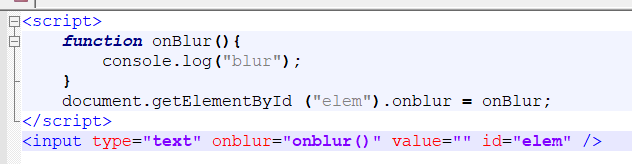
ארז אטון – 311530539, רפאל חנוכייב - 312064041

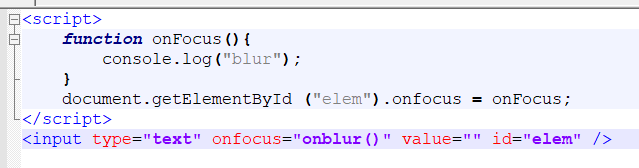
# פיתוח בסביבת אינטרנט - תרגיל 2 – חלק תאורטי

**JavaScript, DOM&BOM**

תאריך פרסום: 18.05 תאריך הגשה: 30.5

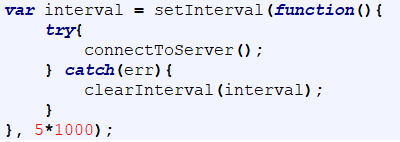
# חלק תיאורטי (50 נקודות) –

1. א. 



הפעולה הראשונה היא onBlur היא מתרחשת כאשר המשתמש מוריד את הפוקוס מאלמנט מסויים. הפעולה השנייה היא onFocus והיא בדיוק ההפך מonBlur והיא מתרחשת כאשר המשתמש נותן פוקוס לאלמנט מסויים.  
במקרה שלנו מדובר על שדה input אז בעזרת JS נוכל לדעת מתי המשתמש לחץ עליו ומתי המשתמש עזב אותו.

ב.

ישנה אפשרות של setTimeout אשר מבצע פעולה לאחר זמן נתון. כדי לממש התחברות לשרת כל 5 דקות עד שיהיה כישלון, אז נצטרך בכל ריצה של הcallback להגדיר timeout חדש שירוץ בעוד 5 דקות, ובמקרה של כישלון לא ליצור timeout חדש.  
האפשרות היותר נוחה היא של הפונקציה setInterval אשר מבצעת פעולה במרווחי זמן קבועים, ולכן נעביר לה פשוט את הפונקציה ו-5 שניות כערך, והיא תרוץ כל 5 שניות. ברגע שנקבל שגיאה אז נצטרך פשוט לאפס את האינטרוואל. זוהי אופציה הרבה יותר נוחה מאשר להגדיר מחדש בכל ריצה timeout חדש, ולכן עדיף להשתמש בה במצב כזה. נשתמש בsettimeout רק כאשר נרצה להריץ פעולה באופן חד פעמי לאחר זמן מסויים.  
דוגמא למימוש:  


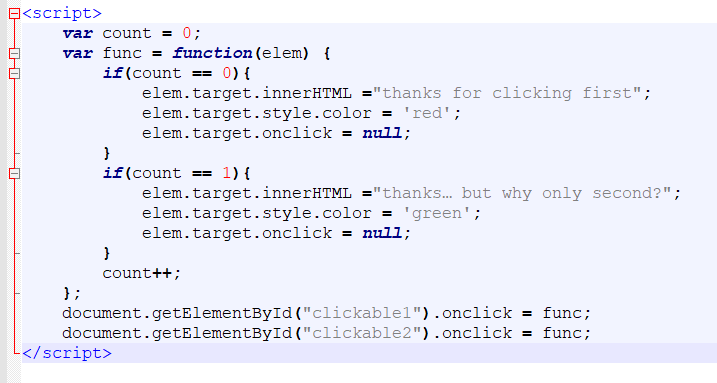
ג. JSON – ראשי תיבות של JS Object Notation, בגדול מדובר בטקסט אשר מכיל אובייקט ומערכים של JS. משתמשים בו כדי להעביר מידע בין שירותים שונים בצורת טקסט (לדוגמא בAPIים שונים). היתרון בו שהוא מאוד נוח לקריאה, והוא דיי מינימליסטי ולא תופס הרבה נפח מיותר.

Anonymous Functions – פונקציה אנונימית היא פונקציה שהעברנו ישירות ל-callback מסויים ולא הגדרנו לה שם ספציפי. לדוגמא בסעיף הקודם בדוגמא של setInterval השתמשנו בפונקציה אנונימית. השימוש בה הוא למקרים בהם אנחנו רוצים להעביר לוגיקה מסויימת אבל לא מדובר במשהו שיש לנו צורך להשתמש שוב בהמשך הקוד, אז נעדיף להשתמש בפונקציה אנונימית כדי לעשות את הקוד מובן יותר.

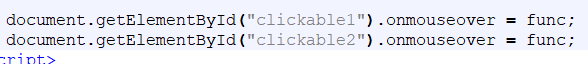
OnEvent – בJS אנחנו בעיקר מגיבים לאירועים שונים שהתרחשו בדפדפן, לדוגמא כאשר משתמש לחץ על כפתור, הזיז את העכבר וכו'. אנחנו יכולים להגדיר לאלמנטים שונים ב-DOM אוונטים שונים שיתרחשו במקרים שונים (onclick,onblur,onsubmit וכו'), ועבור כל איוונט להגדיר פונקציית callback שתרוץ.

BOM – תו יוניקוד שנמצא בתחתית המסמך שמטרתו לסמן את סוג הקידוד של המסמך. הבעיה עם זה שהוא שומר תו בתחילת הקובץ ואז זה עושה בעיות מוזרות, לדוגמא שאנחנו לא נוכל לשלוח headers לאחר שהתו הזה הודפס (כי הדפדפן כבר התחיל לשלוח תוכן), או שיופיע סימן מוזר בתחילת הדף. לכן יש צורך לשמור כל קובץ כ-UTF8 Without BOM

1. א.



ב.



1. א. לאחר הטעינה נראה כותרת, פסקה וכפתור. הפעולה שבכפתור מוגדרת לרוץ רק לאחר לחיצה על הכפתור, ואפשר ללחוץ על הכפתור רק אחרי שהדף יופיע, ולכן הפעולה בהכרח לא תרוץ לפני שהדף יעלה. (בנוסף הכפתור הוא האלמנט האחרון בדף, אז בהכרח כל הדף ירונדר לפני שהכפתור יוצג).

לאחר לחיצה על הכפתור הפלט במסך ישתנה ל-11, משום שהפונקציה document.write כותבת במקום הHTML הנוכחי שלנו.

ב. שיטת הפלט העדיפה בJS היא להדפיס את הקלט לאלמנט אחר שנמצא בDOM שלנו. ככה יש לנו שליטה איפה הפלט יופיע וכיצד הוא יוצג. לדוגמא:  


5. א.

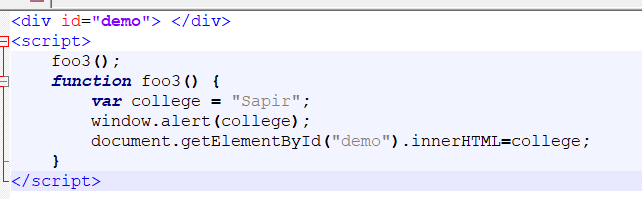
var json = JSON.parse('[{ "first\_name":"Erez", "last\_name":"Aton", "Phone":"052-6838263", "Address":"Sderot"},{ "first\_name":"Repahel", "last\_name":"Khanokayev", "Phone":"054-2355295", "Address":"Sderot"}]');

ב. 1. JSON תופס הרבה פחות מקום מ-XML שמכיל הרבה תוספות רק להגדרה של התוכן.

2. JSON יותר קריא לעין מאשר XML שיותר קשה להבין מה התוכן שלו.

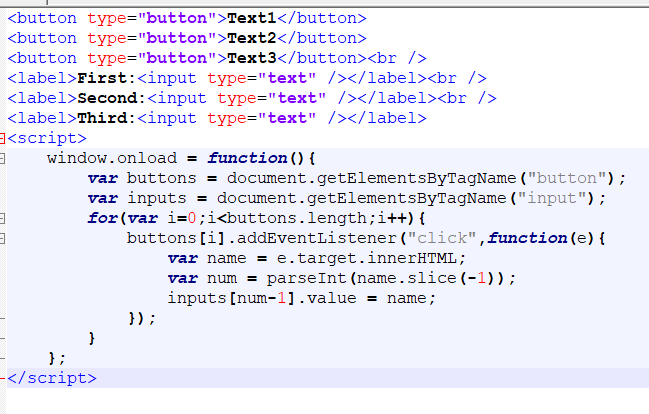
3. הרבה יותר קל לנתח JSON ולהמיר אותו למערך ואובייקט, מאשר XML שצריך להבין את המבנה שלו ולסרוק את התגיות שלו כדי לפענח אותו.

7. א. ראשית ב-alert ישנה שגיאה, משום שהמשתנה college מוגדר בscope שונה מהalert אז זה לא ירוץ. יש צורך להעביר את ההגדרה לאותו סקופ של הalert. בנוסף, הפונקציה getElementById תחזיר null מאחר ואין אלמנט כזה בדף. עלינו לשנות את המזהה כדי שהיא תמצא את האלמנט:

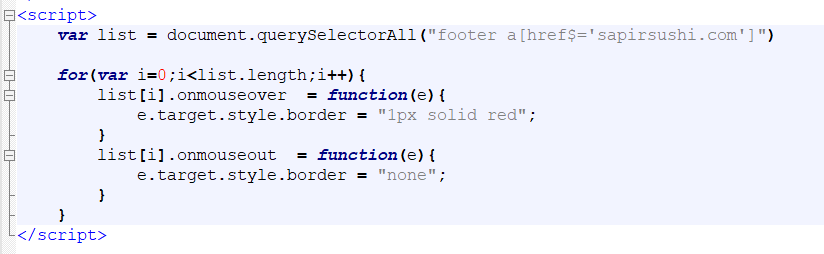


ב. קטע הקוד הזה יעבוד טוב, מאחר והגדרנו את המשתנה בglobal scope אז ניתן לגשת אליו גם מתוך פונקציות, אז לאחר ההגדרה בפונקציה foo4 נוסיף למשתנה את הערך " college" ואז נעביר את זה לalert שירוץ כמו שצריך.

8.



9. א.



ב. נוסיף תנאי בonmouseover:

