שיעור 3 Svcollege css

מארגין אוטו זה בסדר הוא ממרכז באמצע המסך את מה שנירצה מבחינת ימינה שמאל לא מלמעלה למטה שזה דופק הכל מבחינת מסכים

בשביל למעלה למטה נבין מה זה display flax רכרגע אנחנו משתמשים ב בלוק ולוא בפלקס כי אין לנו מידע על גודל הארון והוא לא נותן מידי פתרון לכן נישתמש בפלקס אתה עכשיו יודע מה גודל הארון הוא חילק את זה בשורה ואני לא רוצה את זה אני רוצה שזה יהיה בעמודה יש דיירקשן לפלקס יעני flax - directuon והוא אומר לי בוא נעשה את זה בעמודה מסודר לאחר מכן אנו רואים שיש בעיה של צפיפות בין האלמנטים לכן נוסיף gap שזה רווח בין האלמנטים

נסביר את כל הדברים החדשים שהוספו לבינתים

1. font-family

css

Copy

Edit

font-family: system-ui, -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, Oxygen, Ubuntu, Cantarell, 'Open Sans', 'Helvetica Neue', sans-serif;

🔹 מה זה עושה?

קובע את סוג הגופן (font-family) של הטקסט בתוך האלמנט.

רשימת הגופנים נכתבת לפי סדר עדיפות: הדפדפן ינסה להשתמש בגופן הראשון, ואם הוא לא קיים במערכת המשתמש, הוא ינסה את הבא ברשימה.

2. min-height

css

Copy

Edit

min-height: 100vh;

🔹 מה זה עושה?

מגדיר גובה מינימלי (min-height) של האלמנט כך שהוא יהיה לפחות 100% מגובה המסך (viewport height).

vh (Viewport Height) – יחידת מידה שמתייחסת לגובה החלון:

100vh = כל גובה המסך

50vh = חצי מגובה המסך

🔹 מתי זה שימושי?

כאשר רוצים שהאלמנט יתפוס את כל גובה המסך גם אם התוכן שבו קטן.

לדוגמה: כאשר יוצרים עמוד ראשי מלא מסך.

3. margin

css

Copy

Edit

margin: 0;

🔹 מה זה עושה?

מגדיר שהמרווח (margin) החיצוני של האלמנט יהיה 0 פיקסלים מכל הצדדים.

מבטל מרווחים ברירת מחדל של הדפדפן (למשל body מקבל margin אוטומטי).

🔹 מה זה עושה?

כאשר הוא משמש בתוך אלמנט עם display: block או display: flex, הוא מרכז את האלמנט אופקית.

שיטה טובה למרכז אלמנט בתוך הדף.

4. text-align

css

Copy

Edit

text-align: center;

🔹 מה זה עושה?

גורם לכל הטקסטים בתוך האלמנט להיות ממורכזים אופקית.

🔹 חשוב לדעת:

עובד רק על טקסט בתוך אלמנטים מסוג block, ולא משפיע ישירות על אלמנטים מסוג inline או inline-block.

5. display

css

Copy

Edit

display: flex;

🔹 מה זה עושה?

הופך את האלמנט ל"קונטיינר גמיש" (Flexbox), שמאפשר לשלוט על הפריסה והיישור של הילדים שלו.

האלמנטים שבתוכו הופכים לflex items, כלומר הם מתארגנים בהתאם לכללים של Flexbox.

🔹 למה להשתמש ב-Flexbox?

הוא מאפשר יישור דינמי של אלמנטים (אופקי / אנכי).

קל למרכז אלמנטים גם כשיש גודל דינמי.

6. font-weight

css

Copy

Edit

font-weight: bold;

🔹 מה זה עושה?

הופך את הטקסט לעבה יותר (מודגש).

ניתן להשתמש בערכים מספריים (100 עד 900) לקביעת דרגת העובי.

7. width

css

Copy

Edit

width: fit-content;

🔹 מה זה עושה?

האלמנט יתאים את רוחבו לפי התוכן שבו ולא יתפוס מקום מיותר.

במקום להגדיר רוחב קבוע, האלמנט פשוט יתאים את עצמו.

8. padding

css

Copy

Edit

padding: 60px;

🔹 מה זה עושה?

מוסיף 60 פיקסלים של ריווח פנימי בין גבולות האלמנט לתוכן שלו.

🔹 דוגמה ויזואלית: אם יש לך <div> עם טקסט בפנים, padding מרחיק את הטקסט מקצוות הdiv.

9. flex-direction

css

Copy

Edit

flex-direction: column;

🔹 מה זה עושה?

משנה את כיוון היישור של האלמנטים הפנימיים (flex items) לטור אנכי.

ברירת המחדל של flex היא שורה (row), אך column הופך את זה לרשימה אנכית.

🔹 דוגמה להבדלים:

css

Copy

Edit

display: flex;

flex-direction: row; /\* כל האלמנטים בשורה \*/

css

Copy

Edit

display: flex;

flex-direction: column; /\* כל האלמנטים בטור אנכי \*/

10. gap

css

Copy

Edit

gap: 20px;

🔹 מה זה עושה?

מוסיף מרווח של 20 פיקסלים בין כל אלמנט בתוך flexbox (רק עבור אלמנטים פנימיים).

הרבה יותר נקי מלהשתמש ב-margin לכל אלמנט בנפרד.

סיכום -

מאפיין מה הוא עושה?

font-family קובע את סוג הגופן לפי סדר עדיפות

min-height: 100vh האלמנט יהיה לפחות בגובה המסך

margin: 0 מבטל מרווחים חיצוניים

text-align: center ממרכז את הטקסטים בתוך האלמנט

display: flex הופך את האלמנט לקונטיינר גמיש

font-weight: bold הופך את הטקסט לעבה

width: fit-content הרוחב מותאם לתוכן באופן דינמי

padding: 60px מרווח פנימי של 60px מכל הצדדים

flex-direction: column מסדר את האלמנטים בטור אנכי

gap: 20px מוסיף רווח של 20px בין האלמנטים

המשך השיעור :

נימאס לכתוב כמות כזו של גרידים אף אר נכיר את ריפיט

### סיכום הנושא: \*\*שימוש ב-Grid עם repeat() והתאמה למובייל באמצעות Media Queries\*\*

#### \*\*שימוש ב-`repeat()` ב-CSS Grid\*\*

במקום לכתוב מספר רב של עמודות או שורות באופן ידני, ניתן להשתמש בפונקציה `repeat()` שמאפשרת להגדיר מספר חזרות של יחידות רשת בקלות.

##### \*\*מבנה הפקודה:\*\*

```css

grid-template-columns: repeat(3, 1fr); /\* שלוש עמודות בגודל שווה \*/

grid-template-rows: repeat(2, 100px); /\* שתי שורות בגובה 100px \*/

```

\*\*דוגמאות נפוצות:\*\*

```css

grid-template-columns: repeat(auto-fit, minmax(200px, 1fr));

```

> מאפשר התאמה אוטומטית של מספר העמודות בהתאם לגודל המסך.

```css

grid-template-columns: repeat(auto-fill, minmax(150px, 1fr));

```

> דומה ל־`auto-fit`, אך משאיר מקומות ריקים אם אין מספיק אלמנטים.

---

#### \*\*התאמה למסכים קטנים (Mobile-First)\*\*

כדי לוודא שהאתר נראה טוב בכל גודל מסך, יש להשתמש ב-`media queries` ולעצב \*\*מהגדול לקטן\*\*.

##### \*\*כללי עבודה עם `media queries`:\*\*

- \*\*הסדר הוא מהגדול לקטן\*\* – קודם נגדיר את הגריד למסכים הגדולים ואז נקטין עבור מובייל.

- \*\*השתמשו ביחידות יחסיות\*\* כמו `fr`, `minmax()`, `vw` ו-`vh` כדי להימנע מעיצוב קשיח.

##### \*\*דוגמה לעיצוב Grid רספונסיבי:\*\*

```css

.container {

display: grid;

grid-template-columns: repeat(4, 1fr); /\* גריד עם 4 עמודות \*/

gap: 10px;

}

@media (max-width: 1024px) {

.container {

grid-template-columns: repeat(2, 1fr); /\* במסכים בינוניים – שתי עמודות \*/

}

}

@media (max-width: 768px) {

.container {

grid-template-columns: repeat(1, 1fr); /\* במסכים קטנים – עמודה אחת \*/

}

}

```

---

#### \*\*תגיות קשורות ב-HTML & CSS:\*\*

- `display: grid` – הפעלת Grid על אלמנט.

- `grid-template-columns` – הגדרת מספר עמודות.

- `grid-template-rows` – הגדרת מספר שורות.

- `repeat(n, value)` – חזרה על יחידות רשת.

- `gap` – ריווח בין האלמנטים.

- `media queries` (`@media`) – שינוי עיצוב בהתאם לגודל המסך.

- `minmax(min, max)` – קביעת טווח גודל לעמודות.

- `auto-fit` ו-`auto-fill` – התאמה אוטומטית של מספר העמודות.

---

### \*\*סיכום עיקרי:\*\*

- השתמשו ב-`repeat()` כדי לכתוב פחות קוד ולקבל גריד מסודר.

- צרו עיצוב רספונסיבי עם `media queries` מהגדול לקטן.

- `auto-fit` ו-`auto-fill` מסייעים בהתאמת כמות העמודות בהתאם למקום הפנוי.

- תמיד בדקו את העיצוב במסכים שונים כדי לוודא שהתוצאה טובה.

---

אם יש לך שאלות נוספות או שתרצה דוגמאות נוספות, אני כאן! 🚀

נלמד איך לעשות את האתר בגריד בשביל פון מובייל

צריך גרשיים בתוך

לשים לב במדיה שצריך לשים לכל גודל של מסך וכמובן מגודל גדול לנמוך מבחינת סדר קודם הגדול ואחר מכן קטן