**15.01**

שפת CSS (**Cascading Style Sheets**). שכבת קוד שאחראית ספציפית על העיצוב, לטובת סדר (שהקוד הראשי יהיה מורכב רק מהתוכן ולא מהעיצוב הארוך). לדוגמה אם רוצים לשנות את עיצוב האתר אז יהיה קובץ נפרד שאותו אפשר לשנות בלי לשנות את התוכן.

יש דרגות עיצוב בקוד. הכי חלש הוא יבוא של קובץ חיצוני (CSS). מה שידרוס אותו הוא כתיבה של עיצוב בתוך הhead. הדרגה השלישית (החזקה יותר, שתדרוס את קודמיה) היא תכנות של עיצוב בתוך הbody. שלושתם יכולים לשלוט על אותם תכונות בעמוד הסופי.

**17.01**

כשמגדירים לdiv כלשהוא position: absolute – אז לא נכתוב margin-top לדוגמה אלא top, כי זה תמיד יהיה ביחס לדיב הראשי. לעומת ברירת המחדל שהיא position: relative שבמצב כזה אם אני אגדיר ערכים של מיקום זה יהיה ביחס לעמוד ולא לdiv שהוא נמצא בתוכו.

בעזרת <span> אני אעטוף מילה או כמה מילים מתוך משפט שלם שאני רוצה להתייחס אליו בנפרד בcss (נגיד להגדיל רק את המילה הזו)

כמה סוגי position (דוגמאות ויזואליות בgit בdisplay)  
static – נשאר באותו מקום שאמור להיות על פי שאר האובייקטים. אי אפשר להזיז.  
relative – זז ביחס למיקום שהיה אמור להיות האובייקט ע"פ שאר האובייקטים.  
absolute – מתעלם מאובייקטיבים אחרים, מתייחס רק לעמוד.  
fixed – כמו absolute אבל מתייחס רק לview point, ז"א נשאר תמיד במיקום הנראה לעין גם אם גוללים את המסך וכו'.  
sticky – יש לו מיקום אבל אם עוברים אותו הוא ישאר למטה (כמו שורה ראשונה מקובעת בexcel)

**22.01**

Bootstrap. חוסך הרבה דברים שלמדנו לעשות בCSS, ולהשתמש במשהו שכבר מוכן. הוא רספונסיבי לגמרי, ז"א מתאים את עצמו לכל המכשירים והמסכים מבחינה עיצובית. מקשרים את הhtml לקישור חיצוני של bootstrap.

גודל ברירת מחדל של אובייקט בbootstrap הוא 12. (לא פיקסלים אבל כשמציינים בclass זה יכול להתחלק ב12 יחידות, אא"כ ממלאים את כל רוחב המסך בהגדרות הclass).

כשרוצים להכניס אובייקטים מכניסים כמו בבושקה בclass: container.

**24.01 – מבוא לתכנות**

כל כתיבת תוכנה מתחילה בתרשים זרימה. לכל תוכנה יש התחלה וסוף שרושמים קודם כל בתכנון התרשים. בינתיים אנחנו לא מפצלים מסלולים. גם אם יש 2 פלטים, הם ישורשרו אחד אחרי השני.

**29.01**

Modulo (%) נותן את השארית שנותרת אם מחלקים מספר במספר אחרי המודולו. אמנם חילוק (/) בקוד לא נותן שארית, אבל אם מוסיפים נקודה עשרונית במחלק\מחולק אז גם התשובה תהיה עשרונית.

**31.01**

הקיצור של num+=something יכול להיות גם עם סימני חשבון אחרים: num-=something , num\*=something, num/=somthing.

לולאת while מתאימה יותר במצב שאנחנו לא יודעים כמה איטרציות צריכות להיות, ולולאת for במצב שיש לנו את הגבול של איטרציות.

**05.02**

בJS כדי לקבל קלט ולהכיר בו כמספר, בזמן הכרת המשתנה, להוסיף + צמוד לפני הפקודה (לדוגמה prompt) ואז המחשב קולט את זה כnumber.

**19.02**

Pop אפשר לעשות גם על string? כי הרי הוא לא עשה jsnon.parse לפני שעשה pop. – לא. הוא עשה Pop על המערך שהיה שמור על משתנה (בעצם לפני שעשה רענון לדף).

כדי להוציא אובייקט מתוך מערך, אפשר להשתמש בפונקציה filter.

**26.02**

map, reduce, filter. Filter רק מסנן אבל לא יכול לכלול גם פקודות. אפשר לעשות map על filter על מערך.

url לקבלת מידע חיצוני (API) נקרא גם end point.

**28.02**

כשאני מגדיר משתנה בconst בתוך פונקציה – המשתנה נשמר רק בתוך הפונקציה. ברגע שהפונקציה תסתיים המשתנה לא ישמר בזיכרון. אז למה בכל זאת נכתוב const בפונקציה? שפה משותפת של תכנתים כדי להדגיש שהמשתנה הזה לא אמור להשתנות במהלך הפונקציה ולא יהיו עליו שינויים אלא רק שימושים בו.