

# javascript

שפה המאפשרת הוספת אינטראקטיביות לדפי HTML.  
HTML, CSS, JAVASCRIPT מהוות יחד שפות צד לקוח,  
כלומר שפות שיוצאות לפועל באמצעות הדפדפן, ללא צורך  
בשימוש בשרת.

# מיקום הקוד

את הקוד מכניסים בתוך תגית `<script>` וכמובן סוגרים בתגית `</script>` לאחר מכן. כמו כן, אפשר לקשר לקובץ js חיצוני כמו בעבודה עם CSS.

```
<script src="js/myscript.js"></script>
```

במידה ויש לנו פקודות שפועלות לבד מיד בעליית העמוד, נעדיף למקם את הקוד בסוף איזור body לפני סגירת התגית. זאת כדי להימנע ממצב שאנו מבקשים מjs להריץ פקודה על אלמנט כלשהו והאלמנט עצמו עדיין לא נטען בכלל בדפדפן.

במידה ויש לנו מספר פונקציות – נעדיף להימנע מעומס בדף הhtml ולרכז את כולן בקובץ js נפרד אליו נקשר את העמוד שלנו בhead.

# העיקרון הבסיסי בJS

העיקרון הבסיסי איתו עובדת JS הוא שישנם אירועים שונים המתרחשים בעמוד האינטרנט שלנו, חלקם באופן אוטומטי וחלקם בעקבות פעולות המשתמש, ונוכל לזהות מתי הם מתרחשים ולהפעיל בהתאם כל מיני פקודות שונות, וכך ליצור אינטראקטיביות (=תגובתיות ותקשורת) עם הגולש שלנו.

## אירועים <--- מפעילים פונקציות <--- מבצעות פקודות

לדוגמא – אירוע של שליחת טופס.

נוכל לזהות שהתרחש אירוע כזה ולהפעיל בדיקה שתבדוק האם כל השדות שרצינו שהמשתמש ימלא באמת מולאו.

אם כן – נוכל לכתוב למשתמש הודעת תודה ולידע אותו שהפרטים נשלחו בהצלחה.  
אם לא – נוכל להודיע למשתמש שיש שדות שחובה עליו למלא והטופס לא יישלח בלעדיהם.

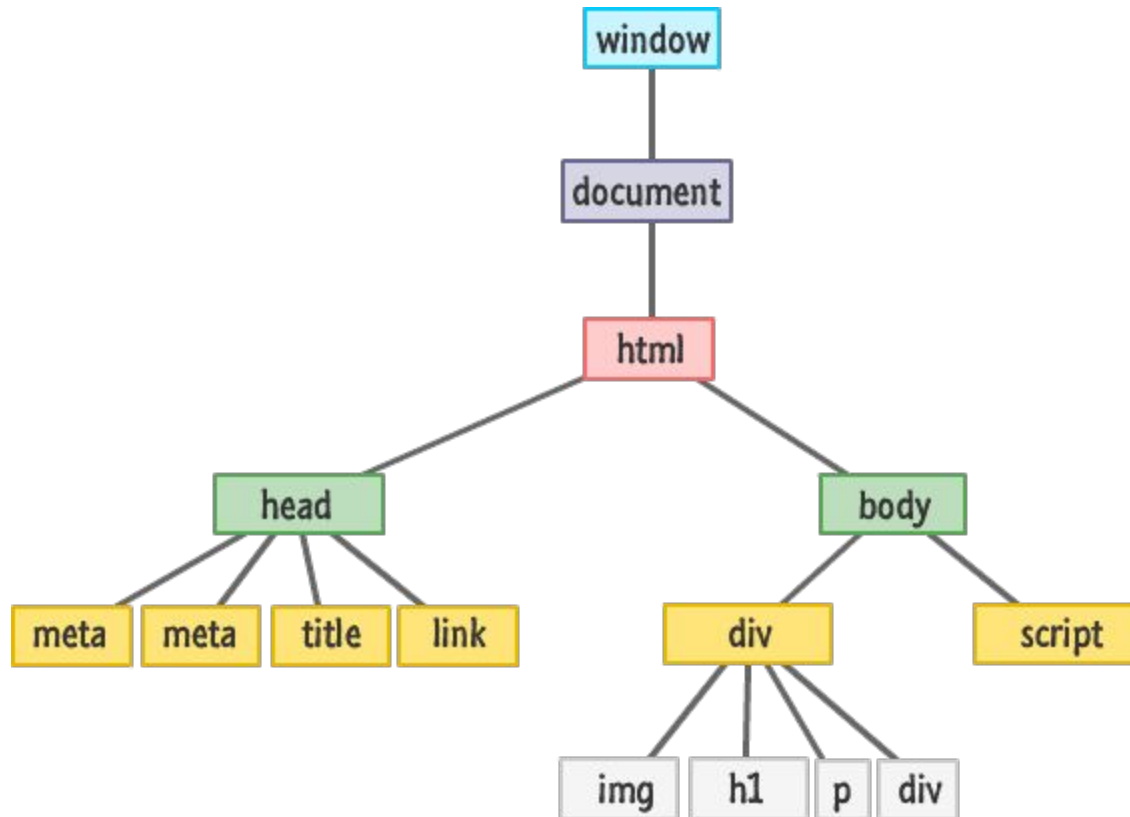
אירועים נפוצים נוספים הם לחיצת עכבר (קליק), דאבל קליק, מעבר עכבר על אלמנט, לחיצה על מקש במקלדת, שינוי גודל החלון, פקודת הדפסה, פעולת גלילה עם העכבר ועוד.

# חוקי שפת JS

- שפת JS רגישה להבדלים בין אותיות גדולות וקטנות (case sensitive). לכן חשוב מאוד להקפיד על שמות שנכתבים בצורה זהה, ופקודות שמכילות אותיות גדולות וקטנות – שאם לא ייכתבו בצורה הנכונה פשוט לא יזוהו ולא יעבדו.
- כל פקודה מסתיימת תמיד ב ; ובד"כ נקפיד שכל פקודה תיכתב בשורה נפרדת.
- הפקודות מתבצעות לפי סדר הכתיבה, מהראשונה לאחרונה.
- יש אפשרות לכבות את הjs בדפדפן, ולכן חשוב להקפיד שהאתר ייראה טוב ויפעל תקין גם בלי הפונקציונאליות של הjs.

# DOM

מבנה המסמך וסדר האובייקטים בו נקרא DOM (document object model).  
כך למשל 'רואה' הדפדפן ושפת הjson את המבנה כשהעמוד עולה:



כלומר הDOM מהווה מפה ברורה של כל האלמנטים, מי בתוך מי = מי ה"הורה" של מי, ומי הילד של מי, מה מכיל כל אלמנט, ואיזה אטריביוטס יש לו.

# פנייה ושינוי ה-DOM

אחת הפקודות השימושיות ביותר כדי לפנות לאלמנטים ב-DOM ולבצע עליהם שינויים היא פקודת `document.getElementById` (יש לשים לב לאותיות הגדולות שבראש כל מילה מחוברת).

פקודה זו בעצם מבקשת גישה לכל המידע שישנו על אלמנט מסוים שאותו המסמך יאתר ויביא לנו ע"י ה-ID של האלמנט הרצוי. כך נוכל לשנות לאלמנט את התוכן, העיצוב והאטריביוטס השונים.

**פניה לתוכן האלמנט ושינוי שלו לתוכן אחר:**

```
document.getElementById("div_id").innerHTML = "some text";
```

**פניה להגדרות העיצוב של האלמנט ונתינת ערך שונה לאחד מהמאפיינים, למשל `color`, `fontSize` וכו':**

```
document.getElementById("div_id").style.color = "red";
```

**פניה לאטריביוטס של האלמנט (מאפיינים שנכתבים בתוך התגית כמו `title`, `src`, `href`, `alt`) ושינוי שלהם:**

```
document.getElementById("div_id").title = "div title";
```

פקודות שמתבצעות על אלמנט, כמו `getElementById`, נקראות ב-JS גם מתודות (method). ערכים (properties) שניתן לפנות אליהם או לשנות אותם, כמו `innerHTML`, נקראים ב-JS values.