תרגיל ביניים - המשך:

טוב אז אחרי שבניתם את המחשבון המדהים בתרגיל ה CSS & HTML הגיע הזמן לגרום לו לעבוד עם קצת TS מהסרטים 😉.

בתיקייה שיצרתם את הHTML ואת ה CSS שלכם, תיצרו את הקבצים הבאים:

entities.ts - קובץ שיכיל את הישיויות שבהם אנחנו נשתמש

calculator.ts - קובץ שיכיל את הפעולות בהן נשתמש

כמו שאתם יודעים, הדפדפן לא יודע לקרוא את קבצי הts ולכן צריך לקמפל אותם לקבצי js.

איך???

tsc XXXXX.ts

פקודת הtsc מקמפלת את קבצי ה ts ולאחר מכן שם הקובץ שלנו אומר לה מה לקמפל.

ממש אחרי שתריצו את הפקודה אתם תראו שנוצר קובץ חדש בשם: XXXXX.js.

מעולה!!!!

עכשיו יש עם מה לעבוד 👌.

לקובץ שנוצר צריך לשים reference מהhtml שלנו (אם אתם לא יודעים איך זה הזמן ללכת לקרוא).

אחרי שהוספנו את הreference לקובץ שנוצר אפשר להתחיל להוסיף דברים.

שימו לב ❤!!

**הreference הוא לקובץ הjs ולכן כל שינוי בts לא יהיה מוכר לhtml כל עוד לא עשינו tsc מחדש!!!!!**

1. יש להוסיף למחשבון פונקציונליות של חישוב כך שהוא יאגור את כל הפרמטרים שמכניסים ויציג את התוצאה רק לאחר שלחצנו על '=' (כמובן להתחשב בסדר פעולות חשבון).

לדוגמא: כאשר המשתמש יזין 3 \* 3 + 3 הוא יקבל 12 כתוצאה ולא 18.

1. יש להוסיף פונקציית sum אשר מקבלת מספר לא ידוע של פרמטרים (כולם מספרים, אין צורך לבדוק תקינות) וצריכה להחזיר את הסכום שלהם. לדוגמא:

sum(1,2,3,4) -> 10

sum(1,1,1) -> 3

1. יש להוסיף פונקציית sum אשר יכולה לקבל מספר לא ידוע של פרמטרים (גם כן מספרים, לא צריך לבדוק תקינות) אך תוכל לקרוא לפונקציה כמה פעמים רצוף. מה זאת אומרת??? נוכל לקרוא לפונקציה כך:

sum(1,2,3)(1,1,2,3,null) === (1+2+3+1+1+2+3) -> 13

נוכל להמשיך לקרוא לפונקצייה כמה פעמים שנרצה עד שנגיע ל null באחד מהפרמטרים וכך נדע שהקלט הסתיים.

**\*\*\* יש לאפשר למשתמש להכניס את אותם ערכים גם באמצעות המקלדת!**

**דגשים:**

* אסור להשתמש בeval
* לחשוב על מקרה קצה - אופרטור ואז נקודה, אופרטור אחרי אופרטור וכו'