Javascript DOM

http://www.nathankrasney.com/

דוגמא לדף HTML עם Javascript

```
<head>

</head>

<body>

  <button onclick="clickHandler()">Click Me</button>

  <p id="id1">This text is static</p>

  <script>

    function clickHandler() {

      document.getElementById('id1').innerHTML = 'This text is dynamic'; }

  </script>

</body>
```

שם מאפיין DOM של אלמנט אשר מייצג שם מאורע - event

פונקציה אשר נרשמת בתוך תגית script ויקראו לה כאשר יתרחש המאורע - במקרה זה כאשר ילחצו על הכפתור הזה

מאפיין DOM של אלמנט אשר מייצג את התוכן - content שלו

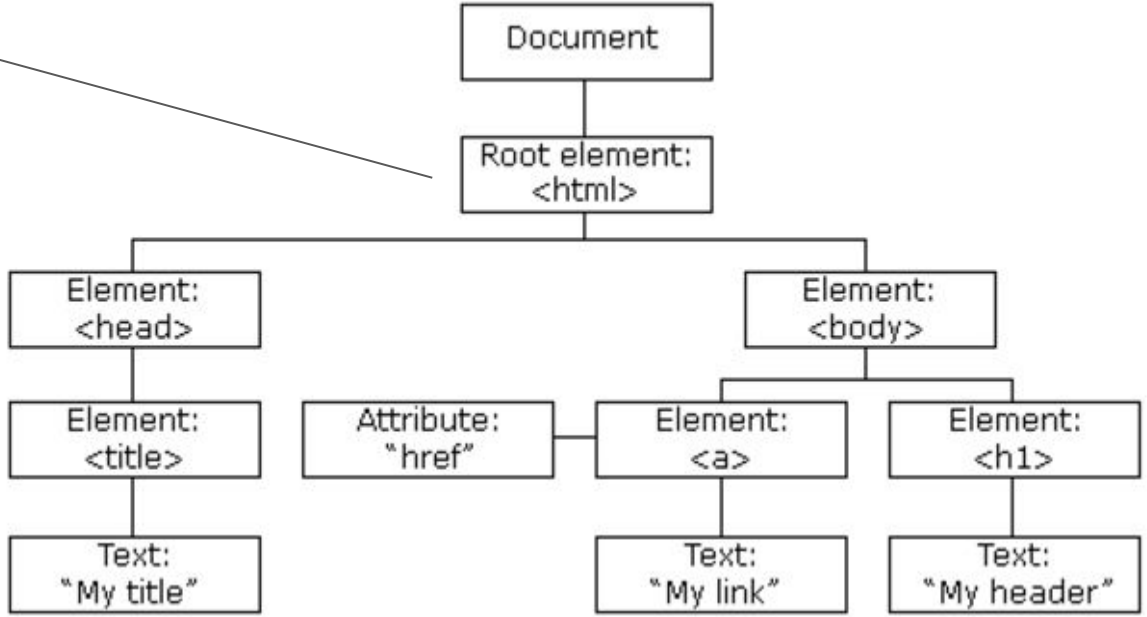
אוביקט שמיצג דף HTML אחרי שנטען לדפדפן

פונקציה של documentשמקבלת id של אלמנט ומחזירה אוביקט שמיצג את האלמנט

http://www.nathankrasney.com/

יצוג דף HTML כעץ אוביקטים ב DOM

```
<html>
<head>
    <title>My Title</title>
</head>
<body>
    <h1>My header</h1>
    <a href="#">My Link</a>
</body>
</html>
```



http://www.nathankrasney.com/

מה Javascript יכול לעשות בעזרת HTML DOM

1. לשנות את כל האלמנטים בדף HTML
2. לשנות את כל מאפייני האלמנטים בדף HTML
3. לשנות את כל ה css style בדף (זה הרי מאפיין של אלמנט)
4. להסיר אלמנטים ומאפיינים של אלמנטים
5. להוסיף אלמנטים ומאפיינים של אלמנטים
6. להאזין לכל המאורעות - events של כל האלמנטים (לדוגמא onclick)
7. להוסיף מאורעות חדשים בדף
8. לקבל מידע על האלמנטים והמאפיינים שלהם

DOM נותן לנו פעולות Create\Read\Update\Delete - CRUD על האלמנטים והמאפיינים שלהם בדף ה HTML

תגית script

תגית script מאפשרת לנו לכתוב קוד של javascript

נהוג למקמה בסוף אלמנט body על מנת שלא תעכב את עלית הדף

```
<html>

<head></head>

<body>

<script>

    alert("hello world");

</script>

</body>

</html>
```

בהמשך נראה שניתן לכתוב
קוד גם בקובץ javascript
ונראה איך דף ה HTML
מתייחס לקובץ החיצוני הזה

http://www.nathankrasney.com/

דוגמא נוספת - שינוי style

```
<body>

    <button onclick="clickHandler()">Click Me</button>

    <p id="id1">This text is static</p>

    <script>

        function clickHandler() {

            document.getElementById('id1').style.color = 'red';

            document.getElementById('id1').innerHTML = 'This text is dynamic and red';}

    </script>

</body>
```

רושמים קוד של javascriptבתוך תגית script . נהוג למקם את התג הזה בסוף הbody על מנת שלא יעכב את עלית האלמנטים והתצוגה

שינוי צבע באמצעות JS וזאת דרך מאפיין DOM של style ודרכו מאפיין DOM בשם color

http://www.nathankrasney.com/

output - כתיבה

אפשר לכתוב מידע למסך בכמה דרכים

דוגמה	מימוש	אמצעי כתיבה
document.getElementById("someId").innerHTML=”newText”;	innerHTML	לתוך התוכן - content של האלמנט
document.write(3+3);	document.write()	לoutput של הדף
window.alert("pop up message");	window.alert()	לתיבת הודעות
console.log(“debug info”);	console.log()	לconsole של הדפדפן. לחיצה על F12 כדי לראות את המידע

מאורע - event

מאורע מתרחש בדרך כלל עקב אינטראקציה של המשתמש עם דף ה HTML לדוגמא דף הhtml סיים לעלות , משתמש לחץ על כפתור וכך הלאה.

Javascript מאפשר לרשום JS קוד אשר יופעל כשהמאורע יתרחש

```
<element event=’javascript code’>
```

להלן שתי צורות לטפל במאורע : אחת עם רישום שם הפונקציה (יש להגדירה בתגית script) והשנייה ללא

```
<body>

  <button onclick="alert('i was clicked')">Click Me</button>

  <img src="https://homepages.cae.wisc.edu/~ece533/images/boat.png" onclick="boatClickHandler()" alt="boat"/>

</body>
```

http://www.nathankrasney.com/

רשימת מאורעות נפוצים

הסבר	שם מאורע
מתרחש כשהיה שינוי באלמנט לדוגמא הכניס טקסט לאלמנט input	onchange
מתרחש בעקבות לחיצת משתמש על אלמנט לדוגמא כפתור	onclick
מתרחש כשהמשתמש עובר עם העכבר מעל לאלמנט לדוגמא מעל לינק	onmouseover
מתרחש כאשר הדפדפן סיים להעלות את הדף	onload
מתרחש כשהמשתמש לוחץ על מקש לדוגמא בתוך אלמנט input	onkeydown

HTML DOM

HTML DOM מכיל מאפיינים ופונקציות ש javascript משתמש בהן לדוגמא :

getElementById היא פונקציה של HTML DOM

innerHTML הוא מאפיין של HTML DOM

HTML DOM Document

document הוא האובייקט שמיצג את הדף והוא מכיל פונקציות בקטגוריות שונות

דוגמה לפונקציה	קטגוריה
document.getElementById(id)	מציאת אלמנט
element.setAttribute(attribute, value) אפשר גם להשתמש במאפיין לדוגמא element.id או element.style.color	שינוי אלמנטים
document.getElementById(id).onclick = function(){code}	הוספת handler למאורע
document.body	מציאת אובייקטים (בשימוש מאפיין)

אוביקט JS אשר מייצג אלמנט HTML

פונקציות הוספה והסרה של אלמנטים מן ה DOM

document הוא אובייקט שמייצג את הדף והוא מכיל פונקציות בקטגוריות שונות

אפשר גם להוסיף אלמנט בצורה פשוטה דרך innerHTML של האבא

createElement מופעל תמיד על document

דוגמה לפונקציה	קטגוריה
document.createElement(tag name) ,document.body.appendChild(element)	הוספת אלמנט
element.remove()	מחיקת אלמנט

appendChild מופעלים תמיד על אלמנט האבא .בדוגמא הזו הנחתי שאלמנט האבא הוא body . הערה, אפשר להוסיף בקלות דרך innerHTML

http://www.nathankrasney.com/

דוגמה לשימוש ב createElement

דוגמה ליצירת אלמנטים בצורה דינמית

https://github.com/NathanKr/VS2015GitRep/blob/master/Telrad/Web/Javascript/js_create_element.html

אפשר גם לעשות בצורה פשוטה יותר באמצעות הוספה ל innerHTML

Element selectors

Name	Description
getElementById	מחזיר את האלמנט עם id שזהה למה שהועבר כארגומנט
getElementsByClassName	מחזיר את כל האלמנטים עם class name אשר זהה למה שהועבר כארגומנט
getElementsByTagName	מחזיר את כל האלמנטים עם tag name אשר זהה למה שהועבר כארגומנט
querySelector	מחזיר את האלמנט הראשון עם css selector שזהה למה שהועבר כארגומנט
querySelectorAll	מחזיר את כל האלמנטים עם css selector שזהה למה שהועבר כארגומנט. מה שמוחזר הוא NodeList וזה לא מערך !! ניתן להמיר למערך בעזרת Array.from

http://www.nathankrasney.com/

Sample using querySelectorAll

```
<body>
 <p id='id1'>1</p>
 <p id='id2'>2</p>
 <p id='id3'>3</p>
 <button onclick="list()">List elements</button>
 <script>
  function list(){
   var nodeList  = document.querySelectorAll('body > p');
   for (var index = 0; index < nodeList.length; index++) {
    const element = nodeList[index];
    console.log(element.id , element.innerText);

   }
  }
 </script>
</body>
```

querySelectorAll returns a **list** not an array .We can iterate it using for loop but not do push or pop as we can for array

http://www.nathankrasney.com/