

CSS = Cascading style sheets

. מה זה CSS?

- ראשי תיבות של **Cascading Style Sheets** = גיליונות סגנון מדורגים.
- שפה המגדירה איך אלמנטים ב-HTML יראו:

- צבעים
- גופנים
- גדלים
- מיקומים
- מרווחים

How CSS works with HTML

- CSS עובד על ידי יצירת **חוקים (Rules)** שמחברים לאלמנטים מסוימים ב-HTML.

CSS Syntax

חוק CSS מורכב מ-**Selector** ו-**Declaration Block**:

- **Selector** – קובע על איזה אלמנט החוק חל (למשל <p> או .className).
- **Declaration Block** – מה ההגדרות (המאפיינים) שיוחלו על האלמנט.

איך מפרידים בין מאפיינים?

- כל מאפיין מסתיים ב-; (נקודה־פסיק).
- בין המאפיין לערך מפרידים ב-: (נקודה־תיים).

הנחיות נוספות

- סוגריים מסולסלים {} עוטפים את כל ההגדרות.
- נקודה־פסיק בסוף כל שורה חוץ מהאחרונה **מומלץ תמיד להשאיר** (מונע שגיאות).
- רווחים ושבירת שורות לא משנים לפעולה – הם רק לקריאות.

Ways to Add CSS

- **Inline CSS (בתוך תגית)**

```
<p style="color:red; font-size:20px;">שלום</p>
```

- **Internal CSS (באותו קובץ HTML)**

```
<head>
```

```
<style>
```

```
p {
```

```
  color: blue;
```

```
}
```

```
</style>
```

```
</head>
```

- **External CSS (בקובץ נפרד)**

- כתיבת ה-CSS בקובץ נפרד (למשל style.css) וקישור אליו עם <link> ב-<head>.

```
<head>
```

```
<link rel="stylesheet" href="style.css">
```

```
</head>
```

Selectors

סוגי Selectors עיקריים:

1. Simple selectors – בחירה פשוטה

- לפי שם תגית: p { ... }

- לפי מחלקה (className): .className { ... }

- לפי מזהה (id): #idName { ... }

2. Combinator selectors – לפי קשר בין אלמנטים

- למשל לבחור אלמנט בתוך אלמנט אחר (p div).

3. Pseudo-class selectors – מצב מיוחד

○ כמו hover (כשהעכבר מעל אלמנט) או first-child.

4. Pseudo-element selectors – חלק מאלמנט

○ למשל first-line:: (שורה ראשונה בפסקה).

5. Attribute selectors – לפי תכונה

○ לדוגמה: ["input[type="text"] יבחר רק אינפוטים מסוג טקסט.

מה זה Class ב-CSS?

- **Class** מאפשר לנו ליצור שם שנוכל להצמיד לאלמנטים שונים ב-HTML כדי להחיל עליהם עיצוב מסוים.
- היתרון – אפשר להחיל את אותו Class על כמה אלמנטים שונים, בלי קשר לסוג התגית שלהם (<h1>, <div>, <p> וכו').
- **השם תמיד מתחיל בנקודה** כשאנחנו כותבים את הסלקטור ב-CSS

מה זה ID ב-CSS?

- **ID** הוא מזהה ייחודי לאלמנט ב-HTML.
- המטרה שלו – להחיל עיצוב או לזהות אלמנט מסוים **באופן חד-פעמי** בעמוד.
- בשונה מ-Class, ID אמור להופיע **פעם אחת בלבד** בכל עמוד.
- ב-CSS – הסלקטור מתחיל בסולמית #.

ID	Class	מאפיין
# (סולמית)	. (נקודה)	התחלה ב-CSS
לא, רק על אלמנט אחד	כן, על כמה אלמנטים	שימוש חוזר
גבוהה יותר	נמוכה יותר	עדיפות ב-CSS

כתיבת הערה ב-CSS

- התחלה: /*
- סיום: */

Inheritance CSS

- הורשה היא מנגנון שבו אלמנטים ב-HTML מקבלים מאפייני עיצוב מהאלמנט ההורה שלהם

מה כן נורש ומה לא?

- **כן נורש:** צבע טקסט (color), פונט (font-family), גודל פונט (font-size), מרווח בין שורות (line-height) ועוד.
- **לא נורש:** צבע רקע (background-color), גבולות (border), מרווחים (margin, padding) – צריך להגדיר אותם ישירות

Siblings – אחים

- **מה זה:** בוחר אלמנטים שנמצאים באותה רמה (אותו הורה).
- **General sibling selector (~)** – בוחר את כל האחים שמגיעים אחרי האלמנט.

```
h1 ~ p {  
  color: blue;  
}
```

(+) Adjacent sibling selector

- בוחר רק את האלמנט **האח הראשון** שמגיע אחרי האלמנט

```
h1 + p {  
  color: green;  
}
```

Child Selector – בחירת ילד ישיר

- מה זה: בוחר אלמנטים שהם ילדים ישירים של אלמנט מסוים.
- איך כותבים: עם סימן < בין ההורה לילד.

```
div > p {  
  
  color: red;  
  
}
```

Descendant Selector – בחירת כל צאצא

- מה זה: בוחר את כל האלמנטים שנמצאים בתוך האלמנט ההורה, בכל רמה (ילדים, נכדים וכו').
- איך כותבים: עם רווח בין ההורה לצאצא.

```
div > p {  
  
  color: red;  
  
}
```

Pseudo-elements

- תתי-אלמנטים ב-CSS הם "אלמנטים וירטואליים" שמאפשרים לעצב חלק מסוים מתוך אלמנט ב-HTML, או להוסיף תוכן לפני/אחרי האלמנט — **בלי לשנות את ה-HTML עצמו**.
- הם לא קיימים בפועל ב-HTML, אלא נוצרים על-ידי ה-CSS בזמן שהדפדפן מציג את הדף.

הדוגמאות הכי נפוצות

p,h1,h2::first-line

p,h1,h2::first-letter

p,h1,h2::before

p,h1,h2::after

- רוב ה-pseudo-elements עובדים רק עם אלמנטים **בלוקיים** (כמו פסקאות, כותרות).
- המאפיין content הוא חובה כשמשתמשים ב-::before או ::after.
- הסינטקס המומלץ הוא עם **שתי נקודות (::)**, אבל דפדפנים תומכים גם בגרסה הישנה עם נקודה אחת (:).

ארבעת מצבי הקישורים ב-CSS

ב-CSS יש לנו ארבע פסאודו־מחלקות עיקריות לעיצוב <a> בהתאם למצבו:

ב-CSS יש כלל שנקרא **LVHA** (או "LoVe HAte") כדי שהעיצוב יעבוד נכון:

1. **a:link** – מצב של קישור רגיל שעדיין לא נלחץ
2. **a:visited** – מצב של קישור שכבר ביקרנו בו בעבר.
3. **a:hover** – מצב כשהעכבר עובר מעל הקישור.
4. **a:active** – מצב של קישור שנמצא בלחיצה (רגע הלחיצה עצמה).

אם אתה משלב גם `before::` או `after::` עם המצבים האלה, פשוט מוסיפים אותם אחרי הפסאודו־מחלקה, למשל:

```
a:hover::after {  
  content: " ➡";  
}
```

עיצוב תגיות form

האלמנטים הנפוצים:

form, label, input (כולל type=text |email |number |password |file |checkbox |radio |range |submit),

textarea, select, option, fieldset, legend, button.

חלוקה כללית של יחידות מידה

ב-CSS יש שתי קטגוריות עיקריות:

1. **יחידות מוחלטות (Absolute Units)** – תמיד אותו גודל, ללא קשר למסך/קונטקסט.
2. **יחידות יחסיות (Relative Units)** – הגודל תלוי בגודל אחר (למשל גודל הפונט של ההורה, רוחב המסך וכו').

Absolute Units – יחידות מוחלטות

יחידות שלא משתנות בהתאם למסך:

- **px** – פיקסלים.
- **cm** – סנטימטרים.
- **mm** – מילימטרים.
- **in** – אינצ'ים ($1\text{in} = 2.54\text{cm}$).
- **pt** – נקודות ($1\text{pt} = 1/72$ אינץ').
- **pc** – picas ($1\text{pc} = 12\text{pt}$).

בפועל, בוב כמעט תמיד משתמשים ב־`px` כי הוא הכי יציב בין דפדפנים.

Relative Units – יחידות יחסיות

יחסיות לפונט

- **em** – מבוסס על גודל הפונט של האלמנט ההורה.
אם ההורה מוגדר `font-size: 20px`, אז `1em = 20px`.
- **rem** – מבוסס על גודל הפונט של `html` (השורש).
אם `html { font-size: 16px }`, אז `1rem = 16px` בכל מקום.

יחסיות לגודל המסך/קונטיינר

- % – אחוז מתוך הממד של ההורה.
- **vw** – אחוזים מרוחב חלון התצוגה (Viewport Width).
- $1vw = 1\%$ מרוחב המסך.
- **vh** – אחוזים מגובה חלון התצוגה (Viewport Height).
- $1vh = 1\%$ מגובה המסך.
- **vmin / vmax** – אחוזים מהערך הקטן/הגדול מבין רוחב או גובה החלון.

יחסיות לשורה

- **ex** – מבוסס על גובה האות הקטנה "x" בפונט.
- **ch** – מבוסס על רוחב הספרה "0" בפונט.

שימושים מומלצים

- **טקסט** – rem או em כדי לאפשר קנה-מידה נגיש (משתמש יכול להגדיל/להקטין פונט).
- **מרווחים פנימיים/חיצוניים** – em (קשור לגודל הטקסט), לפעמים px.
- **פריסות רספונסיביות** – vw, vh, %.
- **אנימציות / אלמנטים גרפיים** – לרוב px או יחידות יחסיות למסך.

הכללים לכתיבת URL ב-CSS:

1. תמיד נשתמש במילת המפתח `url()`.
2. הערך בתוך הסוגריים יכול להיות:
 - **כתובת יחסית** (Relative URL) – למשל 'ballon1.jpg' שנמצאת באותה תיקייה כמו קובץ ה-CSS.
 - **כתובת מלאה** (Absolute URL) – למשל `https://example.com/images/ballon1.jpg`.
3. אם לנתיב יש **רווחים** או **תווים מיוחדים**, חובה לשים את הכתובת בתוך **גרשיים** או **כפולים**.
4. אפשר להשתמש בגרשיים גם בנתיבים רגילים, וזה אפילו נחשב לכתיבה מומלצת.

פורמטים נפוצים לכתיבת צבעים

- שם צבע (Named Colors)

CSS מכיר מעל 140 שמות צבעים באנגלית:

- Hexadecimal

```
color: #ff0000; /* אדום */  
color: #f00; /* קיצור של אדום */
```

- RGB (red,green,blue)

ערכים בין 0–255:

```
color: rgb(255, 0, 0); /* אדום */
```

- RGBA (עם שקיפות)

האחרון הוא 0 **Alpha** = שקוף לגמרי, 1 = אטום לגמרי).

עיצוב גופנים

- **font-family** – קובע את סוג הגופן

ניתן להגדיר גופן בודד או רשימת גופנים (הדפדפן יבחר את הראשון שקיים במחשב המשתמש)

- **font-size** – קובע את גודל הטקסט

אפשר להשתמש ביחידות pt, %, em, rem, px.

- **font-style** – קובע סגנון כתב:

normal – רגיל.

italic – כתב נטוי.

oblique – כתב נטוי (בדומה ל-italic, אך עם נטייה שונה)

- **font-weight** – קובע עובי כתב:

מילים: normal, bold, lighter, bolder.
מספרים: 100–900 (normal, 700 = bold = 400).

- **color** – צבע הטקסט.

תכונות נוספות לעיצוב טקסט

- **text-align** – יישור טקסט (left, right, center, justify).
- **text-decoration** – קו תחתון/קו מעל/קו חוצה (underline, overline, line-through, none).
- **letter-spacing** – ריווח בין אותיות.
- **word-spacing** – ריווח בין מילים.
- **line-height** – גובה שורה (מרווח אנכי בין שורות).
- **text-transform** – שינוי מצב אותיות (uppercase, lowercase, capitalize).
- **direction** – כיוון כתיבה

Border - עיצוב גבולות

border: [עובי] [סוג] [צבע];

גבולות בצדדים שונים

```
p {  
  border-top: 2px solid red;  
  border-right: 3px dashed blue;  
  border-bottom: 4px double green;  
  border-left: 5px dotted purple;  
}
```

border-radius – פינות מעוגלות

```
div {  
  border: 2px solid black;  
  border-radius: 10px;  
}
```

עיצוב רקע

1. background-color – קובע את צבע הרקע.
2. background-image – מציב תמונת רקע.
3. background-repeat – קובע האם ואיך התמונה תחזור (repeat, repeat-x, repeat-y, no-repeat).
4. background-position – קובע את מיקום התמונה בתוך האלמנט (למשל center top).
5. background-size – קובע את גודל התמונה (auto, cover, contain, ערכים מספריים).
6. background-attachment – קובע אם הרקע יזוז עם התוכן או יישאר קבוע (scroll, fixed, local).
7. background – תכונת קיצור המאפשרת להגדיר את כל הפרמטרים בשורה אחת.

תכונות CSS לעיצוב רשימות

- list-style-type – קובע את סוג הסימן
- list-style-image – שימוש בתמונה במקום סימן.
- list-style-position – מיקום הסימן ביחס לטקסט.
 - outside (ברירת מחדל) – הסימן מחוץ לטקסט.
 - inside – הסימן בתוך בלוק הטקסט.
- list-style – תכונת קיצור לשלוש התכונות הנ"ל.

עיצוב מתקדם של רשימות

- שימוש ב-::marker לעיצוב הסימן:

```
li::marker {  
  
  color: red;  
  
  font-size: 20px;  
  
}
```

display

block

- האלמנט תופס שורה שלמה (שובר שורה לפני ואחריו).
- אפשר לשנות לו גובה (height) ורוחב (width).
- דוגמאות בברירת מחדל: <h1>–<h6>, <p>, <div>.

inline

- האלמנט מופיע באותה שורה עם אלמנטים אחרים.
- לא אפשר לקבוע לו רוחב/גובה ישירות.
- דוגמאות בברירת מחדל: , <a>, .

inline-block

- כמו inline (נשאר בשורה), אבל כן אפשר לשנות לו רוחב/גובה.

• none

• Flex

• Grid

cursor

- default – סמן ברירת מחדל (חץ רגיל).
- pointer – סמן יד (בד"כ עבור קישורים).
- text – סמן כתיבה (קו אנכי).
- move – סמן הזזה (ארבעה חצים).
- crosshair – סמן בצורת צלב.
- wait – סמן טעינה (שעון חול / עיגול מסתובב).
- not-allowed – סמן איסור.
- help – סמן עזרה (סימן שאלה).
- url ("...") – סמן מותאם אישית מתמונה, עם אפשרות ל-fallback.

Box model - מודל הקופסא

- Content – התוכן עצמו (טקסט, תמונה וכו').
 - הרוחב והגובה (width / height) משפיעים על השטח הזה בלבד.
- Padding – הרווח בין התוכן לגבול (border).
 - שקוף, אבל מגדיל את שטח הרקע.
 - אפשר להגדיר לכל צד בנפרד (padding-top, padding-right וכו').
- Border – הגבול שסביב padding וה-content.
 - יש לו עובי (border-width), סגנון (border-style) וצבע (border-color).
- Margin – הרווח מחוץ לגבול עד לאלמנטים אחרים.
 - שקוף, לא צובע את הרקע.
 - משמש להפרדה בין אלמנטים.



טיפוסי תגיות

. תגיות Block-Level

- מאפיינים:

- תופסות את כל רוחב השורה.

- אפשר לקבוע להן רוחב (width) וגובה (height).

- דוגמאות:

.<div>, <p>, <h1>–<h6>, , , , <table>, <form>, <header>, <footer>

תגיות Inline

- מאפיינים:

- מופיעות בתוך השורה יחד עם אלמנטים אחרים.

- לא אפשר לשנות להן רוחב וגובה ישירות (אלא אם משנים ל-`display: inline-block`).

- דוגמאות:

., <a>, , , , <label>, <abbr>

תגיות Inline-Block

- מאפיינים:

- נשארות בשורה (כמו Inline).

- כן מאפשרות שינוי רוחב וגובה (כמו Block).

- דוגמאות:

- אין תגיות Inline-Block כברירת מחדל, אבל אפשר להפוך כל תגית ל-`inline-block` בעזרת CSS:

תגיות מיוחדות (Replaced Elements)

- אלו אלמנטים שהדפדפן "מחליף" בתוכן חיצוני (כמו תמונה או וידאו).

- דוגמאות:

., <video>, <iframe>, <canvas>

Division - div

- סוג: Block-level element
- תופס את כל רוחב השורה.
- מתחיל בשורה חדשה.
- שימוש נפוץ: יצירת אזורים (Sections) בדף, עטיפה של תוכן בשביל CSS או JavaScript.

Span –

- סוג: Inline element
- לא מתחיל שורה חדשה.
- משמש להדגשה או עיצוב חלק קטן בטקסט.
- שימוש נפוץ: עיצוב מילה או חלק מטקסט מבלי לשבור את השורה.

טיפ:

- אם אתה רוצה להפריד בלוקים – השתמש ב-`<div>`.
- אם אתה רוצה לעצב רק חלק קטן בטקסט – השתמש ב-``.

מה עושה

float

- המאפיין float מזיז אלמנט לצד שמאל או ימין בתוך הקונטיינר שלו, ומאפשר לאלמנטים אחרים (כמו טקסט) לזרום לצידו.
- ערכים אפשריים:
 - left – מצמיד את האלמנט לשמאל.
 - right – מצמיד את האלמנט לימין.
 - none – ברירת מחדל, אין הצמדה.
 - inline-start / inline-end – תומך בכיווני שפות.

בעיית ה-FLOAT קריסת גובה

כאשר כל האלמנטים בקונטיינר צפים (float), הקונטיינר "קורס" ולא לוקח את הגובה שלהם.

הפתרון: להשתמש ב-clear או ב-overflow: auto; על הקונטיינר.

ניקוי הצפה (Clearfix)

```
.container::after {  
  content: "";  
  display: table;  
  clear: both;  
}
```


סיכום CSS

מה זה CSS

- CSS = Cascading Style Sheets – קובץ הוראות לעיצוב דפי HTML.
- מפריד בין **תוכן** (HTML) לבין **עיצוב** (CSS).
- ניתן לשלב ב-3 דרכים:
 1. **Inline** – בתוך התגית (`"...="style`).
 2. **Internal** – ב-`<style>` בתוך `<head>`.
 3. **External** – בקובץ נפרד (css.) שמקושר עם `<link>`.

מבנה כלל ב-CSS

```
selector {  
  property: value;  
}
```

- **Selector** – מה לבחור (למשל `#id`, `.class`, `p`).
- **Property** – שם התכונה לעיצוב.
- **Value** – הערך של התכונה.

נושאים עיקריים שלמדנו

- **כתיבת הערות** – `/* טקסט */`.
- **הורשה** – חלק מהעיצובים עוברים מהורה לצאצאים.
- **סלקטורים** – לפי תגית, מחלקה, מזהה, קבוצתי, קונטקסטואלי.
- **תתי-אלמנטים** – `::first-letter::`, `after::`, `before::` ועוד.
- **עיצוב קישורים (a)** – מצבים: `:active`, `:hover`, `:visited`, `:link`.
- **עיצוב טפסים** – התאמה של `input`, `textarea`, `select`.

- **יחידות מידה** – מוחלטות (px, pt) ויחסיות (em, rem, vw, vh, %).
- **צבעים** – שמות, HEX, RGB/A, HSL/A.
- **רקעים** – צבע, תמונה, חזרה, מיקום, גודל, קיבוע.
- **גבולות** – צבע, עובי, סגנון.
- **רשימות** – שינוי bullets, מיקום, תמונות מותאמות.
- **display** – block, inline, inline-block, flex, grid, none.
- **cursor** – סוגי מצביעים (pointer, text, wait וכו').
- **Box Model** – content, padding, border, margin.
- **auto** – חישוב אוטומטי (למשל margin: auto למרכז).
- **סוגי תגיות** – block-level, inline, inline-block.
- **div מול span** – מבנה לעומת הדגשה מקומית.
- **float** – יישור לימין/שמאל עם ניקוי הצפה (clearfix).
- **pseudo-classes** – כמו p:hover (מצבים).
- **pseudo-elements** – כמו p::before (חלקים וירטואליים באלמנט).