**תרגיל בית 4**

|  |  |  |  |
| --- | --- | --- | --- |
| מקור | הערת משוב | האם התבצע שינוי בעקבות ההערה? | נימוק |
| סטודיו | נראות | כן | בוצע שינוי כללי בדף הבית |
| סטודיו | רק לציין שהחוות דעת מציינות כוכבים כלומר כמה שיותר אומר שהאתר יותר טוב | כן | עודכן |
| סטודיו | מחיר יקר | לא | המחיר נקבע על ידי בעל העסק ולא על ידי יוצרי האתר |
| סטודיו | הייתי מוסיפה רקע של תמונה לדפים | כן | נוספו תמונות |
| סטודיו | לשפר את נגישות חוות הדעת כך שיוסבר בצורה יותר נוחה למשתמש | כן | הובהר שמשמעות המספרים הינה כמות כוכבים |
| סטודיו | צבעים, סדר UX UI ושגיאות כשמכניסים אינפוט שגוי | כן | בוצע |
| סטודיו | אופי סידור האתר בפנים מבחינת עיצוב | כן | בוצע חילוק של עמודים |
| סטודיו | עיצוב של תמונות | כן | בוצעה האופציה להוסיף תמונה למוצר |
| סטודיו | צבעים נוחים יותר לקריאה | כן | הוחלפו |
| סטודיו | שימוש בצבעים שפחות מכסים את הטקסט | כן | הוחלפו |
| פרוייקט | האם יש בדיקת תקינות של כרטיס האשראי? | כן | נוספה בדיקה של תקינות אשראי ות.ז. |
| פרוייקט | תמונות באתר |  | נוספו |

**תבנית העיצוב:**

יצרנו מחלקת Singleton בשם Theme שאחראית לשינוי ערכת הנושא באתר.​

לTheme -  מתודה יחידה אשר משנה את ערכת הנושא בהתאם לאופציה שנבחרה בתפריט הנפתח.​

בכל שימוש בה, אנו קוראים לבנאי שלה, שמחזיר את המופע היחיד אם הוא קיים.​

**תכנות מקבילי:**

השימוש שלנו בתכנות מקבילי הוא: תוך כדי שהמשתמש מקליד את הטקסט בעמוד הזמנת המוצר, הטקסט מופיע בקאנבס מיידית. השתמשנו בזה כדי להעצים את חווית המשתמש, ועל מנת שיוכל לראות את השינויים במוצר הסופי בזמן אמת.

**Canvas:**

ב-JavaScript, קנבס הוא רכיב שמשמש ליצירת והצגת גרפיקה בדף אינטרנט. הקנבס מצייר או מקבל פעולות על אובייקטים גרפיים באמצעות קוד JavaScript. בעזרת הקנבס, ניתן ליצור תמונות, אנימציות, תרשימים ועוד. הקנבס מאפשר ציורים, צביעה, טקסט וקווים באמצעות מספר פונקציות שמאפשרות יצירת תוכן ושדרוגו באופן דינמי.

קנבס ב-JavaScript הוא אובייקט המשמש ליצירת והצגת גרפיקה דינמית בדף האינטרנט. נהוג להשתמש בקנבס כדי ליצור גרפיקה ואנימציות בדף האינטרנט, כגון תרשימים, תמונות מותאמות אישית, תיבות טקסט מעוצבות ועוד.

מתי כדאי להשתמש בקנבס ב-JavaScript?

1. כאשר רוצים ליצור גרפיקה מותאמת אישית ודינמית בדף האינטרנט.

2. כאשר רוצים ליצור אנימציות או אפקטים ויזואליים מעניינים עבור המשתמשים.

3. כאשר רוצים ליצור משחקים או יישומים אינטראקטיביים בדף האינטרנט.

כיצד משתמשים בקנבס ב-JavaScript?

1. יצירת קנבס: הגדרת אלמנט קנבס ב-HTML וקביעת מאפיינים כגון רוחב וגובה באמצעות JavaScript.

2. קביעת השטח הגרפי: קביעת השטח הגרפי בקנבס, שהוא השטח שבו מתבצעים הציורים והאנימציות.

3. שימוש ב-API: שימוש בפעולות וב-API המובנות של הקנבס כדי ליצור גרפיקה, לצייר צורות, להוסיף טקסט, לצבוע ולבצע פעולות אחרות.

4. אנימציה: יצירת אנימציות בעזרת פעולות ציור ושימוש בפונקציות של זמן כדי לשנות את התצוגה של הקנבס לאורך זמן.

בחרנו בנושא זה כי אחד מחברי הקבוצה ידע כי נושא זה עלול לשפר את עיצוב האתר באמצעות תכונותיו היחודיות.

הסבר על פעולותיו של הCanvas:

previewCanvas.getContext('2d')

שיטת השגת גישה לקונטקסט ה-2D של הקנבס. זה מאפשר לצייר גרפיקה באופן דינמי על הקנבס באמצעות פעולות כמו ציור צורות, הוספת טקסט וביצוע פעולות נוספות.

ctx.fillText()

משמשת להוספת טקסט לקנבס במיקום ועם תכונות טקסט מותאמות.

ctx.drawImage()

משמשת לשרטוט תמונות על הקנבס במיקום ובגודל מותאם.

הסבר על אובייקטים בJavaScript:

סינטקס:

var obj = {

    food: "Apple",

    car: "BMW",

    color: "Red"

}

אובייקט מורכב מזוגות של (מפתח: ערך), כל זוג כזה נקרא תכונה.

כדי לגשת לערך של תכונה יש להשתמש בפעולת נקודה ( . ) או בפעולת "אינדקס" ( [ ] ) על האובייקט.

דוגמא: obj.car, obj[“car”] – שניהם יחזירו BMW.

כדי להכניס תכונה שלא נמצאת באובייקט נשתמש באותה הפעולה בצירוף עם פעולת השמה.

דוגמא:

obj.drink ="Coca Cola" ,obj["drink"] = "Coca Cola" שניהם ייצרו תכונה בעל מפתח "drink" וערך "Coca Cola"

הסבר הפונקציות אשר לא נלמדו:

* הפונקציה:

JSON.parse()

מנתחת מחרוזת אשר בנויה בצורת אובייקט, יוצרת את האובייקט הזה ומחזירה אותו.

* הפונקציה:

JSON.stringify()

מקבלת אובייקט JavaScript ויוצרת ממנו מחרוזת אשר מייצגת את המידע ששמור באובייקט ומחזירה אותה. כלומר זוהי פונקציה הופכית ל JSON.parse()

* נסביר על מחרוזת מפורמטת (back ticks - ` `):

`rgb(${randVals[0]},${randVals[1]},${randVals[2]})`

אנו יכולים להבחין כי בדוגמא מעל ניתנת לנו האפשרות בשימוש של $ עם סוגריים מסולסלות {} באמצעות ה back ticks.

השימוש של ה-$ עם הסוגריים מאפשר לנו קבלת ערך אשר נמצא בתוך הסוגריים ולהשתילם במחרוזת

דבר זה שקול ל:

"rgb("+randVals[0]+","+randVals[1]+","+randVals[2]+")"

* הפונקציה:

document.querySelectorAll(`input[name=${radioName}]`);

הפונקציה מחפשת את כל האלמנטים שמתאימים לסינון שהועבר בארגומנט ומחזירה אותם במערך.

הארגומנט מועבר באותו הפורמט שCSS משתמש על מנת לסנן אלמנטים.

במקרה זה הפונקציה תחפש תגיות מסוג input בעלות תכונה name שהיא ערך המשתנה radioName.

* הפונקציה:

Element.addEventListener("click", function () {})

הפונקציה מוסיפה לאלמנט מאזין לאירוע מסוג "לחיצה", בעת הלחיצה מתבצעת פונקציה אשר מועברת בתור ארגומנט שני.

סוגי אירועים נוספים:

“change” – הפונקציה תיקרא בעת שינוי תכונה באלמנט

* האובייקט FileReader:

על מנת לקרוא קבצים בJavaScript, עלינו להשתמש באובייקט מסוג FileReader. עבור אובייקט זה השתמשנו במתודה ואירוע, והן:

* בנאי: (constructor):

let fr = new FileReader();

בנאי מאפשר לנו ליצור אובייקט מסוג FileReader שבאמצעותו ניתן לקרוא למגוון מתודות.

* האירוע: (event)

fr.onload = function () {}

ל-onload משימים(assigning) פונקציה שתיקרא בעת קריאה מקובץ.

let fileSelect = document.getElementById("image");

לאלמנט input מסוג file קיים תכונה files המכילה מערך של שמות הקבצים שנבחרו בלחיצה על select files.

בקובץ הHTML קיים אלמנט input מסוג file בעל id=image.

* המתודה: (method)

fr.readAsDataURL(fileSelect.files[0]);

המתודה מקבלת שם של קובץ וקוראת אותו. ובנוסף, היא מפעילה (trigger) בין היתר את האירוע onload.

* התכונה: (attribute)

fr.result

מחזירה את התוכן של הקובץ אשר נכנס אל המתודה readAsDataURL() בתור מחרוזת. במקרה שלנו יוחזר קידוד Base64 עבור התמונה.

* האובייקט Image:

על מנת לקרוא תמונות בJavaScript, עלינו להשתמש באובייקט מסוג Image. עבור אובייקט זה השתמשנו במתודה ואירוע, והן:

* בנאי: (constructor):

var img = new Image()

בנאי מאפשר לנו ליצור אובייקט מסוג Image שבאמצעותו ניתן לקרוא למגוון מתודות.

* האירוע: (event)

img.onload = function(){}

ל-onload משימים(assigning) פונקציה שתיקרא בעת קריאת תמונה.

img.src

תכונה המאפשרת גישה למקור התמונה

מחרוזות:

* המתודה: (method)

indexOf()

המתודה מחזירה את האינדקס הראשון שבו נמצא האלמנט הנתון במערך, אחרת תחזיר 1- אם הוא לא קיים.

* המתודה: (method)

slice()

המתודה מקבלת לכל היותר שני אינדקסים ומבצעת את הפעולות הבאות:

כאשר לא מועבר ארגומנט - המתודה מתפקדת כהעתקת זיכרון.

כאשר מועבר ארגומנט יחיד - המתודה תחזיר את המחרוזת החל מהאינדקס ועד סוף המחרוזת

כשאר מועברים 2 ארגומנטים - המתודה תחזיר את המחרוזת החל מהאינדקס הראשון ועד האינדקס השני (לא כולל).

מערכים:

* המתודה: (method)

push()

המתודה מוסיפה את האלמנטים שצוינו לסוף המערך ומחזירה את האורך החדש של המערך.

ספריית Math:

* המתודה: (method)

Math.floor

המתודה מעגלת מספר מסוים כלפי מטה, ומחזירה אותו.

* המתודה: (method)

Math.random()

המתודה מחזירה מספר עשרוני פסודו-אקראי בין 0 ל- 1.

* לולאת For: (For-Of Loop)

for (const element of s) {

        if (element.checked) {

            return element;

        }

    }

זוהי לולאה אשר נלמדה בקורס קודם "מבוא למדעי המחשב" (לולאת for-in בפייתון) אך בעל סינטקס שונה.

* המתודה: (method)

location.replace("index.html")

המתודה משנה את מיקום המשתמש באתר (redirect). במקרה שלנו אל index.html (דף הבית).

* הפונקציה:

confirm()

למדנו את הפונקציה alert, אשר מציגה הודעה. הפונקציה הזאת מביאה את האופציה ללחוץ על "אישור" או "ביטול", מחזירה true או false בהתאמה.

* הפונקציה:

appendChild()

הפונקציה משמשת ליצירת תוכן חדש בjavascript והוספה שלו לאלמנט שכבר קיים בHTML

* הפונקציה:

createElement()

הפונקציה משמשת ליצירת אלמנטים באופן דינמי בjavascript

* הפונקציה:

createTextNode()

הפונקציה משמשת להכנסת טקסט לתוך אלמנט בHTML