHTML += "Moused out!<br>"
    }
  </script>
</body>
```

addEventListener כיתן להוסף אירועים שונים

```
<body>
```

```
  <h1>The Element Object</h1>
```

```
  <h2>The addEventListener() Method</h2>
```

```
  <p>How to pass parameter values with addEventListener().</p>
```

```
  <p>Click the button to perform a calculation.</p>
```

```
  <button id="myBtn">Try it</button>
```

```
  <p id="demo"></p>
```

```
  <script>
```

```
    let p1 = 5;
```

```
    let p2 = 7;
```

```
    document.getElementById("myBtn").addEventListener("click", function () {
      myFunction(p1, p2);
    });
  
```

```
  function myFunction(a, b) {
```

```
    document.getElementById("demo").innerHTML = a * b;
```

```
  }
```

```
</script>
```

```
</body>
```

העברת פרמטרים יכולת לבצע רק דרך

פונקציה פנימית שמקבלת פרמטרים

הfonקצייה מבירה את this
addEventListener

```
<body>
  <button id="myBtn">Try it</button>
  <script>
    document.getElementById("myBtn").addEventListener("click", function () {
      this.style.backgroundColor = "red";
    });
  </script>
</body>
```

תרגיל ביתה חלק ב

הוסף אירוע קליק לכל div אשר לחיצה תשנה את צבעו לצהוב

הוסף אירוע דאבל קליק לכל div אשר לחיצה עליו תמחק אותו השתמש
בפונקציה (`remove()`) על האלמנט עצמו להסרתו



פתרונות

```
<body>
```

```
<script>
    const colors = ["green", "orange", "red"];

    for (let i = 0; i < 3; i++) {
        let div = document.createElement("div");
        div.onclick = function () { this.style.backgroundColor = 'yellow' };
        //div.addEventListener("click", function () { this.style.backgroundColor = 'yellow' });
        //div.setAttribute("onclick", "this.style.backgroundColor = 'yellow'");
        div.ondblclick = function () { this.remove() };
        let h = document.createElement("h1");
        div.append(h);
        h.innerHTML = "header " + (i + 1);
        div.style.backgroundColor = colors[i];

        document.body.append(div);
    }
</script>
</body>
```

הציגה בכיתה

כתב את כל הֆונקציות שנמצאות עבורי או ירעד click

```
<!DOCTYPE html>
<html>
<head>
    <style type="text/css">
        #specialDiv {
            color: red;
        }

        .colorBlue {
            color: blue;
        }

        .buttons {
            padding: 10px;
            border: 1px solid grey;
            border-radius: 4px;
        }
    </style>
<script>
    let counter = 0;
    function getCounter() {
        counter += 1;
        updateCounter();
        return " " + counter;
    }
    function updateCounter() {
        let counterElement = document.getElementById("counter");
        counterElement.innerText = " " + counter;
    }
</script>
</head>
<body>
    <div class="buttons">
        <input type="button" onclick="addFirstElement();" value="Add element" />
        <input type="button" onclick="addElementToSpecialDiv();" value="Add element within div" />
        <input type="button" onclick="addLink();" value="Add a link" />
        <input type="button" onclick="innerTextVsinnerHTML();" value="innerText VS innerHTML" />
        <input type="button" onclick="createElementWithCSSClass();" value="Create with CSS class" />
        <input type="button" onclick="createElementWithStyle();" value="Create with style" />
        <input type="button" onclick="insertElement();" value="Insert (before) element" />
        <input type="button" onclick="createBullets();" value="Create bullets" />
        <input type="button" onclick="addButtons();" value="Add a lot of buttons" />
        <br><br>
        <div>
            Elements added: <span id="counter">0</span>
        </div>
        <br>
    </div>
    <div id="specialDiv"></div>
</body>
</html>
```

- 1) [addFirstElement](#) - add <p> to body
- 2) [addElementToSpecialDiv](#) - insert <p> to <div id="specialDiv"></div>
- 3) [addLink](#) - add href
- 4) [innerTextVsInnerHTML](#) - add text "This text is with a bolded part." to div
innerHTML / innerText
- 5) [createElementWithCSSClass](#) - add Css class to <p>
- 6) [createElementWithStyle](#) - add style inline to <p>
- 7) [insertElement](#) - insert <p> to <div id="specialDiv"></div> to show first
- 8) [createBullets](#) - create ul..li
- 9) [addButtons](#) - add 100 buttons, click will remove it - **בסוף הרצאה נסיף**

תרגיל כתה

צור שני תא טקסט:

1. לחיצה על התא הראשון הוא יפותר לכפתור

2. לחיצה על התא השני תחזיר אותו להיות טקסט.

על ה `body` להיות ריק – יש ליצור באופן דינמי את האלמנטים באמצעות פונקציות של ה DOM

[change to button](#)

[change to text](#)

[change to button](#)

[change to text](#)

פתרונות DOM_Element.html

תרגיל כיתה מתקדם Ex Fruit

Store

Click to Add



Basket

Double click to remove

1. ב click על פרי מהחנות הוסף אותו לסל
2. ב dbclick על פרי מהסל הסר אותו
3. Skelton נמצא בmodel moodle

Objects in JS

Objects

- אובייקט מיוצג על ידי סגירים מסוללים ובתוכו properties
- key : value של Property
- string הוא Key
- Value יכול להיות כל ערך

```
let user = new Object(); // "object constructor" syntax  
let user = {}; // "object literal" syntax
```

```
let user = {  
    name: "John",  
    age: 30  
};
```

```
console.log(user.name);  
console.log(user.age);
```

```
user.isAdmin = true;
```

```
delete user.age;
```

```
//We can also use multiword property names, but then they must be quoted:  
let user = {  
    name: "John",  
    age: 30,  
    "likes birds": true // multiword property name must be quoted  
};
```

```
// this would give a syntax error  
console.log(user.likes birds); // true
```

```
console.log(user["likes birds"]); // true
```

```
let user = {};  
  
// set  
user["likes birds"] = true;  
  
// get  
console.log(user["likes birds"]); // true  
  
// delete  
delete user["likes birds"];  
  
let key = "likes birds";  
  
// same as user["likes birds"] = true;  
user[key] = true; //OR user.key = true;
```

for...in Loop

עובר בלולאה על המפתחות של האובייקט

```
for (let key in object) {  
    // executes the body for each key among object properties  
    // key, object[key]  
}
```

for...in Loop

```
const student = {  
    name: "Dana",  
    age: 21,  
    grade: "A",  
    isActive: true  
};  
  
for (let key in student) {  
    console.log(key + ": " + student[key]);  
}
```

עובר בלולאה על המפתחות של האובייקט

תרגיל ביתה

כתב פונקציה המקבלת אובייקט שערכי המפתחות שלו הם מספרים ומחזירה את סכום
המספרים

```
let grades = {  
    John: 100,  
    Ann: 60,  
    Pete: 85  
};
```

למשל עבור האובייקט:

יוחזר הסכום 245

פתרונות

```
let grades = {
    John: 100,
    Ann: 60,
    Pete: 85
};

sumProp(grades);

function sumProp(obj) {
    let sum = 0;
    for (let key in obj) {
        sum += obj[key];
    }
}

console.log(sum); // 245
}
```

יצירת אובייקט

3 דרכים ליצור אובייקט

1. **שירות** - על ידי הצהרה ואז השמה של ה properties כפי שראינו
2. **динמית** – על ידי שימוש ב `(new Object)`
3. **בנאי** - "**קונסטרוקטור**" וממנו יצירת מופעים

1. הצחה והשמה

```
let personObj = {
    firstname: "John",
    lastname: "Doe",
    age: 50,
    tellYourage: function () {
        console.log("The age is " + this.age);
    },
    tellSomething: function (something) {
        console.log(something);
    }
}
personObj.tellYourage();
personObj.tellSomething("Life is good!")
```

```
personObj.newField = "some data";
personObj.fullName = personObj.firstname + " " +
personObj.lastname;
personObj.myfunction = function () {
    console.log(this.fullName + " is " + this.age);
}
personObj.myfunction();
```

2. דינמית - יצרת מופע שימוש ב new Object

```
personObj= new Object();
personObj.firstname = "John";
personObj.age = 50;
personObj.tellYourage = function () {
    console.log("This age is " + this.age);
}
// You can call then tellYourage function as following
personObj.tellYourage();
```

פונקציה מוגדרת מראש

```
function tellYourage() {  
    console.log("The age is" + this.age);  
}  
  
personObj.tellYourage = tellYourage;
```

שים לב להבדל:

מציב בשדה פונקציה //

```
personObj.tellYourage = tellYourage;
```

מציב בשדה ערך מוחזר של פונקציה //

```
personObj.tellYourage = tellYourage();
```

3. "קונסטרקטור"

```
// תבנית ליצירת אובייקט מتبצע על ידי function ניתן להסתכל על זה לעל
function Person(firstname, lastname, age)
{
    this.firstname = firstname;
    this.lastname = lastname;
    this.age = age;
    this.tellYourage = function () {
        console.log("This age is " + this.age);
    }
}
```

מופעים של האובייקט//

```
myFather = new Person("John", "Doe", 50);  
myMother = new Person("Sally", "Rally", 48);
```

/ ניתן להוסיף תכונות ופונקציות//

```
myFather.newField = "some data";  
myFather.fullName = myFather.firstname + " " + myFather.lastname;  
myFather.myfunction = function () {  
    console.log(this.fullName + " is " + this.age);  
}
```

Prototype

```
function MyObject(name, size) {  
    this.name = name;  
    this.size = size;  
}  
  
// Add a function to the prototype  
MyObject.prototype.tellSize = function() {  
    return "size of " + this.name + " is " + this.size;  
}  
  
// ייצרת מופע לאובייקט וקריאה לפונקציה שלו  
let myObj = new MyObject("Sang", "30 inches");  
Console.log( myObj.tellSize() );
```

תרגיל ביתה

```
let nums = [1,2,3,4,5]
```

```
let colors = ["blue", "red", "green"]
```

1,blue
1,red
1,green
2,blue
2,red
2,green
3,blue
3,red
3,green
4,blue
4,red
4,green
5,blue
5,red
5,green
▶ Array(15)

- 1) צור מערך של אובייקטים המייצגים קלפים (השתמש ב `new object` או יצר קונסטרקטור `Card`) כל אובייקט מכיל – מספר וצבע `obj.color` `obj.type`
- 1) על המערך להכיל את כל הקלפים האפשריים, כלומר כל האפשרויות של הצירוף מספר-צבע.
- 1) הדפסו ל `log` את כל אברי המערך

המשר תרגיל ביתה

בנו פונקציה כללית שמקבלת מערך ועושה shuffle – חפשו ברשת פונקציה כזו הבינו אותה והעתיקו אליכם.

בצעו shuffle על המערך והדפינו ל `console.log`

```
let nums = [1,2,3,4,5]
let colors = ["blue", "red", "green"]
let arr = [];

for (let i in nums) {
  for (let j in colors) {
    let obj = new Object();
    obj.type = nums[i];
    obj.color = colors[j];
    arr.push(obj);
    console.log(obj.type + "," + obj.color);
  }
}

function shuffle(a) {
let j;
for (let i = a.length - 1; i > 0; i--) {
  j = Math.floor(Math.random() * i);
  x = a[i];
  a[i] = a[j];
  a[j] = x;
}
return a;
}
arr = shuffle(arr);
console.log(arr);
```

פתרון

תרגיל כיתה

צור אובייקט מסוג Deck

שדות האובייקט: מערך של קלפים cards

fonkציות האובייקט:

1. - יצרת חבילה קלפים מהמספרים 1,2,3,4,5 createDeck .1

ומהצבעים 'blue', 'red', 'green' (15 קלפים)

2. - מעربבת את החבילה (fonקציה קיימת בראש חפסו

אותה והעתיקו אליהם)

3. - מקבלת קלף ומוסיפה אותו לחבילה addCard .3

4. - מקבלת קלף ומסירה אותו מהחבילה removeCard .4

צור אובייקט מסוג Card

שדות האובייקט: num, color

fonkציות האובייקט:

1. isEqualColor - מקבלת צבע ומחזירה true או false אם הקלף באותו צבע שהתקבל

תרגיל ביתה

מהלך התוכנית:

- צור חבילה קלפים (15 קלפים)
- ערבל אותה על ידי shuffle
- הגרל קלף מהחבילה
- הוציא את כל הקלפים בפעם שהוגרל והעביר אותם לחבילה חדשה.
למשל : אם הוגרל יрок – כל הקלפים הירוקים יוצאו מחבילת הקלפים ויעברו לחבילה חדשה שתיצרו.
בסוף התהילה – בחבילה הישנה יהיו 10 קלפים בחבילה החדשה יהיו 5 קלפים באותו צבע שהוגרל.
- הדפינו ל `console.log` את חבילות הקלפים

```
<script>
function Card(num, color,) {
this.num = num;
this.color = color;
}
function Deck() {
this.cards = [];
}

Card.prototype.isEqualColor = function (i_color) {
if (this.color == i_color)
return true;
else
return false;
}

Deck.prototype.createDeck = function () {
var nums = ['1', '2', '3', '4', '5'];
var colors = ['blue', 'red', 'green'];
for (var i = 0; i < colors.length; i++) {
for (var j = 0; j < nums.length; j++) {
this.cards.push(new Card(nums[j], colors[i]));
//console.log(this.cards);
}
}
}
Deck.prototype.shuffle = function () {
var i = this.cards.length, j, temp;
while (--i > 0) {
j = Math.floor(Math.random() * (i + 1));
temp = this.cards[j];
this.cards[j] = this.cards[i];
this.cards[i] = temp;
}
}
Deck.prototype.addCard = function (card) {
this.cards.push(card);
}
Deck.prototype.removeCard = function (card) {
var index = this.cards.indexOf(card);
//console.log(index);
this.cards.splice(index, 1);
}

var deckMain = new Deck();
var deckNew = new Deck();

deckMain.createDeck();

var randIndex = Math.floor(Math.random() * deckMain.cards.length);
//console.log(randIndex);
var randColor = deckMain.cards[randIndex].color;

createNewDeckByRandColor(randColor);

function createNewDeckByRandColor(randColor) {
var deckTemp = new Deck();
for (var i = 0; i < deckMain.cards.length; i++) {
if (deckMain.cards[i].isEqualColor(randColor)) {
deckTemp.addCard(deckMain.cards[i]);
}
}

for (var i = 0; i < deckTemp.cards.length; i++) {
deckNew.addCard(deckTemp.cards[i]);
deckMain.removeCard(deckTemp.cards[i]);
}
}

console.log(deckMain.cards);
console.log(deckNew.cards);
</script>
```

By Val By Ref in javascript

Primitive & Non-Primitive Data Types

Value Types/ Primitive

Number
String
Boolean
Symbol
Undefined
Null

Reference Types/ Non- Primitive

Object
Array
Function

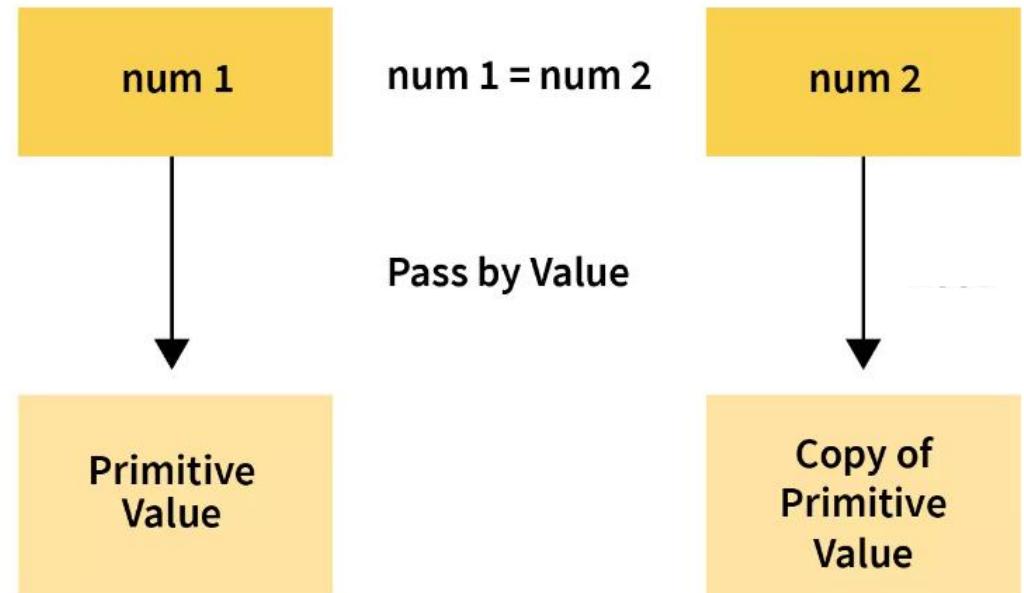
By Val

```
<script>
  let num1 = 70
  let num2 = num1

  console.log(num1) // 70
  console.log(num2) // 70

  num1 = 40

  console.log(num1) // 40
  console.log(num2) // 70
</script>
```



By Val

```
<script>
function multiplication(tmp) {
    tmp = tmp * 50;
    return tmp;
}
let num = 30;
let result = multiplication(num);
console.log(num); // 30
console.log(result); // 1500
</script>
```

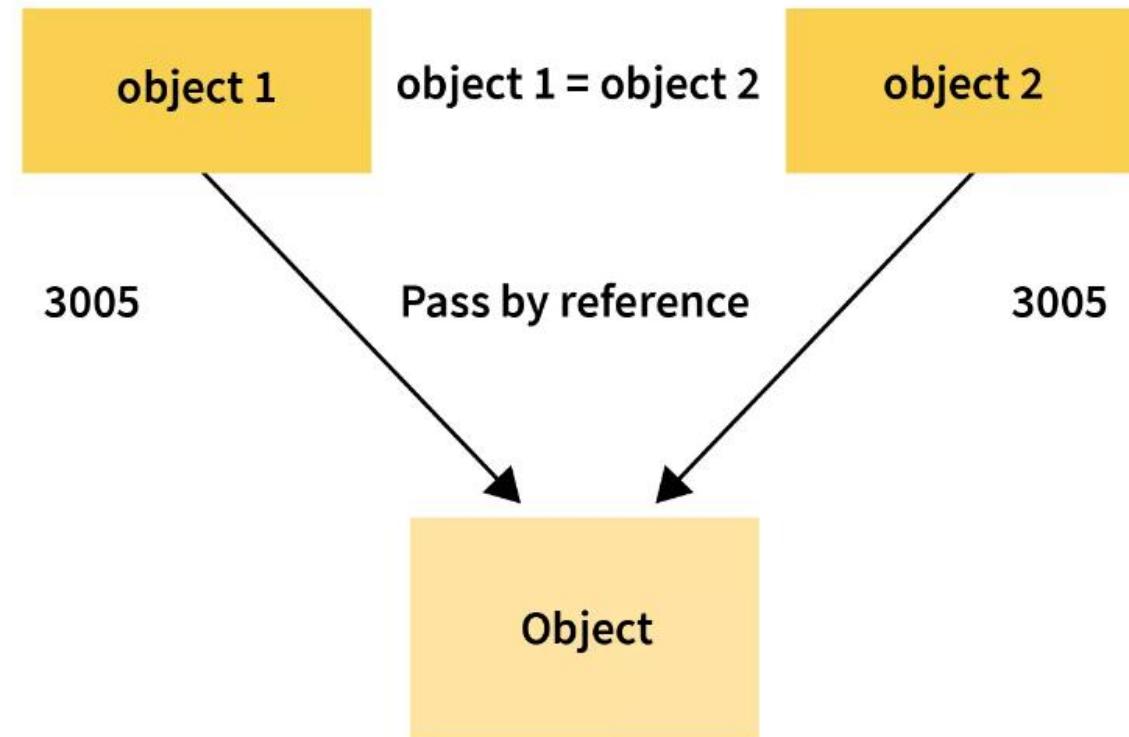
By Ref

```
<script>
  let obj1 = { website: "Yael" }
  let obj2 = obj1;

  console.log(obj1)      // {website: "Yael"}
  console.log(obj2)      // {website: "Yael"}

  obj1.website = "Uri"

  console.log(obj1)      // {website: "Uri"}
  console.log(obj2)      // {website: "Uri"}
</script>
```

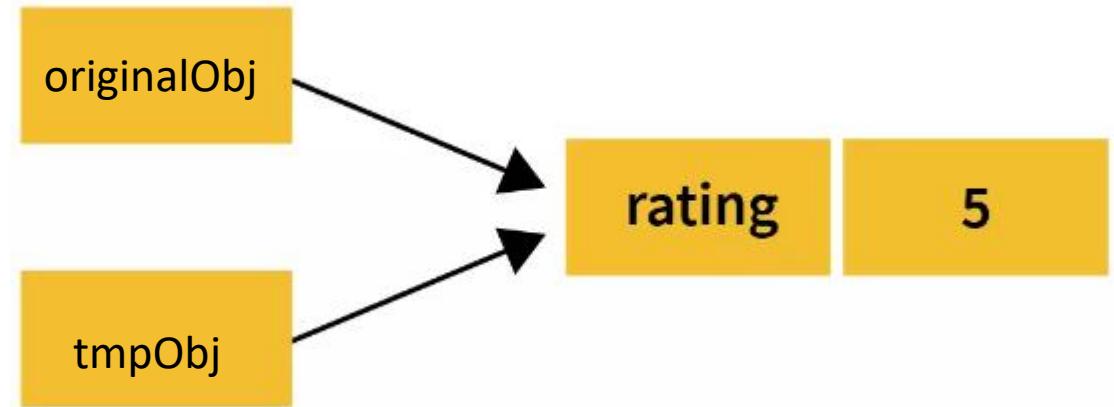


By Ref

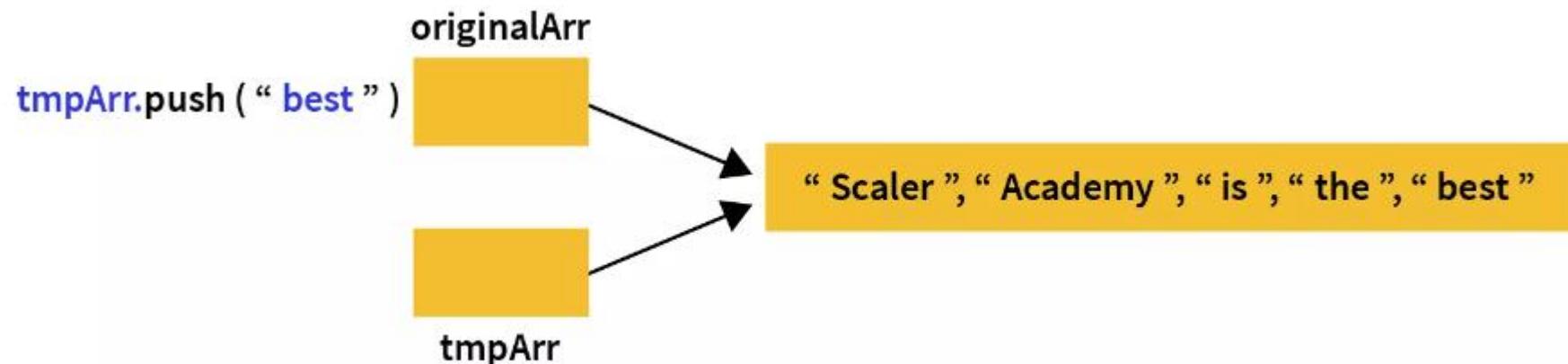
```
<script>
let originalObj = {
    name: "Yael",
    rating: 4.5,
    topic: "JavaScript"
};

function demo(tmpObj) {
    tmpObj.rating = 5;
    console.log(tmpObj.rating);
}

console.log(originalObj.rating);    // 4.5
demo(originalObj);                // 5
console.log(originalObj.rating);    //5
</script>
```



```
let originalArr = ["Client", "Course", "is", "the"];  
  
function pushArray(tmpArr) {  
    tmpArr.push('best')  
    console.log(tmpArr);  
}  
  
console.log(originalArr);      // ["Client", "Course", "is", "the"]  
pushArray(originalArr);       // ["Client", "Course", "is", "the", "best"]  
console.log(originalArr);     // ["Client", "Course", "is", "the", "best"]
```



שאלות?