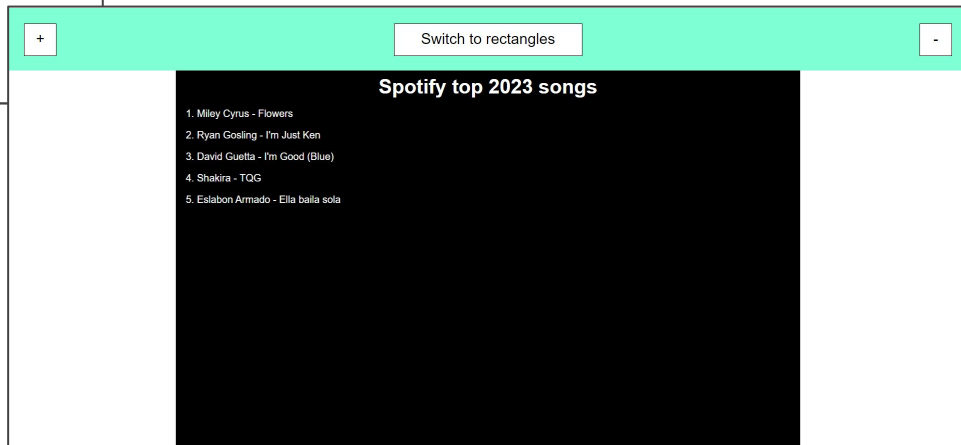
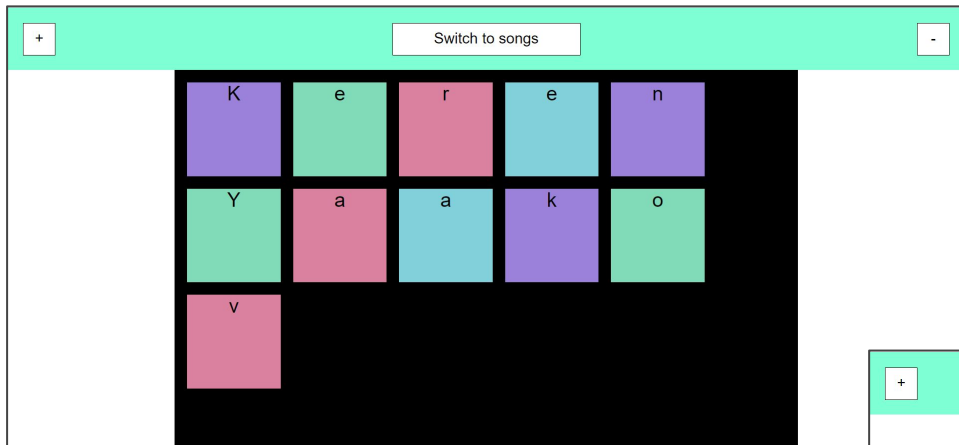


תרגיל 3 להגשה – Javascript

כל הזכויות שמורות לקרן יעקב 2024

תרגיל 3 להגשה – דוגמה



הגשת תרגיל 3 – זמנים ודגשים

- תחילת התרגיל – 3.6.24
- תאריך אחרון להגשה – 17.6.24 – 6:00
- יש שבועיים, אבל **לא לחכות לרגע האחרון**, התרגיל לא קצר ולוקח כמה שעות גם למתכנתים מנוסים
- אין הגשות לאחר המועד האחרון להגשה, במקרים **מיוחדים** (כמו מילואים) לפנות למרצה.
- בתוך תיקיית dev_XX (שכבר פתחתם) תהיה תיקיה בשם ex3 ושם יהיו הקבצים של הפרוייקט
- אין העתקות
- בתרגיל הנוכחי יש אפשרות לקבל 105 נקודות

התרגיל לא יבדק במקרים הבאים:

- אין דרישות מינימום HTML
- אין לינקים במודל לדף הבית של תרגיל 3 (הקוד ב-FTP) ולריפודיטורי בגיטהאב
- הגשה באיחור ללא אישור
- חשד להעתקה

הגשת תרגיל 3 – שינויים בתרגיל זה

- יום לפני סוף ההגשה לא יהיה מענה של הצוות לשאלות בנוגע לתרגיל, זאת אומרת ב- 16.6.24 כבר לא נענה. הסיבה: התרגילים המוגשים בלחץ של זמן פחות טובים בדר"כ
- גיט וגיטהב כבר לא בונים, נשתמש ב-Github Classroom. עליכם להכנס ללינק <https://classroom.github.com/a/zOaoFzWW> ושם תקבלו תיקיית גיט פרטית (private) עם קוד התחלתי לתרגיל
- חובה לעשות קומיטים במשך כל התרגיל, מינימום 10 קומיטים משמעותיים, יבדקו גם הודעות הקומיטים.
- בלי לינקים במודל לדף הבית של תרגיל 3 (הקוד ב-FTP) ולריפוזיטורי בגיטהב – התרגיל לא יבדק. הקוד יהיה זהה בשני המקומות!

מטרות תרגיל 3

- תרגול החומר הנלמד - HTML Tags, חלוקה לאיזורים, עיצוב CSS ומיקומים, JSON, Javascript
- המרת הוראות כתובות לקוד: HTML, CSS, Javascript
- מבנה דף HTML ודרישות המינימום
- חשיבה על המיקומים של האובייקטים
- שימוש ב-JS בשביל להטמיע פונקציונליות
- קוד נקי, מסודר, הזחות, שמות משמעותיים
- שימוש בגיט ובגיטהאב
- חיפוש חומר-עזר באינטרנט ושימוש בו
- התרגיל משלב את כל החומר שלמדנו עד היום, כדאי לעבור שוב על המצגות לפני תחילת העבודה על התרגיל

הוראות תרגיל 3 (1) – כללי

בתרגיל זה יש עמוד HTML, עמוד CSS, עמוד Javascript ועמוד json, כל אחד במקומו מבחינת היררכיית התיקיות. את הצבעים של כל האלמנטים אתם יכולים לבחור, חלק מהגדלים גם נתונים לבחירתכם, את המבנה והתוכן של השירים ב-JSON אתם תקבעו, אבל צריך להיות הגיוני, יעיל ולפי הדרישות

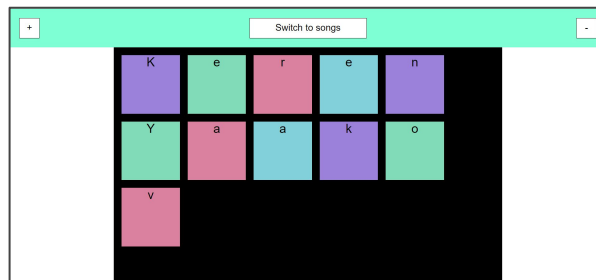
הרוחב של השורה העליונה 100% והאורך 100 פיקסלים

הרוחב של התוכן הוא 1000 פיקסלים והאורך המינימלי מגיע בדיוק עד לתחתית המסך (בבל גודל של מסך), כאשר יכולה להתווסף גלילה אנכית בהמשך, עם הוספת ריבועים.

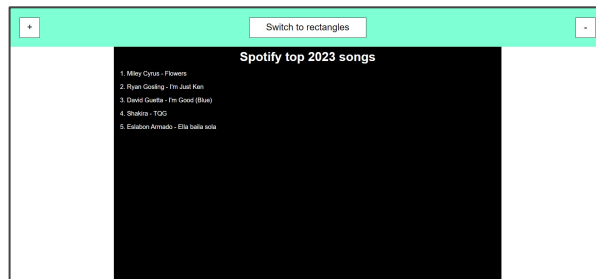
גודל כל ריבוע באיזור התוכן הוא 150 פיקסלים, המרחקים ביניהם 20 פיקסלים.

הוראות תרגיל 3 (2) - HTML

זהו עמוד ה-HTML. בלחיצה על הכפתור האמצעי "מתחלף" התוכן בין שתי תצוגות, הלוח ושוב:



יש תצוגה של ריבועים
(ברירת המחדל)



יש תצוגה של שירים
(שמגיעים מה-JSON)

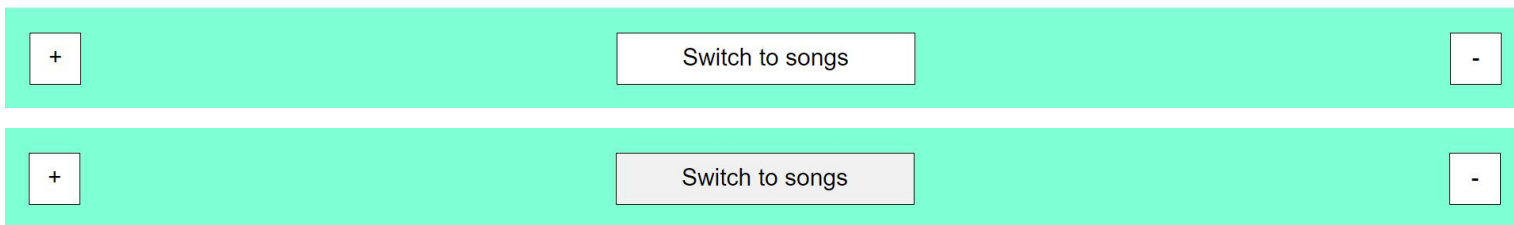
השורה העליונה של העמוד נשארת כמעט אותו הדבר בשתי התצוגות השונות, חוץ מהכיתוב על הכפתור שמתחלף

הוראות תרגיל 3 (3) – שורה עליונה

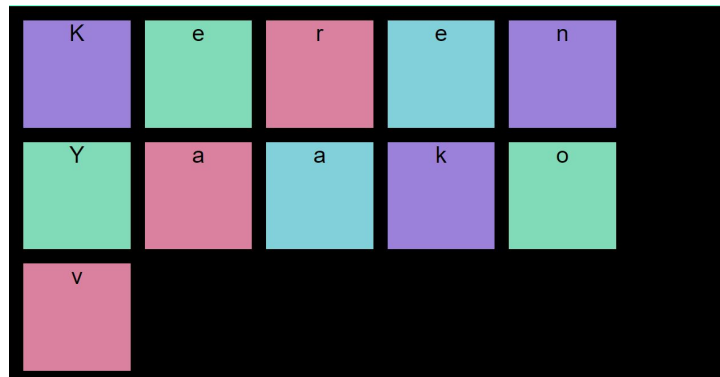
בראש עמוד ה-HTML יש שורה עליונה שתמיד נשארת במקומה (צילום מסך – שורה עליונה למטה). בשורה זו יש 3 אלמנטים לחיצים, מעבר עכבר עליהם מחליף את צבע הרקע (צילום מסך – שורה תחתונה למטה).

על האלמנט השמאלי מופיע "+" והוא מוסיף ריבוע אחד בכל לחיצה עליו, רק כשאנחנו בתצוגת ריבועים על האלמנט הימני מופיע "-" והוא מוריד ריבוע אחד מהסוף בכל לחיצה עליו, רק כשאנחנו בתצוגת ריבועים על האלמנט האמצעי מופיע בהתחלה הכיתוב "Switch to songs". לחיצה עליו תחליף את התצוגה לתצוגת שירים והכיתוב יתחלף ל-"Switch to rectangles". לחיצה נוספת תחליף בחזרה את הכיתוב ותחליף את התצוגה, וכך הלאה.

בזמן החלפת התצוגות, התוכן של התצוגה הנגדית ישאר בדיוק אותו הדבר ולא יוכל להשתנות.



הוראות תרגיל 3 (4) – איתחול הריבועים



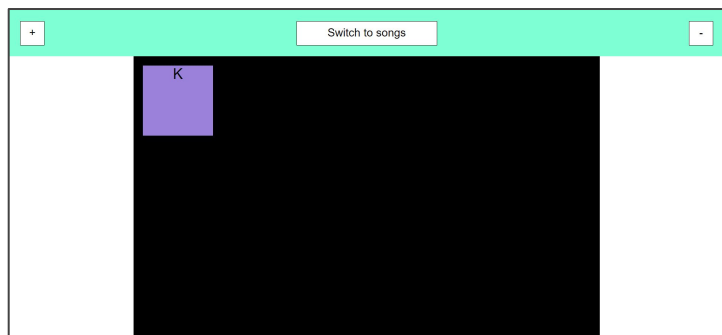
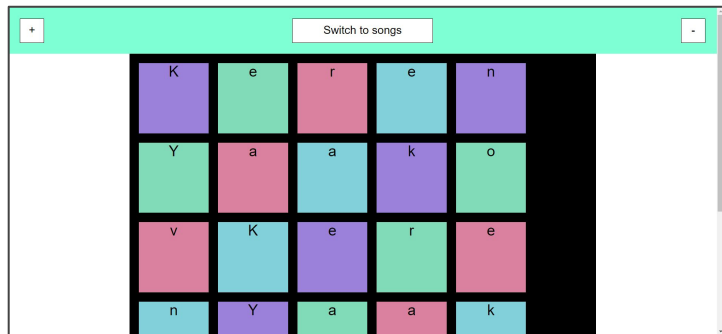
מספר הריבועים מאותחל למספר האותיות בשם המלא שלכם, ללא רווחים.

על כל ריבוע תהיה אות אחת, לפי הסדר בשם שלכם.

הריבועים יסתדרו בשורה, כל עוד יש מקום. אם נגמר המקום השורה הריבועים ימשיכו בשורה הבאה, וכך הלאה.

הצבעים ילקחו ממערך שבו יש מספר צבעים קבוע ומוגדר מראש (בדוגמה יש 4 צבעים), ובכל פעם שנגיע לסופו – נחזור לתחילת המערך ונתחיל שוב מהצבע הראשון.

הוראות תרגיל 3 (5) – כפתורי הריבועים

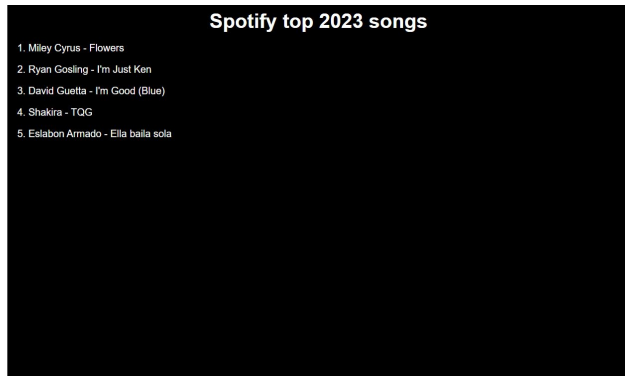


לחיצה על כפתור ההוספה תוסיף ריבוע בסוף, עם הצבע הבא בתור במערך ועם האות הבאה בתור בשמכם. אם נגמרו האותיות של השם – חוזרים להתחלה ומתחילים את השם מחדש. כשצריך תתווסף גלילה אנכית לעמוד.

לחיצה על כפתור ההורדה תוריד ריבוע מהסוף. כשיגמרו הריבועים המסך ישאר "ריק" מריבועים (בדוגמה רק רקע שחור) והכפתור לא יעשה כלום (לא יזרוק שגיאה)

לחיצות על הכפתורים אמורות לגרום לאותה תוצאה על המסך. כלומר אם מורידים את כל הריבועים ומוסיפים אותם בחזרה – האותיות והצבעים יהיו בדיוק אותו הדבר כמו בהתחלה.

הוראות תרגיל 3 (6) – תצוגת השירים



התוכן של תצוגת השירים מגיע כולו מקובץ json שנמצא בתיקיית data בפרוייקט.

ל-JSON יש שני מפתחות ראשיים – הכותרת ומערך שירים. לכל שיר יש id, artist, song name

יש לקחת את התוכן, את הכותרת לשם בתגית `h1` ואת השירים לסדר ברשימה לא ממויינת – עם `3` הפרטים שיש לכל שיר.

אתם יכולים לבחור את הטקסט של הכותרת והשירים, ואת מבנה ה-JSON כל עוד הוא הגיוני.

חלוקת הניקוד

- 35 נק' - פונקציונליות של טעינת הריבועים, הוספת ריבוע, הורדת ריבוע, החלפת תצוגות
- 20 נק' - מבנה קובץ JSON, שליפה והכנסה לתוך רשימה וכותרת
- 20 נק' - קוד מסודר ונקי, שמות משמעותיים, ללא הערות בקוד, יעילות (כולל שימוש חוזר בקוד)
- 15 נק' - קומיטים משמעותיים במהלך העבודה (לפחות 10) מהתחלת הפרוייקט
- 10 נק' - צמצום קלאסים ו-ids, שימוש בהיררכיות CSS וסלקטורים (במידה)
- 5 נק' בונוס - הוספת שדה בשורה העליונה בו המשתמש יקליד את השם המלא ורק אז נאתחל את הריבועים. נעשה זאת רק פעם אחת באיתחול העמוד ואח"כ השדה יהפוך ללא פעיל

התרגיל כולו הוא 10% מהציון הסופי בקורס.
הבונוס הוא הכל-או-כלום - צריך לעשות את הבונוס במלואו

הדרכה ועזרה (1) – ההגשה

- ההגשה ביחידים.
- אין הגשות באיחור, לא תהיה הארכה.
- יש להכנס ללינק בתחילת המצגת וכך לקבל תיקיית גיט עם הקבצים ההתחלתיים של הפרוייקט
- ההגשה של הקוד כולו נעשית ל-FTP ולגיטהב (אותו קוד בשני המקומות) במודל חובה לפרסם בטקסט:
 - הקישור לדף הבית של התרגיל השלישי
 - הלינק לתיקיית Github
- לאחר כל עדכון בשרת ה-FTP יש לעשות בדיקות QA ולוודא שהכל מוצג טוב. לתקן במידת הצורך ולבדוק שוב.
- אם אתם לא רואים את השינויים לאחר החלפת הקובץ בשרת ה-FTP – נקו את ה-cache ורעננו.
- בדאי לבדוק שוב **לפני שההגשה מסתיימת** שהתיקיה ב-FTP עדיין במקום ומוכנה להגשה.

הדרכה ועזרה (2) – הקוד

- כדאי להעביר את הקוד בוולידטור לפני ההגשה, גם לבדיקת HTML וגם לבדיקת javascript
- אסור שתהיה גלילה אופקית בדף, רק אנכית ורק אחת
- לא להשאיר שורות רווח **מיותרות**, לשמור על הזחות, שמות משמעותיים לפי הכללים שלמדנו
- יש רווחים שמותר להשאיר, כמו בין פונקציות, או חלקים בפונקציה שעושים דברים שונים.
- טעויות נפוצות מהתרגיל הראשון להגשה הועלו למודל כדי לעזור לכם להמנע מאותן הטעויות
- שימוש בהיררכיות של CSS – עד 3 רמות, או להגדיר אחרת את הכלל CSS
- מומלץ להעזר ב-dev tools של הדפדפן להגדרות CSS
- ההוראות בתרגיל צריכות להיות נכונות לכל גודל מסך של מחשב
- על כל אלמנט לחיץ סמן העכבר יהיה יד (לעומת חץ אם לא לחיץ)
- מומלץ קודם להגדיר איזורים, לתכנת אלמנטים סטטיים ורק אז להפוך את האלמנטים לדינמיים

הדרכה ועזרה (3)

- לא לערבב שכבות! למשל לא JS או CSS ב-HTML
- לתכנת בעזרת ES6 במידת האפשר (פונקציות חץ, string interpolation, הצהרה על משתנים וכו')
- לחשוב טוב מתי נגדיר משתנה לוקאלי ומתי גלובלי, מתי const ומתי let
- השתמשו במוסכמות שלמדנו לגבי שמות משתנים, שמות פונקציות
- יש לבצע במהלך העבודה קומיטים משמעותיים באופן סדיר. הודעת קומיט טובה היא כזו שמתארת לנו מה יש בקומיט כולו בלי להסתכל על הקוד ועל הקבצים.
- דוגמאות לשמות לא טובים: fix, update, second, finish
- דוגמאות לשמות טובים: fixed the footer icons css, fixed the js function save וכו'