

HTML

פרק מבנה דף האינטרנט

תוכן:

1. המסמכים בהקשר לדפי אינטרנט.
2. איך נעשה שימוש בדפים ליצור מסמך באופן מובנה.
3. בניית מסמכים מילולים.
4. HTML מתאר את המבנה של דפי אינטרנט.
5. HTML משתמש באלמנטים לתאר את מבנה של דפי האינטרנט.
6. הסבר על תגים ב - HTML.
7. התכונות (attributes) מספרים לנו יותר אודות האלמנטים.
8. הסבר על האלמנטים `<html>` `<head>` `<body>` `<<title>`
9. הסבר משמעות ה - HTML.
10. הסתכלות באיך אתרים אחרים להיות נבנים.
11. קוד במערכת ניהול תוכן CMS.
12. הסבר על core attribute שהן `id` ו `class` – `title` ו `style`
13. הבנת `block-level element` ו `inline element`
14. הסבר על אלמנטים `div` ו `span`
15. הסבר על אלמנט `<article>`.
16. הסבר על אלמנט `<section>`
17. הסבר על האלמנט `<nav>`

18. הסבר על אלמנט <header>

19. הסבר על אלמנט <aside>

20. הסבר על אלמנט <footer>

המסמכים בהקשר לדפי אינטרנט

אנחנו נתקלים באופן יומיומי בחיים בהרבה סוגים של מסמכים כמו עיתונים, טפסי ביטוח, קטלוגים של חנויות וכו'.

רבים מדפי אינטרנט מתנהגים כמו גירסאות אלקטרוניות של המסמכים הנ"ל. לדוגמא, עיתונים מציגים את אותם סיפורים או מאמרים במהדורה המודפסת כפי שהם נעשים באתר אינטרנט שלהם, ניתן ליישם זאת אף בתחום הביטוח על טפסי ביטוח שהם גם מוצגים על גבי אינטרנט כפי שהם מוצגים בגרסה המודפסת שלהם, וכן חנויות שיש להם קטלוגים אונליין כפי שמוצגים במהדורה מודפסת ואף מערכות סחר אלקטרוני שמדמים די את המציאות של סחר ומכר אך באופן אלקטרוני.

בכל סוגים של המסמכים שימוש במסמך בצורה מובנית הוא חשוב מאוד מאחר והוא מסייע לקורא להבין את המסר שמנסה מחבר המסמך להעביר ואף מסייע לניווט בגוף המסמך. לכן על מנת ללמוד איך לכתוב דפי אינטרנט זה מאוד חשוב להבין איך מבנה המסמך מובנה.

איך נעשה שימוש בדפים ליצור מסמך באופן מובנה

חשבו אודות הסיפורים והמאמרים שאתם קוראים בעיתון כאשר בכל סיפור או מאמר, ישנו כותרת וטקסט שהינו גוף הסיפור או גוף המאמר ויתכן לפעמים שיש איזשהו תמונות. אם הסיפור או המאמר הינו מתפרס על פני שטח נרחב של הדף אז יתכן שיהיו כותרות משניות שיפצלו את הסיפור או המאמר לחלקים נפרדים או שישומו ציטטות של אלו הכרוכים בתוך הסיפור או המאמר. כך שאנחנו רואים שהמבנה של המסמך כפי שהשתמשנו באופנים הללו בסיפור או במאמר עוזרים לקורא להבין את הסיפור או המאמר בעיתון ע"י חלוקת דף העיתון באופן המובנה בשימוש באופנים הנ"ל.

המבנה בתוך עיתון במהדורה מודפסת הינו דומה מאוד לאותו אופן כאשר סיפור חדשותי או מאמר נראים אונליין (אע"פ שזה עשוי להכיל בתוכו גם תכונות של וידאו או אודיו).

חשבו אודות סוג מאוד שונה של מסמך כטופס ביטוח. כאשר טופס ביטוח מורכב לרוב מכותרות עבור חלקים שונים של הטופס, כאשר כל חלק מורכב מרשימת שאלות עם שטחים המיועדים לעניית פרטים הנשאלים או תיבות סימון כדי לסמן בהם וי על מנת לענות לשאלות. מבנה טפסי הביטוח במהדורה המודפסת דומים מאוד למבנה טפסי הביטוח באונליין.

בניית מסמכים מילולים

השימוש של כותרות וכותרות משניות באיזשהו מסמך לרוב הוא משקף את המדרג של המידע. לדוגמה מסמך עשוי להתחיל עם כותרת גדולה ולאחר מכן מבוא או מרבית המידע החשוב שמכיל המסמך.

הטקסט במסמך עשוי להתרחב על פי הכותרות המשניות ולהשתרע כלפי מטה על פני דף המסמך. כאשר אנו משתמשים במעבד תמלילים ליצור מסמך אנו מפרידים את הטקסט הנכתב בגוף המסמך לחלקים נפרדים על מנת לארגן את מבנה המסמך. המבנה המסמך יכול כאשר בכל נושא בגוף המסמך עשוי להיות פיסקה חדשה וכאשר כל חלק במסמך יהיה כותרת שתתאר מה החלק זה בא לכסות.

באותו דרך של שימוש שעושים במבנה המסמכים מילולים במעבד תמלילים כ-וורד על מנת להקל את הבנת הנאמר בגוף המסמך אנחנו מבצעים גם בדפי האינטרנט כאשר אנו כותבים דפי אינטרנט.

HTML מתאר את המבנה של דפי אינטרנט

בחלון הדפדפן אנחנו יכולים לראות שדף אינטרנט שאנו יוצרים הוא בדיוק מאפיינים של אותו מבנה של תוכן כאשר אנו יוצרים זאת במסמך מילולי של וורד. בדפי האינטרנט על מנת לתאר את המבנה של הדף אינטרנט אנחנו מוסיפים קוד לתמליל שאנו רוצים שיופיע על הדף אינטרנט כאשר לקוד זה קוראים HTML. אתם יכולים להבחין שהקוד HTML שמוצג בדף הבא להיות בכחול והטקסט להיות בשחור.

```
<html>
<body>
  <h1>זה להיות הכותרת הראשית</h1>
  <p>זה טקסט שעשוי להיות מבוא לשאר הדף. ואם הדף להיות אחד ארוך </p>
  <p>הוא עשוי להיות מחולק ע"י כותרות משניות.
  <h2>זה להיות כותרת משנית</h2>
  <p>הרבה מאמרים ארוכים ישנם כותרות משניות על מנת לעזור לעקוב אחר
  המבנה של מה להיות נכתב. ישנו להיות אפילו כותרות משניות – משניות) או
  <p>כותרות ברמה נמוכה יותר).
  <h2>כותרת משנית אחרת</h2>
  <p>פה אתה יכול לראות כותרת משנית אחרת.</p>
</body>
</html>
```

הקוד HTML (הכחול) להיות מורכב מתווים ששוכנים בתוך הסוגריים המשולשים כאשר הללו להיות נקראים אלמנטים של HTML. האלמנטים לרוב מורכבים משתי תגים: התג הפותח והתג הסוגר. (התג הסוגר נוסף לוכסן בחלק קדמי של התג.) כל האלמנטים של

HTML מספרים לדפדפן כל דבר אודות המידע שיושב בין התג הפותח והתג הסוגר.

HTML משתמש באלמנטים לתאר את מבנה של דפי אינטרנט

התגים מתנהגים כמו מיכלים. שהם מספרים לנו כל דבר אודות המידע שהם מכילים. כאשר שוכן בין התג הפותח והתג הסוגר.

הסבר על תגים ב - HTML

התג שבו התג מצביע על המטרה של התג. לדוגמה התג `<p>` שהינו קיצור עבור **paragraph** כלומר פסקה. התג הסוגר יש לוכסן / אחר הסמל `<`.

לסירוגין משתמשים במונחים אלמנטים ותגים אף שמדברים עליהם כדברים נפרדים אך התגים נחשבים כאלמנטים כאשר אלמנטים כוללים בתוכם את התג הפותח והתג הסוגר והתוכן ששוכן בניהם.

התכונות (attributes) מספרים לנו יותר אודות האלמנטים

התכונות מספקים מידע נוסף אודות התכנים של האלמנט. הם מופיעים בתג הפותח של האלמנט ומורכב משני חלקים חלק השם וחלק של הערך, שמופרדים בניהם ע"י סימן השוויון =.

לדוגמא:

```
<p lang="he"> הפסקה להיות בעברית </p>
```

השם של התכונה מצביע מה הסוג של המידע הנוסף שאתם מספקים אודות התוכן של האלמנטים. זה צריך להיות נכתב באותיות קטנות.

הערך להיות המידע או הגדרה עבור התכונה. זה צריך להיות ממוקם בגרשיים". בתכונות שונות יש ערכים שונים.

פה בדוגמא המוצגת למעלה ישנו תכונה הנקרא `lang` שהוא להיות בשימוש להצביע על השפה שמשתמשים באלמנט זה. הערך של התכונה על דף זה מפורט שזה להיות בעברית.

HTML 5 מאפשר לכם להשתמש בשמות של התכונות אף באותיות גדולות וגם להשמיט

את הגרשיים בערכים אך זה לא מומלץ לנהוג כך.

רוב התכונות יכולים להיות רק בשימוש על אלמנטים מסוימים, למרות זאת מעט מן התכונות יכולים להופיע על איזשהו אלמנט.

הסבר על האלמנטים `<html>` `<title>` `<head>` `<body>` `</html>`

האלמנט `<html>` מגדיר שכל התוכן של האלמנטים שהוא מכיל בתוכו הינו קוד HTML כאשר בעצם זהו מסמך HTML.

האלמנט `<body>` מציין שכל דבר השוכן בתוך האלמנט הזה יוצג בדף האינטרנט ע"י הדפדפן בחלון העיקרי שבו מוצג דף האינטרנט עצמו ולא ע"י הצצה של קוד המקור בלבד.

האלמנט `<head>` הינו אלמנט שמופיע לפני האלמנט `<body>`. כאשר האלמנט זה מכיל מידע אודות הדף יותר מאשר המידע שלהיות מוצג בתוך הדף האינטרנט עצמו בחלון הראשי של הדף. וניתן לראות באמצעות קוד המקוד. לרוב נמצא בתוך האלמנט הזה את האלמנט `<title>` בתוך האלמנט `<head>`.

האלמנט `<title>` כאשר התוכן של האלמנט להיות נמצא בלשונית בראש הדפדפן שמעל הכתובת URL של הדף ההוא. התוכן שבו הינו בעצם הכותרת של דף האינטרנט.

הסבר משמעות ה – HTML

HTML זה ראשי תיבות של Hypertext markup language.

החלק של HyperText כלומר הלינק(קישור) כאשר מתייחס לעובדה ש HTML מאפשר לנו ליצור לינקים שמאפשרים למשתמש לעבור מדף אחד לדף אחר באופן מהיר ובקלות.

החלק של Markup Language כלומר שפת תיוגים המתייחס לעובדה ש HTML מאפשר לנו לפרש את הטקסט כאשר הפירושים הללו מספקים משמעות נוספת לתוכן של המסמך.

ניתן לחשוב על כך ש-Markup Language בדף האינטרנט שבעצם הינו מסמך טקסט שאנו מוסיפים קוד סביב הטקסט המקורי שאנחנו רוצים להציג והדפדפן אחר כך משתמש בקוד להציג את הדף האינטרנט באופן תקין. כך שהתגים שאנו מוסיפים להם זה בעצם פעולת התיג. ומחמת סיבה זו המתייחסת לתיג, HTML נחשבת שפה בצד לקוח מאחר שכל שימושה יהיה עבור הדפדפן ביחס לאופן מבנה הדף אינטרנט שיוצג בדפדפן שהינה תוכנה של לקוח.

קוד במערכת ניהול תוכן CMS

אם אתם עובדים עם מערכת לניהול תוכן, פלטפורמת בלוגים, או מערכת סחר אלקטרוני אתם קרוב לוודאי נכנסים לתוך חלק הניהול המיוחד של האתר (אדמיניסטרטור) של האתר כדי לשלוט בנעשה בו. הכלים המסופקים ע"י חלק הניהול המיוחד (אדמיניסטרטור) של האתר לרוב מאפשרים לכם לערוך חלקים של הדף יותר מאשר מכלול הדף עצמו שמשמעות העניין הוא שלעיתים נדירות אתם נתקלים באלמנטים `<html>` `<head>` או `<body>` שעוסקים במכלול הדף דבר המלמד על כך שהמאפיין במערכת הנ"ל שניהול אתרים הללו בדרך כלל הם התעסקות רק בעריכה של חלקי הדף ללא התעסקות של הקוד HTML עצמו.

לדוגמה כאשר מתארת איך במערכת ניהול תוכן (CMS) חלק ניהול באתר מספק כלים המתעסקים בחלקי הדף ולא במכלול הדף עצמו כאשר יש לנו תיבה המאפשרת להכניס כותרת עבור הדף, תיבה אחרת עבור עריכת מאמר ראשי ותיבה אחרת המאפשרת להכניס תאריך פרסום של המאמר וכל דבר שמצביע איזה חלק של האתר הדף משתייך.

ודוגמה לכך במערכת חנות סחר ומכר אלקטרוני כאשר מתארת איך חלק ניהול באתר מספק כלים המתעסקים בחלקי הדף ולא במכלול הדף עצמו שעשוי להיות תיבות שמאפשרות לנו להכניס כותרת עבור המוצר, תיאור של המוצר, מחיר שלו, וכמות הזמינה שלו.

זה להיות כך במערכות ניהול תוכן ומערכות חנות סחר ומכר אלקטרוני מכיוון שהם משתמשים בתבנית יחידה וקבועה לשלוט על כל דפים עבור המקטע של האתר כלומר התוכן (לדוגמה מערכת סחר ומכר אלקטרוני עשויה להשתמש באותו תבנית כדי להציג את כל המוצרים שלהם). המידע שאתם מספקים להיות ממוקם בתוך התבנית ולא על חלק המתייחס לתבנית שהינו מכלול הדף ולכן כל השליטה של הניהול הוא רק על התוכן של חלקי האתר אך לא על המבנה האתר בכללותו.

היתרון של הגישה זו להיות לגבי אנשים שאינם יודעים איך לכתוב דפי אינטרנט כאשר הגישה זו מאפשרת להם יכולת להוסיף מידע לאתר אינטרנט וגם מאפשר להם לעשות שינוי בתצוגה של כל דבר בתוך התבנית ועי"ז עתיד באופן אוטומטי לעדכן כל דף שמשמש בתבנית זו באתר. ניתן להמחיש את היתרון אם אתם מדמיינים שמערכת חנות סחר ומכר אלקטרוני עם 1,000 פריטים למכירה כאשר רק שינוי אחד בתבנית להיות הרבה יותר קל מאשר שינוי הדף עבור כל מוצר באופן פרטני. במערכות כמו זו כאשר יש לכם בלוק גדול של טקסט שאתם יכולים לערוך כגון מאמר חדשות, פוסט בלוג (Blog entry) או תיאור של מוצר במערכת חנות סחר ומכר אלקטרוני אתם עתידים לראות באופן תכוף עורך טקסט (text editor) מוצג.

עורך טקסט (text editor) לרוב יש שליטה על הכותרת כמו מעבד תמלילים שלכם, נתון לכם אפשרויות שונות לעצב טקסט להוסיף לינקים או להכניס תמונות. מאחורי התפאורות הללו עורכי הטקסט להיות מוסיפים קוד HTML לטקסט שלכם כמו שקוד HTML נעשה באופן רגיל. הרבה מעורכי הטקסט הללו ישנו להיות אפשרות שתאפשר לכם לראות ולערוך את הקוד שהם מייצרים.

ישנם עורכי טקסט לדוגמה שיש להם לשונית המראה HTML שזה מה שהמשתמש הכניס. ישנם מערכות אחרות שעשוי להיות כפתור כאשר מציע סוגריים משלושים שמצביע אל לגשת לקוד HTML.

פעם אחת שאתם יודעים איך לקרוא ולערוך את הקוד אתם יכולים לקחת שליטה נוספת מעבר לחלק המקטע של האתר אינטרנט שלכם.

ישנם מערכות ניהול תוכן (CMS) שמציעות כלים שגם מאפשרים לכם לערוך את קבצי התבנית. אם אתם מנסים לערוך קבצי תבנית אתם צריכים לבדוק את התייעוד עבור המערכת ניהול תוכן שלכם (CMS) כשהם נבדלים כולם מכל האחרים. אתם צריכים להיות זהירים כלפי קבצי תבניות עריכה בגלל שאם אתם מוחקים החלק השגוי של הקוד או מוסיפים כל דבר בתוך המקום השגוי אז האתר עשוי לעצור את העבודה לחלוטין

הסבר על core attribute שיהיה id – class – title – style

הסבר id attribute:

אתם יכולים להשתמש בתכונת id על מנת לזהות באופן ייחודי איזשהו אלמנט בתוך הדף. כאשר אתם עשויים לרצות לזהות באופן ייחודי אלמנט כך שאתם יכולים ע"י התכונת id להתחבר לחלק ספציפי במסמך. או לפרט את הסגנון ש-css מתייחס לאותו אלמנט לדוגמא כאשר ישנם שני אלמנטים שהם מאותו סוג אלמנט אז תכונת id מאפשרת לזהות באופן ייחודי לאיזה מהם מפורט ב-css. או ב-JavaScript שצריך ליישם לתוכן של רק אלמנט אחד בתוך המסמך, אז תכונת id מאפשר להגדיר לאיזה אלמנט בדיוק ליישם עליו כאשר לדוגמא ישנם כמה אלמנטים מאותו סוג אלמנט שע"י תכונת id ניתן לדעת איזה אלמנט מאותו קבוצה עליו ה-JavaScript מיישמת, ואף הדפדפן עושה שימוש בכך על מנת לאפשר ללינק שנמצא בחלק אחד של הדף לגלול לחלק אחר באותו דף באמצעות תכונת id.

תחביר של תכונת id הינו מורכב מאלמנט והתכונה id כאשר הערך מופרד בסימן שיוויון עם שם id כשאתם קובעים.

```
<p id="accounts">This paragraph explains the role of the accounts department.</p>
```

```
<p id="sales">This paragraph explains the role of the sales department.</p>
```

בשתי דוגמאות הללו מדובר בשני אלמנטים מאותו סוג אלמנט p כאשר יש להם id שונה.

לערכים של תכונת id צריכים ששמות הערך id יתחילו מאותיות (a-z או A-Z) או יתחילו בקו תחתון ויכולים אח"כ להיות בהמשך הערך מוכל במספר של איזשהו אותיות, או בספרות מטווח 0-9 או מקף או קו תחתון, או נקודתיים, או נקודה.

התכונת id מתייחסת לאותו אות הכתובה באות גדולה או באות קטנה כשני אותיות שונות.

חייב להיות שערכים של תכונת id בתוך המסמך שלא יהיו אותו ערכים בעלי אותו שם באלמנטים בתוך מסמך HTML אחד.

הערכים של תכונת id אינם יכולים להיות ריקים או עם מרווחים בתוכם.

הסבר class attribute

אתם יכולים להשתמש בתכונת class על מנת לפרט שאלמנט משתייך למחלקה של שמפרטת את המשמעות של תוכן האלמנט כאשר ישנם כמה אלמנטים מאותו סוג כאשר ע"י המחלקה ניתן להבדיל בין אותם אלמנטים שיהיו מסווגים כחברים באותה מחלקה לאחרים. לדוגמא אתם עשויים לפגוש במסמך שמכיל הרבה פסקאות, ומעט מן הפסקאות הללו עשויות להכיל סיכום של נקודות מפתח כאשר במקרה זה אתם יכולים להוסיף את תכונת class לאותם אלמנטים של פסקאות <p> שמכילות את הסיכומים כאשר הערכים של התכונה class יכנונו summary (סיכום) וכל זאת יעשה על מנת להבדיל אותם מן שאר הפסקאות האחרות במסמך.

```
<p class="summary">Summary goes here</p>
```

התכונה class משמשת בדרך לשימוש עם CSS או JavaScript כאשר מאפשר להתמקד לגבי איזו אלמנט מבוצע פעולת העיצוב CSS או איזו אלמנט מבוצע עליו היישום ע"י JavaScript.

תחביר של המחלקה הינו class="className". ואף יכול להכיל רשימה של ערכים של שמות מחלקה נוספים

לערכים של תכונת class צריכים ששמות הערך class יתחילו מאותיות (a-z או A-Z) ויכולים אח"כ להיות בהמשך הערך מוכל במספר של איזשהו אותיות, או בספרות מטווח 0-9 או מקף או קו תחתון, או נקודתיים, או נקודה.

התכונת class מתייחסת לאותו אות הכתובה באות גדולה או באות קטנה כשני אותיות שונות.

ההבדל בין class ל id – כאשר id יכול להופיע רק פעם אחת באותו ערך בדף משא"כ class יכול להופיע עם אותו ערך יותר מפעם אחת בדף.

הסבר title attribute

אתם יכולים להשתמש בתכונת title כאשר מספקת מידע נוסף אודות האלמנט כאשר מציג את המידע בצורת tooltip שהינו ממשק גרפי במבנה חלון קטן שמציג מידע על איזור מסוים במסך ומקבלים מידע לגביו.

התצוגה של tooltip שנוצר ע"י התכונה title מתבצע כאשר העכבר נוגע במסך על בתוכן של האלמנט שבו קיימת התכונה title.

תכונה זו הינה ניתנת ליישום על גבי כל האלמנטים היות והיא משתייכת לקטגוריה של global attribute

הבנת inline element ו- block-level element

ישנם שתי קטגוריות של אלמנטים המוצגים בדף האינטרנט כאשר הם מוכלים באלמנט body כשהם: א. block-level elements. ב. inline elements

block-level elements

אלמנטים שהינם **block-level elements** שוכנים בתוך התחום הכולל של אלמנט המוצא (**parent element**) ובשל כך יוצרים בלוק של תבנית ע"י שהאלמנטים הללו פועלים כיחידות של חלקים בתוך אלמנט המוצא.

הם מוצגים ע"י הדפדפן באופן ויזואלי כאשר הם מתחילים בשורה חדשה ושורה ריקה מפרידה לפני האלמנט הקודם ושורה ריקה המפרידה בינה לאלמנט הבא, ואף יכולים בעצמם להכיל את האלמנטים **inline** ואף להכיל טקסט, מידע, ונתונים או **block-level element** אחרים.

inline elements

אלמנטים שהינם **inline elements** שוכנים רק בתחום ספציפי של אלמנטים **inline** שבהם הם מוגדרים בתוך לדוגמא כאשר הם מוגדרים בתוך אלמנט **p** המציין פיסקה הם ישכנו רק בתוך פיסקה רק בחלק של אלמנט **inline** שבו הוגדרו בתוך הפיסקה, לכן אינם יוצרים שורה חדשה כאשר מגדירים אותם ופועלים בלבד במקום שבהם הוגדרו.

האלמנטים שהינם **inline elements** יכולים להכיל נתונים, טקסט, ו- **inline elements** אחרים. ואף קטן יותר לרוב מ- **block-level elements**

הסבר על אלמנטים **div** ו- **span**

האלמנט **div** הינו האלמנט המגדיר את החלוקה בתוך מסמך **HTML** דהיינו את זרימת התוכן של המסמך, או חלק במסמך. ומשמש לקבץ את הבלוקים של האלמנטים על מנת לעצב אותם עם **CSS** בשימוש התכונות **id**-ו-**class**, ומחמת היותו פועל על בלוקים כאשר מוכלים בתוכו נחשב בקטגוריית אלמנטים ברמת בלוק (**block-level element**)

דוגמאות לשימושים באלמנט **div**

לאלמנט **div** אין כלל משמעות סמנטית ומשתמש עם התכונות **id** -ו- **class** -ו- **lang** -ו- **title** להעניק משמעות סמנטית לאלמנטים רציפים בקטגוריית **block level element**.

ניתן להשתמש באלמנט **div** לקבץ אלמנטים לביצוע של **wrapping** (עטיפה) של עיצוב האתר בכללותו כלומר האלמנט **div** משמש פה לקבץ את כל האלמנטים שבאתר רק לסיוע לעיצוב ואין כלל משמעות סמנטית כלל מחוץ לדבר שצוין.

ניתן להשתמש באלמנט **div** כאשר רוצים לעצב פסקה במאמר כדוגמא שפסקה זו מהווה מבוא במטרה להציג אותה בעיצוב ייחודי למבוא.

ניתן להשתמש באלמנט **div** כאשר רוצים להבדיל בין התוכן הראשי באתר לחלק של **nav** וחלק של **HTML header** וחלק של **footer**, אך יש להעיר שזה היה מצוי שבגירסאות קודמות של **HTML** לפני שהחל עידן השימוש ב **HTML 5**.

האלמנט **span**

האלמנט **span** הינו אלמנט המספק לקבץ אלמנטים בקטגוריית **inline elements** על מנת לעצב אותם עם **CSS** בשימוש התכונות **class** -ו- **id**, ואף פעולות של **scripting** יכולות להיות מוגדרות תחת האלמנט **span** אך השוני בינו ל- **div** רק כש- **span** פועל על **inline**

elements ולא על **block level elements** כאשר לדוגמה אני רוצה שרק חלק בתוך הטקסט יהיה מעוצב באופן שונה אני משתמש באלמנט **span**.

אלמנט **span** משמש להבדיל בין טקסט לטקסט הסובב אתו טקסט האופן כאשר אין אלמנט אחר שיכול להכיל זאת.

פעולות העיצוב עם האלמנט **span** בהקשר **CSS** נעשה עם התכונה **style** מאחר ומדובר ביישום של **CSS** באופן שמבוצע **inline**.

הסבר על אלמנט `<article>`

האלמנט **article** מציין תוכן עצמאי במסמך, בדף, באפליקציה או האתר שנועד להיות באופן עצמאי מחולק כחלק עצמאי.

דוגמאות לכך פוסט של פורום (**forum post**), ופוסט של בלוג (**blog entry**), מאמר חדשות, ותגובות או כל תוכן עצמאי אחר.

האלמנט **article** בדרך כלל כולל בתוכו את האלמנטים של **heading**, ואלמנט **p**, ויתכן אלמנט **section** ועוד אלמנטים אחרים.

אלמנט **article** יכול להופיע מספר פעמים במסמך כלדוגמה האלמנט **article** יופיע בתוך אלמנט **section** במסמך המציין חלק אחר במסמך שבו מאמר אחר.

הסבר על אלמנט `<section>`

האלמנט **section** מגדיר חלקים במסמך כמו פרקים או כותרות או הסיומות במסמך וכן איזשהו חלקים אחרים במסמך כאשר בדף האינטרנט או מאמר ישנו הרבים חלקים. כלדוגמה דף הבית יכול להיות חלק עבור הצגת אודות החברה חלק אחר עבור הצגת פריטים חדשים בחברה וחלק אחר עבור מידע התקשרות.

בשימוש האלמנט **section** צריך קודם בתחילה אחד מהאלמנטים הללו **article** או **aside** או **nav** או **header**.

האלמנט זה מכיל את האלמנטים של **heading** שהן **h1-h6** בתחילתו של התוכן האלמנט.

האלמנט זה משתייך לקטגוריה של **block level element**.

באלמנט **section** הינו אלמנט סמנטי אך לא נעשה בו שימוש להגדיר למטרות עיצוב של החלקים או של סקריפטים אלא למשמעות הסמנטית בלבד בהגדרת חלקים בו וזהו ההבדל בינו לאלמנט **div** כאשר האלמנט **div** אין שום משמעות סמנטית כשלעצמו.

הסבר על האלמנט `<nav>`

האלמנט **nav** מציג את החלק הניווט של הדף אינטרנט, ומכיל רשימה של לינקים של ניווט לדפים אחרים או חלקים אחרים בתוך באתר או האפליקציה ואף לניווט חלקים באותו דף כגון שהאלמנט **nav** משמש כטבלת תוכן עניינים עבור מאמר ארוך כאשר מכיל לינקים לנווט לחלקים בגוף מאמר הארוך.

האלמנט `nav` הינו קיצור `navigation` כלומר ניווט.

הלינקים שהוא מכיל האלמנט `nav` רק הם קבוצה של לינקים שהם משמשים לניווט ראשי .

האלמנט `nav` אף יכול להכיל תיבת חיפוש, כאשר תיבת חיפוש הינה חלק מניווט באתר.

האלמנט `nav` הינו ממוקם כחלק של האלמנט `header` או ממוקם כחלק של האלמנט `footer`.

האלמנט `nav` משתייך לקטגוריית `block level element`

הסבר על אלמנט `<header>`

האלמנט `header` מגדיר בתוכו תוכן מקדים של מאמר שהינו המבוא או של דף אינטרנט או קבוצה של לינקים לניווט באתר.

האלמנט `header` מכיל בדרך כלל את האלמנטים של `heading` מ-`h1`–`h6`. לוגו או אייקון של האתר. מידע על מחבר מאמר כגון כתבת חדשות באתר פרטים מיהו המחבר של הכתבה מצויים באלמנט זה.

האלמנט `header` אינו יכול להיות ממוקם בתוך אלמנט `header` אחר, ו- בתוך האלמנט `footer` ו- בתוך האלמנט `address`.

האלמנט `header` ממוקם בתחילת המסמך בתור חלק המוגדר ככותרת הדף או בתוך האלמנט `article` כאשר מגודר ככותרת המאמר או בתוך האלמנט `section` ככותרת החלק או בתוך האלמנט `aside` ככותרת החלק הצדדי בדף.

האלמנט `header` הינו אלמנט סמנטי כאשר כל מטרתו הגדרת חלק תחילת המסמך או בתחילת מאמר.

יכולים היות מספר אלמנטים של `header` באותו מסמך כאשר לדוגמא `header` שבתוך אלמנטים `section` שמופיעים במסמך כאשר האלמנט `header` מגדיר את תחילת אותו חלק שהוא שוכן בתוכו כך ש `header` יכול להופיע אף כראש החלק התחתון של המסמך.

האלמנט `header` משתייך לקטגוריה של `block level element`

הסבר על אלמנט `<aside>`

האלמנט `aside` מגדיר את התוכן שממוקם בחלק הצדדי של הדף כאשר אינו יכול להופיע ביחד אלא בצד בנפרד בכך שהוא מציג את התוכן הזה שהינו קשור לתוכן הראשי של הדף כגון ציטוט מלא שמוצג בחלק צדדי של מאמר, זה כאשר מופיע כמקונן בתוך האלמנט `article` ותוכן הקשור בעקיפין לתוכן של האתר כגון לינקים נוספים לקריאה בעניין המפורסם באתר וביוגרפיה של המחבר ואף מילון מונחים המסבירים המונחים שנעשה בהם שימוש בתוכן האתר ומודעות וזהו בין שמופיע כמקונן באלמנט `article` או שלא מופיע בכלל באלמנט `article`.

האלמנט **aside** בעצם מציג את התוכן המוקף שהינו תוכן עצמאי בהיותו ממוקם בנפרד מן התוכן הראשי.

האלמנט **aside** הינו אלמנט סמנטי כאשר כל מטרתו להציג הגדרת חלק הצדדי במסמך.

הסבר על אלמנט <footer>

האלמנט **footer** מגדיר את הסיום במסמך או סיום של חלק.

האלמנט **footer** צריך להכיל מידע אודות האלמנטים השוכנים בו כגון מפת האתר כאשר באלמנט **footer** מסופק מידע אודות האלמנטים של הלינקים ששוכנים בתוך ה-**footer** שמדובר אלמנטים שמספקים קישורים לכל חלקי האתר שזהו בעצם מפת האתר.

האלמנט **footer** מכיל בדרך כלל מידע אודות המחבר, מידע על זכויות יוצרים, מידע על התקשרות, מפת האתר, לינקים חזרה לראש העמוד, ומסמכים שרלוונטים לתוכן המוצג באתר.

המידע על התקשרות כאשר מוצג בתוך האלמנט **footer** שוכן בתוך האלמנט **address** כאשר אלמנט זה שוכן האלמנט **footer**.

האלמנט **footer** יכול לשכון בתוך האלמנט **section** כאשר הוא מציג את סיום החלק. יכול לשכון גם בתוך האלמנט **article** כאשר הוא מציג את סיום המאמר.

האלמנט **footer** יכול להופיע יותר מפעם במסמך כלדוגמא כאשר האלמנט **footer** יופיע בתוך בשני אלמנטים של **section** שמופיעים במסמך בתור אלמנט המגדיר את סיום החלקים.

שאלות על הפרק

1. תאר הקשר מסמכים לדף אינטרנט ?
2. מדוע חשוב לעשות שימוש מסמך מילולי בצורה מובנית ?
3. תאר כיצד הדף עושה שימוש בארגון מסמך ?
4. מה משקף תהליך בניית המסמך ?
5. תאר בקצרה תהליך של בניית המסמך ?
6. מה משמש לתיאור מבנה האינטרנט ?
7. בשביל מה משמש קוד **HTML** ?
8. מסמך **HTML** איזו סוג מסמך הוא ?
9. הגדר מהו אלמנטים ותחביר שלהם ומהו התפקיד שלהם ?
- 10 הגדר מהו התגים והאם נחשב התייחסות שתגים הם חלק בלתי נפרד מהאלמנטים ?
11. מה תפקידו של **attribute** (תכונות) והתחביר שלו ?
12. תאר תפקידים של שני החלקים ב **attribute** שם וערך, וכלל התחביר שלהם, וציין את ההבדלים בין **HTML 5** לשאר הגרסאות האחרות של **HTML** בעניין זה ?
13. ציין כללים לגבי **attribute** ?
14. מה תפקידו של האלמנט **HTML** ?
15. מה תפקידו של האלמנט **head** ?
16. מה תפקידו של האלמנט **body** ?

17. מה תפקידו של האלמנט `title` ?
18. מהו הקיצור של `HTML` ?
19. למה מתייחס החלק של `Hypertext` ?
20. מדוע `HTML` נחשבת שפה בצד הלקוח ?
21. למה מתייחס החלק של `markup language` ?
22. תאר מה מאפיין את חלק הניהול במערכות ניהול תוכן ומערכות סחר אלקטרוני ?
23. הסבר את סיבת המאפיין את חלק הניהול במערכות ניהול תוכן ומערכות סחר אלקטרוני?
24. מה היתרון בגישה המאפיינת ה"ל בשאלה מס' 21 ?
25. מה מאפיין בעורכי טקסט של מערכות לניהול תוכן וכיצד מודגש מאפיין זה ?
26. מה היתרון בידיעת קוד ה-`HTML` ?
27. מה חשוב לשים לב במערכות ניהול תוכן המאפשרות שמאפשרות עריכת קבצי תבנית ?
28. מהם מאפיינים את תכונת `id` ?
29. מהם כללי התחביר של תכונת `id` ?
30. מה תפקידו של תכונת `id` ?
31. מהם מאפייני את תכונת `class` ?
32. מהם כללי התחביר של תכונת `class` ?
33. מה תפקידו של תכונת `class` ?
34. הסבר את המונח `block level elements` ?
35. הסבר את המונח `inline elements` ?
36. א. הסבר מהם מאפייני האלמנט `div` ?
ב. מהו תפקידו של האלמנט `div` ?
37. א. הסבר מהם מאפייני האלמנט `span` ?
ב. מהו תפקידו של האלמנט `span` ?
38. מה ההבדל בין האלמנטים `div` ו-`span` ?
39. א. הסבר מהו תפקידו של האלמנט `article` ?
ב. הסבר את מיקומו של האלמנט `article` בהיררכיה של המסמך ?
40. איזה אלמנטים לרוב מכיל האלמנט `article` ?
41. א. הסבר מהו תפקידו של האלמנט `section` ?
ב. הסבר את מיקומו של האלמנט `section` בהיררכיה של המסמך ?
42. ציין מהו ההבדל בין האלמנטים `div` ו-`section` ?
43. איזה אלמנטים לרוב מכיל האלמנט `section` ?
- 44.

פרק הטקסט בדף האינטרנט

תוכן:

1. מבוא.
2. הסבר על אלמנט heading `<h1>` - `<h6>`
3. הסבר על אלמנט paragraphs `<p>`
4. הסבר על אלמנטים bold and italic `` `<i>`
5. הסבר על אלמנטים superscript `<sup>` `<sub>` and subscript

6. חלקי דף ריקים בהקשר לדף אינטרנט (white space collapsing).
7. אלמנטים סמנטיים (semantic markup)
8. הסבר על אלמנטים strong and emphasis

9. הסבר על אלמנטים של quotations <blockquote>
<q> ותכונה cite
10. הסבר על אלמנט <abbr> שזהו abbreviations
and acronyms
11. הסבר על אלמנטים citations and definitions
<cite><dfn>
12. הסבר על האלמנט של פרטי המחבר המסמך
<address>
13. הסבר על אלמנט <small>
14. הסבר על אלמנט <mark>
15. הסבר על האלמנט <bdo> ותכונה dir
16. הסבר על אלמנט <bdi>

מבוא

כאשר יוצרים דף אינטרנט אנחנו מוסיפים תגים (ידועים כ **markup**) לתוכן של הדף. התגים הללו מספקים משמעות נוספת ומאפשרים לדפדפן להציג למשתמש את המבנה המתאים לדף האינטרנט.

תגים מובנים: האלמנטים שאנחנו יכולים להשתמש לתאר את מבנה הדף שכולל כגון תיאור כותרות וגם פסקאות.

תגים סמנטיים: אלמנטים שמספקים מידע נוסף כגון הדגשה על התוכן (**emphasis**) להיות ממוקמת בתוך המשפט, זה כל דבר שיש לכם לכתוב שהינו ציטוט (ומי אמר את זה), המשמעות של ראשי התיבות וכן הלאה.

הסבר על אלמנט heading <h1> - <h6>

ב- HTML יש שש רמות של כותרות: <h1> <h2> <h3> <h4> <h5> <h6> זה להיות בשימוש עבור הכותרות הראשיות.

<h2> זה להיות בשימוש עבור כותרות המשניות.

אם ישנו להיות חלקים נוספים תחת הכותרות המשניות <h3> להיות בשימוש וכן הלאה.

הדפדפנים מציגים את התכנים של הכותרות בגדלים שונים. התוכן של אלמנט <h1> יהיה הגדול ביותר, והתוכן של אלמנט <h6> יהיה הקטן ביותר. הגודל המדויק שבו כל דפדפן מציג את הכותרות יכול להשתנות במקצת. משתמשים יכולים גם להתאים את גודל של הטקסט בדפדפן שלהם. אתם עתידים לראות איך שליטת הגודל של הטקסט, הצבע שלו, והפונט שמשתמשים כאשר אנו באים להסתכל ב CSS.

האלמנטים h1- h6 משתייך לקטגוריית אלמנטים block-level elements.

הסבר על אלמנט <p> paragraphs

כדי ליצור פיסקה מקיפים את המילים שיוצרים את הפסקה עם התג הפותח <p> והתג הסוגר </p>.

ברירת המחדל הדפדפן עתיד להציג כל פיסקה בשורה חדשה עם איזשהו מרווח בין זה לפיסקאות העוקבות.

האלמנט p משתייך לאלמנטים בקטגוריית block-level element

הסבר על אלמנטים <i> bold and italic

בסגירת המילים בתגים אנחנו יכולים ליצור תווים שיופיעו בכתב מודגש.

האלמנט גם מציג חלק של טקסט שיהיה מוצג בדרך וויזואלית שונה (לדוגמא מילת מפתח בפיסקה) למרות השימוש של אלמנט לא מרמז איזשהו משמעות נוספת

בסגירת המילים בתוך התגים <i> </i> אנחנו יכולים ליצור תווים שיופיעו בכתב נטוי.

האלמנט <i> גם מציג חלק של הטקסט שיאמר בדרך שונה מהתוכן המוקף בפיסקאות כגון מונחים טכניים, שם של אוניות, מילים זרות, מחשבות ורעיונות, או מונחים אחרים שלרוב להיות מוצגים בכתב נטוי.

הסבר על אלמנטים <sup> <sub> superscript and subscript

האלמנט <sup> להיות משתמשים בו להכיל תווים שצריכים להיות מוצגים בכתב עילי כגון סיומת של תאריך או הפניות להערות שוליים או מושגים מתמטיים כמו לעלות מספר לחזקה כגון 2².

האלמנט <sub> להיות משתמשים בו להכיל תווים שצריכים להיות מוצגים בכתב תחת. זה להיות בדרך כלל בשימוש במתמטיקה או עם נוסחאות כימיות כגון H₂O

חלקי דף ריקים בהקשר לדף אינטרנט (white space collapsing)

על מנת ליצור קוד שיהיה קל יותר לקרוא, יוצרי דף האינטרנט לעיתים קרובות מוסיפים מרווחים נוספים או מתחיל באיזשהו אלמנטים בשורות חדשות.

כאשר הדפדפן נתקל בשני רווחים או מספר רווחים יותר שהם אחד ליד השני, הוא מציג רק מרווח אחד. באופן דומה אם הדפדפן נתקל במעבר לשורה או מעברי שורה נוספים אחד ליד השני הדפדפן מתייחס לזה כאותה שורה. זה ידוע כקריסת חלקי הדף הריקים (**withe space collapsing**).

אנחנו נראה לעיתים קרובות שיוצרי דף אינטרנט לוקחים בחשבון את היתרון של קריסת חלקי דפים ריקים (**white space collapsing**) וכך הם מבצעים הזחה בקוד שלהם על מנת ליצור קלות בקריאת הקוד וכך מאפשר לעקוב אחר הקוד הכתוב כדי להבינו.

הסבר על האלמנטים `
` `<hr>` `line breaks and horizontal rules`

כמו שראיתם כבר שהדפדפן יציג באופן אוטומטי כל פיסקה חדשה או כותרת בשורה חדשה. אבל אם אתם רוצים להוסיף מעבר לשורה נוספת באמצע הפיסקה אתם יכולים להשתמש עם התג של אלמנט `
` שהוא אלמנט המורכב רק מתג אחד שנכתב אף כתג סוגר אך שהלוכסן נמצא בסופו של התג בשונה מתג סוגר שישנם שני תגים כאשר הלוכסן נמצא בתחילתו של התו הסוגר.

באלמנט `br` זה שימושי מאוד עבור כתובות ופואמות.

כדי ליצור הפסק בין נושאים כגון שינוי נושא בספר או סצינה חדשה במחזה אתם יכולים להוסיף קו הפרדה אופקי שיוצר הפרדה בין החלקים משתמשים בתג של האלמנט `<hr />` שהינו קיצור של שם האלמנט `horizontal rules`. האלמנט זה מורכב רק מתג אחד שנכתב אף כתג סוגר אך שהלוכסן נמצא בסופו של התג בשונה מתג סוגר שישנם שני תגים כאשר הלוכסן נמצא בתחילתו של התו הסוגר.

ישנם להיות אלמנטים מעטים שאין איזשהו מילים ששוכנות בין התג הפותח לתג הסוגר. אלמנטים הללו להיות ידועים בכינויים אלמנטים ריקים והם להיות נכתבים באופן שונה משאר האלמנטים.

אלמנטים ריקים לרוב יש רק תג אחד. לפני הסוגר המשולש הסוגר של האלמנט לרוב יהיה מרווח ולוכסן צמוד לסוגר. ישנם איזשהו יוצרי דף אינטרנט שמשמטים את הלוכסן וזה אינו הרגל טוב.

אלמנטים סמנטיים (`semantic markup`)

ישנו להיות איזשהו אלמנטים הקשורים לטקסט שלא מיועדים להשפיע על מבנה הדפי אינטרנט שלכם, אבל הם מוסיפים מידע נוסף לדפים הם להיות ידועים כ-אלמנטים סמנטיים.

בשאר הפרק אנחנו נפגוש אלמנטים נוספים שיעזרו לכם כאשר אתם מוסיפים טקסט לדפי אינטרנט. לדוגמא אתם הולכים לפגוש את האלמנט `` שמאפשר לכם להצביע איפה הדגשה צריכה להיות ממוקמת במילים הנבחרות והאלמנט `<blockquote>` כאשר מצביע שבלוק של הטקסט להיות ציטוט.

הדפדפנים מציגים את התכנים של אלמנטים הללו בדרכים שונות. לדוגמה התוכן של אלמנט `` מוצג בכתב נטוי, והאלמנט `<blockquote>` התוכן של האלמנט לרוב להיות מוצג בהזחה. אבל אתם צריכים לא להשתמש בהם לשנות את הדרך שהטקסט שלכם יראה; המטרה שלהם לתאר את התוכן של הדף אינטרנט שלכם מדויק יותר.

הסיבה עבור שימוש באלמנטים הללו להיות שתוכנות אחרות כגון קוראי מסך או מנועי חיפוש יכולים להשתמש במידע הנוסף שהם מספקים. לדוגמה הקול של קוראי מסך עשוי להוסיף את הדגשה במהלך הקריאה למילים שבתוך האלמנט `` או מנועי חיפוש שעשויים לרשום כמאפיין את הדף אינטרנט שלכם ציטוט כחלק ממפתוח הדף אינטרנט שלכם אם נעשה שימוש בדף שלכם באלמנט `<blockquote>`.

הסבר על אלמנטים `` `` `strong and emphasis`

השימוש של אלמנט `` מצביע שהתוכן שלו יש חשיבות חזקה ע"י שמוצגים בהדגשה. לדוגמה, המילים שאלמנט מכיל עשויים להיאמר עם הדגשה חזקה.

הדפדפנים כשמוגדרים במצב של ברירת מחדל הם יציגו את התוכן של אלמנט `` בהדגשה.

אף שגם אלמנט `b` מוצג ע"י הדפדפן לעיני המשתמש באותו אופן ויזואלי כתוכן השוכן בתוך אלמנט `strong` אך המשמעות מתפרשת לחלוטין באופן שונה ע"י תוכנות שקוראות זאת כגון מנועי חיפוש שמתייחסים לכך שישנו חשיבות רבה לאלמנט `strong` כלומר שישנו רמת חשיבות חזקה לתוכן השוכן בתוך אלמנט ומוסיפים זאת למאפייני הדף לצורך מפתוח הדפים, זה משא"כ התוכן השוכן בתוך באלמנט `b` נחשב חסר חשיבות כלדוגמה משמש יותר מילת מפתח במאמר.

האלמנט `em` מצביע על הדגשה שמשנה בעדינות את המשמעות של המשפט.

הדפדפנים כשמוגדרים במצב של ברירת מחדל הם יציגו את התוכן של אלמנט `` בכתב בסגנון נטוי.

אף שגם אלמנט `<i>` מוצג ע"י הדפדפן לעיני המשתמש באותו אופן ויזואלי כתוכן השוכן בתוך אלמנט `em` אך המשמעות מתפרשת לחלוטין באופן שונה ע"י תוכנות שקוראות זאת כגון תוכנות קוראי מסך כאשר קוראים בהדגשה מוטעמת את התוכן השוכן בתוך האלמנט `em`, זה משא"כ התוכן השוכן בתוך אלמנט `i` שהינו תוכן השונה מטקסט הרגיל כגון מונחים טכניים.

הסבר על אלמנטים של `<blockquote>` `<q>` ותכונה `cite`

ישנו להיות שני אלמנטים שמשמשים באופן נפוץ עבור ציטוט (`quotations`) האלמנט `blockquote` ואלמנט `q`.

האלמנט **blockquote** מוגדר לפי **HTML 5** מפרט חלק של תוכן המצוטט ממקור אחר כאשר יכול להיות מן אתר אחר או קובץ אחר בתוך האתר.

האלמנט **blockquote** מוגדר לפי **HTML 4.01** הינו מגדיר ציטוט ארוך.

האלמנט **blockquote** משתמש בציטוט ארוך המשתרע על גבי פסקה. כאשר במילים אחרות ניתן להגדיר זאת ציטוט מלא.

הדפדפנים מובילים להזחה של התוכן של האלמנט **blockquote**, אולם אנחנו צריכים לא להשתמש באלמנט זה בלבד כדי להזיח חלק טקסט סתמי אלא באופן זה תצטרכו לעשות פעולה זו בשימוש ב **CSS**.

האלמנט **blockquote** משתמש בתכונה של **cite** שמצביעה לגבי מאיפה המקור שהציטוט נלקח. הערך שלו הינו כתובת **URL** שבה מפורט מידע לגבי הכתובת שהינה המקור לציטוט.

האלמנט **blockquote** הינו משתייך לקטגוריית אלמנטים שהם **block-level element**

האלמנט **q** משתמש בציטוט קצר שיכול להיות מוכל בתוך פסקה. הדפדפנים מניחים מרכאות " " סביב התוכן של האלמנט **q**.

האלמנט **q** משתמש בתכונה של **cite** שמצביעה לגבי מאיפה המקור שהציטוט נלקח. הערך שלו הינו כתובת **URL** שבה מפורט מידע לגבי הכתובת שהינה המקור לציטוט.

אלמנט משתייך לקטגוריית האלמנטים **inline elements**

הסבר על אלמנט **<abbr>** שזהו **abbreviations and acronyms**

אם אתם משתמשים בקיצורים או ראשי תיבות אז ניתן להשתמש באלמנט **<abbr>** כאשר האלמנט משתמש בתכונה **title** בתג הפותח כדי לפרט את המונח שעליו נעשה שימוש קיצור או ראשי תיבות במלואו.

התכונה בה נעשה שימוש **title** היא תכונה גלובלית כאשר מספקת מידע נוסף אודות האלמנט כאשר מציג את המידע בצורת **tooltip** שהינו ממשק גרפי במבנה חלון קטן שמציג מידע על איזור מסוים במסך ומקבלים מידע לגביו.

תפקידו של האלמנט **abbr** להגדיר לדפדפן ראשי תיבות וקיצורים לדוגמא "NASA" כאשר האלמנט אף מספק מידע למנועי חיפוש ולמערכות תרגום אודות הראשי תיבות והקיצורים.

ישנו עוד אלמנט כאשר אינו נתמך ב **HTML 5** כאשר ב **HTML 4.01** האלמנט **acronym** משתמשים בראשי תיבות ואילו אלמנט **abbr** משתמשים בקיצורים.

הסבר על אלמנטים **<cite><dfn>** citations and definitions

כאשר אתם מאזכרים חלק של עבודה כגון ספר, סרט, או נייר מחקר, אז יכולים להשתמש באלמנט **cite** כדי להצביע מאיפה מקור האזכור זה עפ"י **HTML 4.01** אבל לפי **HTML 5** הגדרת תפקידו של האלמנט להגדיר כותרת של עבודה.

האזכור יכול באופן שם הדבר שהוא מקור האזכור או כתובת URL שהיא מקור האזכור.

ב HTML 5 האלמנט `cite` צריך לא להשתמש בשמות של בני אדם אבל זה מאופשר ע"י HTML 4.01.

אופן הויזואלי שהדפדפנים מציגים את התוכן של אלמנט `cite` להיות בכתב נטוי.

לדוגמא של אמלנט `cite` :

`<p><cite>The Scream</cite> by Edward Munch. Painted in 1893.</p>`

הפעם הראשונה שאתם מסבירים איזשהו מינוח חדש (מושג אקדמי או ז'רגון) במסמך, זה להיות כהגדרת דוגמא של המינוח החדש. וזהו בעצם תפקיד האלמנט `dfn` כאשר משמש להצביע על הגדרת דוגמא של המינוח החדש במסמך HTML.

האלמנט צריך להיות בתוך האלמנטים הבאים או `p` או `section` או צמד `dd` | `dt`.

אם האלמנט `dfn` מכיל את התכונה `title` אז הגדרה של הדוגמא למינוח צריכה להיות בערך של התכונה `title`.

אם האלמנט `dfn` מכיל בתוכו רק את האלמנט `abbr` עם התכונה `title` אז הגדרה של הדוגמא למינוח יהיה בערך התכונה `title`.

הסבר על האלמנט של פרטי המחבר המסמך `<address>`.

האלמנט `address` יש למעשה שימוש ספציפי להכיל פרטי התקשרות למחבר של דף האינטרנט או בעלי האתר.

האלמנט זה יכול להכיל כתובת פיזית, למשלוח דואר אבל אין לעשות בו שימוש אלא אם כן זה הכתובת של יצירת קשר עם המחבר אך באופן כללי מומלץ בכתובת פיזית להשתמש באלמנט `p`. אך ניתן לעשות ביצירת הקשר באופנים הבאים כ-לדוגמא האלמנט עשוי להכיל מספר טלפון או כתובת דואר אלקטרוני.

האלמנט `address` בדרך כלל נכלל בו המידע הנמצא באלמנט `footer`.

האלמנט `address` את פרטי התקשרות המסמך מחברו או בעליו כאשר שוכן בתוך אלמנט `body`.

האלמנט `address` את פרטי התקשרות לגבי המאמר מחברו או בעליו כאשר שוכן בתוך אלמנט `article`.

האלמנט `address` יוצג ע"י הדפדפן באופן ויזואלי בכתב נטוי.

הסבר על אלמנט `<small>`

האלמנט **small** מגדיר את הטקסט קטן יותר בגודל אחד מגודל המוגדר כברירת מחדל בטקסט.

השימוש באלמנט זה נעשה לכתיבת הערות שוליים או כתיבת טקסט של כל הזכויות שמורות ושאר הטקסט המשפטי או אזהרות.

אין להשתמש באלמנט זה עבור כתיבה נרחבת בדף שכוללת פסקאות או רשימות שנעשה שימוש בלבד באלמנט זה. אלא להשתמש רק בטקסטים קצרים.

הסבר על אלמנט **<mark>**

כאשר אתם רוצים להדגיש חלק מהטקסט שלכם במרקור אתם משתמשים באלמנט **mark**. כאשר אלמנט זה מגדיר פעולת מירקור על גבי התוכן שבתוכו בצבע צהוב שהינו הגדרת ברירת המחדל של צבע מרקור.

האלמנט **mark** הינו אלמנט חדש הנתמך רק בגרסת **HTML 5**.

שימוש בו הינו נועד למרקור עבור מטרת אזכור או הפניה עקב רלוונטיות, להקשר מסוים לדוגמא זה יכול להיות בשימוש ע"י בדף אינטרנט להצגת תוצאות חיפוש כאשר מדגישים ע"י מרקור את כל דוגמאות של המילים המבוקשות כאשר הם מוכלים בתוצאות החיפוש.

בציטוט נעשית הדגשה של מרקור תוכן להפניה לדבר שהינו מחוץ לציטוט או הדגשה של מרקור לבדיקה ספציפית המשווה את הרעיונות שהמחבר המקורי לא החשיב אותם כחשובים.

אין להשתמש באלמנט זה עבור בתור תחביר הדגשה של סתם טקסט אלא יש להשתמש באלמנט **span** עבור מטרה זו.

צריך לשים לב לגבי ההבדל בין האלמנטים **mark** ו **strong** כאשר אלמנט **strong** נועד להדגיש טקסט שיש לו חשיבות לגבי כל שאר הטקסט בהקשר של התוכן המילולי הנוכחי, אבל משא"כ האלמנט **mark** מדגיש טקסט שהינו רלוונטי להקשר שונה.

הסבר על האלמנט **<bdo>** ותכונה **dir**

האלמנט **bdo** הינו קיצור של **Bidirectional Override** (שליטת הכיוונים) כאשר אלמנט זה מוגדר לשליטת הכיווניות של הטקסט. ותוצאה מכך האלמנט זה גורם לכיווניות של התווים של בטקסט יחד עם התכונה **dir** שבה מוגדר הכיווניות של כתיבת הטקסט.

התכונה **dir** בצירוף האלמנט **bdo** מוגדר הכיווניות של הטקסט הכוללת שני ערכים כאשר הערך הראשון הינו **ltr** כאשר מגדיר את כיוון הצגת הטקסט משמאל לימין, ערך השני הינו **rtl** כאשר מגדיר את כיוון הצגת הטקסט הינו ימין לשמאל כפי שבשפות עברית וערבית. וכתוצאה מכך התווים שבטקסט מתהפכים.

האלמנט **bdo** ללא התכונה **dir** אינו יכול לתפקד.

יש לציין שהתכונה **dir** הינה תכונה גלובלית כלומר שהיא לא ספציפית לאלמנט מסוים אלא ניתן לעשות שימוש בה באלמנטים אחרים. אך באלמנטים אחרים התכונה **dir** משנה את תצוגה של כיוון הטקסט לצד שונה.

הסבר על אלמנט **<bdi>**

האלמנט **bdi** הינו קיצור **Bidirectional Isolation** (בידוד דו כיווניות) כאשר מבודד את השתרעות של טקסט שמעוצב באופן בכיוון שונה משאר הטקסט שמחוצה לו. האלמנט זה להיות שמושי כאשר מטמיעים טקסט עם כיווניות שונה, מבסיס נתונים לדוגמא בתוך הטקסט עם כיווניות קבועה.

ניתן לעשות שימוש באלמנט זה לדוגמא כאשר ישנו טקסט המעורב עברית ואנגלית.

השימוש עם האלמנט **bdi** הוא מדובר בתוכן של גולשים כאשר הכיווניות לא ידועה כאשר גולשים יכולים להיות ממדינות שונות כך שהשפות ישנם כיוונים שונים של כתיבה מימין לשמאל או שמאל מימין.

אלמנט **bdi** הינו אלמנט חדש הנתמך אך ורק ע"י HTML 5

הסבר על אלמנטים **<ins>** ו ****

האלמנט **ins** הינו קיצור **insert** (הוכנס) מתייחס לטקסט שהוכנס למסמך כאשר מציג את הטווח של הטקסט שנוסף למסמך.

האלמנט זה בצירוף ביחד עם אלמנט **del** כאשר מתייחס לטקסט שנמחק כאשר מוצג ע"י הדפדפן בקו על הטקסט עצמו המציין שהינו מחוק כאשר **ins** מציין הטקסט החדש שנוסף במקומו.

בשני אלמנטים הללו משתמשים בתכונה **cite** המציגה כתובת **url** שבו מוצג הסיבה לשינוי התוכן והכנסת התוכן החדש ואף עוד תכונה של **datetime** המציג את התאריך שהטקסט הוכנס או שונה.

תחביר של התכונה **datetime** הינו שנה לאחר מכן חודש לאחר מכן יום בחודש ושימוש באות **T** כאשר רוצים לפרט אף השעה כאשר ה **T** הינו המפריד בין התאריך לזמן הזמן מוצג בשעה ולאחר מכן דקות ולאחר מכן שניות ולאחר מכן האות **Z** שהינה מייצגת לפי שעון גרינדיץ אמנם תכונה זו לא מוצגת ע"י הדפדפן אך יכול מוצג ע"י תוכנת קורא מסך.

דוגמא:

<p>This is a text.

<ins datetime="2012-09-15T22:55:03Z">This is an inserted text.</ins></p>

שאלות על הפרק

1. ציין מהם שני סוגי האלמנטים הקיימים ומה השימוש שלהם מציין ?
2. מה תפקידם של האלמנטים **heading** ?
3. מה נכון לומר גבי האופן גודל הטקסט של הכותרות שמוצג ע"י כל הדפדפנים ?
4. מה נכון לומר **CSS** ביחס לטקסט ?
5. הסבר מה תפקידו של האלמנט **p** ?
6. כיצד מוצגים הפיסקאות אשר מוגדרים בברירת המחדל ע"י הדפדפן ?
7. מה תפקידו של אלמנט **b** ?
8. מה תפקידו של אלמנט **i** ?
9. מה תפקידו של אלמנט **sup** ?
10. מה תפקידו של אלמנט **sub** ?
11. הסבר את המושג **white space collapsing** ומהו יתרון שמעניק שמעניק לכותבי הקוד ?
12. מהו תפקידו של אלמנט **br** והגדר במה תחביר שלו שונה מן מרבית האלמנטים ?
13. מהו תפקידו של אלמנט **hr** והגדר במה תחביר שלו שונה מן מרבית האלמנטים ?
14. הסבר מהם אלמנטים ריקים (**empty element**) ?
15. א.תן דוגמא של אלמנטים סמנטיים וציין איך הם מספקים מידע לגבי הטקסט?
 - ב. כיצד הדפדפנים מציגים אותו אלמנטים הללו ?
 - ג. מהו השימוש שאין לעשות בהם הסבר ?
 - ד. תאר כיצד שימוש אלמנטים שבדוגמא מסייע בפועל ?
16. א. מהו תפקידו של אלמנט **strong** ?
 - ב. וכיצד יוצג ע"י הדפדפן כאשר הוא מוגדר במצב של ברירת מחדל ?
17. א. מהו תפקידו של אלמנט **em** ?
 - ב. וכיצד יוצג ע"י הדפדפן כאשר הוא מוגדר במצב של ברירת מחדל ?
18. הסבר מה ההבדל בין האלמנטים **b** ו-**strong** שמוצגים באותו אופן ויזואלי ?
19. הסבר מה ההבדל בין האלמנטים **i** ו-**em** שמוצגים באותו אופן ויזואלי ?
20. א.הגדר את תפקידו של אלמנט **blockquote** עפ"י **HTML 4.01** ו **HTML 5**?
 - ב. תאר את מבנה אלמנט **blockquote** ?
 - ג. הגדר את תפקידה של התכונה **cite** ?
 - ד. כיצד יוצג האלמנט ע"י הדפדפן ?
21. א.הגדר את תפקידו של אלמנט **q** ?
 - ב. האם תפקידה של התכונה **cite** האלמנט זה הוא כאלמנט **blockquote** ?
 - ג. תאר את מבנה האלמנט **q** ?
 - ד. כיצד הוא מוצג ע"י הדפדפן ?
22. א. תאר את מבנה האלמנט **abbr** ?
 - ב. הגדר את תפקידה של התכונה **title** ?
23. איזו תוכנות עושות שימוש באלמנט **abbr** ?
24. מהו ההבדל בין האלמנט **acronym** לאלמנט **abbr** ?
25. הגדר מהו תפקידו של אלמנט **cite** ותאר את מבנהו?
26. כיצד יוצג התוכן באלמנט **cite** ע"י הדפדפן ?
27. א. הגדר את תפקידו של האלמנט **dfn** ?
 - ב. הגדר את מבנה התחביר של אלמנט **dfn** ?
28. כיצד יוצג האלמנט **dfn** ע"י הדפדפן ?
29. א. הגדר את תפקידו של אלמנט **address** ?
 - ב. הגדר את מבנה התחביר של אלמנט **address** ?
30. איזה אופן לא מומלץ להכניס לתוך אלמנט **address** ?
31. כיצד יוצג תוכן האלמנט **address** ע"י הדפדפן ?

32. א. הגדר את תפקידו של אלמנט `small` ?
 ב. הגדר את מבנה התחביר של אלמנט `small` ?
33. כיצד יוצג התוכן באלמנט `small` ע"י הדפדפן ?
34. א. הגדר את תפקידו של אלמנט `mark` ?
 ב. הגדר את מבנה התחביר של אלמנט `mark` ?
35. כיצד יוצג התוכן של אלמנט `mark` ע"י הדפדפן ?
36. מהו יחוד של אלמנט `mark` ביחס לגירסאות שונות של `HTML` ?
37. א. הגדר את תפקידו של אלמנט `bdo` ?
 ב. הגדר את מבנה התחביר של אלמנט `bdo` ?
38. א. הגדר את התכונה `dir` ?
 ב. מה מאפיין את התכונה `dir` ?
39. א. הגדר את תפקידו של אלמנט `bdi` ?
 ב. הגדר את מבנה התחביר של אלמנט `bdi` ?
40. מה יחוד של אלמנט `bdi` ביחס לגרסאות קודמות של `html` ?
41. א. הגדר את תפקידו של אלמנט `ins` ?
 ב. הגדר את מבנה התחביר של אלמנט `ins` ?
42. א. הגדר את תפקידו של אלמנט `del` ?
 ב. הגדר את מבנה התחביר של אלמנט `del` ?
43. א. הגדר את תחביר התכונה `cite` של האלמנטים `ins` ו `del` ותפקידה?
 ב. הגדר את תחביר התכונה `datetime` של האלמנטים `ins` ו `del` ותפקידה?
44. א. הגדר את תפקיד אלמנט `section` ?
 ב. הגדר את תחביר האלמנט `section` ?
45. מה מאפיין את אלמנט `section` ביחס לגרסאות קודמות של `html` ?
46. לאיזו קטגוריה של אלמנטים משתייך האלמנט `section` הסבר ?

פרק meta tag ותכונותיו

תוכן:

1. הסבר על אלמנט `<meta>`
2. הסבר על תכונת `content`
3. הסבר על תכונת `charset`
4. הסבר על תכונת `http-equiv`

הסבר על אלמנט `<meta>`

אלמנט `meta` הינו מספק נתונים המכילים מידע אודות נתוני המסמך `HTML`.

האלמנט `meta` משמש למידע לפירוט אודות מחבר של מסמך `HTML`, תיאור הדף, מילות מפתח, ומתי עודכן לאחרונה הדף ועוד נתונים אודות נתוני המסמך.

באלמנט `meta` ישנו שני סוגי משתנים משתני מערכת ומשתני המשתמש כאשר משתנה המערכת מספק מידע למערכת כגון רענון הדף מתי יש לרענן את הדף ומשתני המשתמש הנותן מידע למשתמש כגון מחבר הדף ותיאור הדף.

באלמנט זה משתמשים בו הדפדפן איך להציג את תוכן דף האינטרנט או לטעון מחדש את הדף. ומנועי חיפוש משתמשי בו במילות מפתח, ועוד שירותים מבוססי אינטרנט.

ניתן לעשות שימוש באלמנט **meta** על מנת לקדם את האתר במנועי חיפוש. כאשר הרובוט או ספיידר של מנועי חיפוש מבקרים בדף אינטרנט ומוסיפים את המידע שבאלמנט **meta** למנועי חיפוש.

האלמנט **meta** תמיד שוכן בתוך האלמנט **head** שזהו אלמנט שמוגדר המידע אודות דף האינטרנט. כך שאינו מוצג ע"י הדפדפן עצמו אלא רק בהצצה לקוד המקור המציג את פרטי הדף.

האלמנט **meta** תמיד צריך להיות מוצג עם **attribute** מאחר והוא אלמנט ריק וכך שאין לו תג סוגר אלא רק תג אחד כאשר השם וערך של התכונה להיות באותו תג היחיד.

הסבר על תכונת content

התכונה **content** יכולה להיות מוגדרת רק אם ישנו קודם לכן את התכונה **name** או התכונה **http-equiv** בלבד מאחר והיא נותנת ערך של טקסט שהינו ערך המשוייך לערכים לתכונות **name** או **http-equiv**.

הסבר על תכונת charset

התכונה **charset** מפרטת מהו הקידוד של מסמך **HTML**. התכונה **charset** ניתן לעקוף את שימושה כאשר משתמשים בתכונה **lang** כאשר מפרשת לדפדפן באיזו שפה מדובר כך שהדפדפן יכול להציג את הדף באופן תקין. מסיבה זו כאשר לא משתמשים בתכונה **lang** אז שישנו חשיבות להשתמש ב **charset** על מנת שהדפדפן יציג אותו באופן נכון דפים בשפות שונות.

הערכים של התכונה **charset** שמפרטים את התווי הקידוד עבור מסמך ה- **HTML** כאשר הנפוצים בהם הם **UTF-8** תו קידוד עבור יוניקוד, או **ISO-8859-1** שהינו קידוד לאלפבית הלטיני ואף זה ברירת המחדל של של **chrset**.

התחביר של התכונה **charset** הינו:

```
charset = character_set
```

ההבדל בין **html 4.01** ל- **html 5** לגבי התכונה **charset** להיות כאשר ב- **HTML 5** התכונה

charset להיות בנפרד בתג **meta** משא"כ ב **HTML 4.01** התכונה היא חלק מ- **http-equiv**

הסבר על תכונת http-equiv

תכונת **http-equiv** להיות בשימוש ע"י השרת על מנת לאסוף מידע אודות הדף אינטרנט בשימוש **http header**. התכונה **http-equiv** של אלמנט **meta** המשובצת באלמנט, דומה במהותה ל- **http header** בדמיון ערכים שבהם. התכונה מאפשרת לכם לשלוח מידע נוסף

לדפדפן ב- **http header**. ואף נשלחת לדפדפן קודם דף האינטרנט עצמו כמידע לגבי אודות דף האינטרנט.

התחביר של תכונת **http-equiv**
<meta http-equiv="value" content="value"/>

אין להכליל את תכונת **name** באלמנט **meta** כאשר תכונת **http-equiv** כלולה בה.

יש להכליל את תכונה של **content** בתכונה של **http-equiv**.

הסבר על הערך **expires** בתכונת **http-equiv**

ניתן להכניס לתוך התכונה **http-equiv** ערך **expires** ולאחר מכן ערך המתייחס ל **expires** הינו **date**. הערך **date** בתוך התכונה **content** מתייחס לתאריך והזמן שהדף אינטרנט להיות נשמר במטמון לפוג (**expires**) כאשר לאחר שעבר זמן המפורט הוא מספק הוראה על הדפדפנים לא לשמור במטמון את דף האינטרנט ההוא. ואף כשישנו כבר את הדף האינטרנט שמור במטמון קודם לתאריך אעפ"כ הדף יטען מחדש כאותו אופן שאין דף אינטרנט מאותו אתר במטמון של הדפדפן. וכך ע"י התכונה **http-equiv expires** ניתן לשלוט על איך הדפדפנים יציגו את הדף האינטרנט עבור הקוראים אם ביחס שימוש במטמון הדפדפן ואף מנועי חיפוש מוחקים את המיפתוח דף אינטרנט הזה מבסיס הנתונים שלהם לאחר סיום תאריך המפורט

השימוש בו חשוב ב **expires meta's tag** עבור אתרי אינטרנט שמעדכנים לעיתים קרובות את האתר ורוצים שמנועי החיפוש לא ישתמשו עותק ישן ושהדפדפן לא ישתמש במטמון של אותו דף השמור אצלו וכתוצאה מכך המשתמשים יראו את הגרסה הנוכחית של הדף אינטרנט.

התחביר **http-equiv="expires" content="date"**
בערך התאריך צריך להופיע יום באופן מקוצר בשלושת האותיות הראשונות של שם היום ולאחמ"כ יום בחודש ולאחמ"כ שם החודש באופן מקוצר בשלושת האותיות הראשונות של שם החודש ולאחמ"כ שנה ולאחמ"כ שעה דקות ושניות עם קיצור **GMT** שהינו קיצור **Greenwich Mean Time**. כך זה יראה **Sun, 15 Dec 2016 17:22:20 GMT**

הסבר על ערך **refresh** בתכונת **http-equiv**

ניתן להכניס לתוך התכונה **http-equiv** ערך **refresh** ולאחר מכן הערך של **url** עם **second** בתוך התכונה **content** כאשר מתייחס לזמן שבו אנו מגדירים את הזמן שעל הדפדפן לרענן את הדף אינטרנט ויטען

התחביר יראה כך בתחילה מצוין הזמן של השניות שבו ירוענן הדף ויטען מחדש ולאחר מכן כתובת **URL** של הדף אינטרנט וכך זה יראה

http-equiv="refresh" content="20;url=http://www.example.com/index.html"

יתרון של שימוש **http-equiv -refresh** הינו בכך שניתן לרענן דף אינטרנט באתר שהתוכן משתנה באופן דינמי ללא צורך בשימוש בג'אווה סקריפט.

הסבר על ערך **X-UA-compatible** בתכונת **http-equiv**

ניתן להכניס לתוך התכונה **http-equiv** ערך **X-UA-compatible** שמאפשר ליוצר הדף אינטרנט לבחור מהו הגרסה של **IE** שתוכן הדף אינטרנט יוצג באופן שתואם את אותה

גירסה בכך שכל אותם תכונות המתקיימות באותו גירסת דפדפן IE יתקיים באותו דף כאשר הוא מפרט בערך של התכונה **content** שמגיע מיד לאחר התכונה **http-equiv** עם הערך **X-UA-compatible** את הערך IE שמתייחס לדפדפן אינטרנט אקספלורר .

X-UA-compatible אינו הופך את הדפדפן שיפעל בעצם כמדמה למבנה הדפדפן ישן שאותו הגדרנו בערך של IE אלא רק מגדיר את תצוגת התוכן בדף אינטרנט שיוצג כאותו ערך של IE

ערכים של התכונה יכולים להיות

1. "IE=5"
2. "IE=EmulateIE7"
3. "IE=7"
4. "IE=EmulateIE8"
5. "IE=8"
6. "IE=EmulateIE9"
7. "IE=9"
8. "IE=EmulateIE10"
9. "IE=10"
10. "IE=Edge"

ערך של IE=5 זאת אפשרות שמגדירה שתצוגת התוכן תוגש כפי שאופן התצוגה של התוכן במסמך הייתה מוגשת דומה לכפי שמוצג ב-IE5.

ערך של IE=EmulateIE7 זאת אפשרות שמגדירה בכך שIE מבצע זאת באמצעות פקודה שמגדירה שתצוגת התוכן תוגש ותוצג באופן שמדמה לחלוטין סביבת הדפדפן IE7.

ערך של IE=7 זאת אפשרות שמגדירה בכך שתצוגת התוכן תוגש ותוצג באופן שמדמה לחלוטין סביבת הדפדפן IE7.

ערך של IE=EmulateIE8 זאת אפשרות שמגדירה בכך שIE מבצע זאת באמצעות פקודה שמגדירה שתצוגת התוכן תוגש ותוצג באופן שמדמה לחלוטין סביבת הדפדפן IE8.

ערך של IE=8 זאת אפשרות שמגדירה בכך שתצוגת התוכן תוגש ותוצג באופן שמדמה לחלוטין סביבת הדפדפן IE8 כאשר כולל תמיכה מליאה ב-CSS2 ותמיכה מוגבלת ב-CSS3.

ערך של IE=EmulateIE9 זאת אפשרות שמגדירה בכך שIE מבצע זאת באמצעות פקודה שמגדירה שתצוגת התוכן תוגש ותוצג באופן שמדמה לחלוטין סביבת הדפדפן IE9.

ערך של IE=9 זאת אפשרות שמגדירה בכך שתצוגת התוכן תוגש ותוצג באופן שמדמה לחלוטין סביבת הדפדפן IE9 כאשר כולל תמיכה מליאה ב-HTML 5 בגירסת draft (טיוטה) ובתמיכה ב-CSS3 למעט אמיניציה שפועלת ב-CSS3.

ערך של IE=EmulateIE10 זאת אפשרות שמגדירה בכך שIE מבצע זאת באמצעות פקודה שמגדירה שתצוגת התוכן תוגש ותוצג באופן שמדמה לחלוטין סביבת הדפדפן IE10.

ערך של IE=10 זאת אפשרות שמגדירה בכך שתצוגת התוכן תוגש ותוצג באופן שמדמה לחלוטין סביבת הדפדפן IE10 כאשר כולל תמיכה מליאה ב-HTML 5 ובתמיכה ב-CSS3 ועוד.

ערך של IE=edge זאת אפשרות שמגדירה בכך ש IE מבצע זאת באמצעות פקודה שמגדירה שתצוגת התוכן תוגש עם תמיכה מיטבית בכך שתוצג בסביבת הדפדפן שבו אנו פועלים כאשר אם זה IE9 אזי ידמה לחלוטין את תצוגת התוכן בסביבת IE9 באופן שזה מוצג בדפדפן IE9.

ערך של Chrome=1 זאת אפשרות להגדיר לדפדפני IE ש תוסף (plug-in) Google Chrome Frame שמוסיפים אותו לדפדפני IE כאשר זה קיים אצל המשתמש הדפדפן יציג אף במצב תצוגה שנותן התוסף זה המאפשר תכונות של הדפדפן Chrome בדפדפני IE. זה נחוץ על מנת להפעיל את plug-in זה לשים את הערך chrome=1

התחביר יראה כך:

```
<meta http-equiv="X-UA-compatible" content="IE-גירסת הדפדפן"/>
```

תחביר נפוץ כאשר הינו נותן את התמיכה המיטבית בתכונות של אותו דפדפן ואף תכונות של chrome בדפדפן הינו:

```
<meta http-equiv="X-UA-compatible" content="IE=Edge,chrome=1"/>
```

פרק רשימות (LIST) ב- HTML

תוכן:

1. מבוא על עניין רשימות
2. הסבר על שלושת סוגי הרשימות.
3. הסבר על אלמנט של הרשימה הממוספרת ותכונותיו.
4. הסבר על אלמנט של הרשימה הלא ממסופרת.
5. הסבר על אלמנט ותכונותיו.
6. הסבר על אלמנט <dl> .
7. הסבר על אלמנט <dt> .
8. הסבר על אלמנט <dd>
9. הסבר על רשימות מקוננות.

מבוא על עניין רשימות

רשימות להיות בשימוש עבור קיבוץ של חלקי מידע הקשורים אחד לשני ביחד וע"י כך שהם יוצרים קשר ברור עם כל האחרים ברשימה וקל לקרוא את מידע זה.

בעיקר משתמשים ברשימות לניווט באתרים וגם בתוכן כללי.

יש להעדיף יצירת רשימות באופן מובנה בשימוש אלמנטים שמשתייכים לקבוצת האלמנטים האחרים ליצירת רשימות מאשר ליצור נקודות מובנות באופן מובנה בשימוש באלמנטים המשתייכים לקטגוריית מובנים מאחר שהאלמנטים המשתייכים ליצירת רשימות מובנים היטב שע"י יצירת הרשימות הם מקלים על התחזוקה של הדף ואף יוצרים יותר נגישות בכך שקל לקרוא יותר המידע מסיבות שפירטנו בראשית דברינו בעניין הרשימות.

הסבר על שלושת סוגי הרשימות

HTML מספק לנו שלושה סוגי רשימות:

1. **order list** (רשימה ממוספרת).
2. **unordered list** (רשימה לא ממוספרת)
3. **description list** (רשימת הגדרות)

Order list הינה רשימה ממוספרת כאשר כל הפריטים ברשימה בה ממוספרים. לדוגמא הרשימה עשויה להכיל קבוצה של צעדים עבור מתכון שצריך להיות מבוצע בסדר, או חוזה משפטי כאשר כל הנקודות בו צריכות להיות מזוהות במספר המקטע .

Unordered list הינה רשימה לא ממוספרת כאשר מתחילה בנקודת כדור (שהינם יותר מאשר שהם תווים המצביעים על סדר).

description list. הינה רשימה שמורכבת מקבוצה של מונחים שביחד עם הגדרות כל המונחים המפורטים ברשימה.

לגבי הרשימה השלישית היא נקראה ב- **HTML 4.01 definition list**

הסבר על אלמנט של הרשימה הממוספרת ותכונותיו

האלמנט **ol** הינו אלמנט האחראי ליצירת רשימה ממוספרת שיכולה להיות מספרית או בספרות רומיות או באותיות או אפילו ע"י כדורים פשוטים כאשר בחירת סגנון עיצוב המספור הינו ע"י **CSS** ולא ע"י **HTML** עצמו.

באלמנט **ol** ישנו תכונה **start** כאשר מפרטת את הערך התחלה של פריט ברשימה הממוספרת. הערך המוגדר בברירת המחדל הינו מס' 1.

באלמנט **ol** ישנו תכונה **type** המאפשר לבחור מהו הערך הממוספר מספר ע"י השמת הערך 1, או ספרות רומיות גדולות ע"י השמת אות גדולה I או ספרות רומיות באותיות קטנות ע"י השמת אותיות קטנות, אותיות גדולות ע"י השמת אות A או אותיות קטנות ע"י השמת הערך האות הקטנה a .

באלמנט **ol** ישנו תכונה **reversed** אשרר מציגה את המספור בסדר יורד את הרשימה בניגוד לברירת המחדל שבו המספור מוצג בסדר עולה אך הפריטים מוצגים כפי הסדר שמתויגים במסמך וכל השינוי הינו במספור. תכונה זו נתמכת רק ב- **HTML 5** ולא נתמכת בדפדפן **internet explorer** אלא ע"י שאר הדפדפנים העיקרים ככרום, ואופרה, וספארי.

התכונה **reversed** נחשבת **Boolean attribute** (תכונה בוליאנית) מאחר ומניתן לעשות בה שימוש רק ע"י שם התכונה וערך הוא רק שם התכונה כאשר ב- **html** בתכונה בוליאנית אין חובה לכתוב את הערך. כאשר בעצם לתכונה בוליאנית אין שום רלוונטיות לערך.

ליצירת הרשימה צריך להשתמש יחד עם האלמנט **li**.

הסבר על האלמנט **** של הרשימה הלא ממסופרת

האלמנט **ul** הינו אלמנט אחראי ליצירת רשימות לא ממסופרות כאשר כל פריט ברשימה מיוצג בתבליט ע"י הדפדפן במצב ברירת מחדל ניתן לשנות זאת בשימוש **CSS** כאשר מאפשר ליצור כאשר כל פריט ברשימה יהיה מיוצג ע"י ריבוע או יהלום או עיגול.

באלמנט **ul** התכונות שלו לא נתמכות ב- **HTML 5** על במקום זאת יש להשתמש ב- **CSS**.

ליצירת הרשימה צריך להשתמש יחד עם האלמנט **li**.

הסבר על האלמנט **** ותכונותיו

האלמנט **li** מגדיר פריט ברשימה כאשר נעשה בו שימוש באלמנטים **ol** ו- **ul**. ואף המילה **li** הינה קיצור **list item** (פריט רשימה).

באלמנט **li** ישנו תכונה הנקראת **value** כאשר מספר הרשימה גדל בסדר המספרים מהמספר המפורט בערך של **value**.

התכונה **value** יכולה להיות שימוש רק ברשימה ממסופרת שזהו בתוך האלמנט **ol** מאחר וזה מתייחס למספור.

התכונה **value** מיישמים רק בפריט הראשון ברשימה מאחר ושאר הפריטים ברשימה ממסופרים החל מאותו ערך מספרי שצוין בתכונה **value** בתחילה.

הסבר על אלמנט **<dl>**

האלמנט **dl** מגדיר רשימת תיאורים של הגדרות שזה אף הקיצור של **dl** **description list** (רשימת הגדרות). רשימת תיאורים של הגדרות מורכבת ממונחים ותיאורם. ואך יתכן אירועים וזמנים.

רשימת תיאורים עם האלמנט **dl** הינו יחד עם האלמנטים **dt** ו- **dd**

הסבר על אלמנט **<dt>**

באלמנט **dt** להיות בשימוש על מנת להכיל מונח שלהיות מוגדר.

הסבר על אלמנט <dd>

האלמנט **dd** מכיל תיאור הסבר של המונח השוכן בתוכן של האלמנט **dt**.

מוצג בהזחה ע"י האלמנט

הסבר על רשימות מקוננות

אתם יכולים לקונן רשימות בתוך רשימות אחרות לדוגמא כאשר אתם רוצים למספר רשימה עם נקודות נפרדות התואמת לאחד מפריטי הרשימה. מספר ברשימה המקוננת הינו נספר באופן נפרד מן הרשימה שבו הוא מקונן אך ניתן סופרו כאחד באופן שמשמשים בתכונת **start**. ברשימה מקוננת האלמנט **li** הקינן נעשה כאשר הרשימה נוצרת בתוך פריט רשימה שהיא המקור כאשר התג הסוגר של אותו אלמנט **li** שבו מקונן הרשימה נמצא מתחת לאותו רשימה המקוננת.

השימוש ברשימה מקוננת טוב ליצירת טבלה של תוכן המכילה תפריט של תוכן הנכתב ברשימה כאשר באופן שישנם ערכים שכוללים ערכים משניים.

בדפדפן מציג את הרשימה המקוננת בהזחה רחוק מן רשימת המוצא ואף ברשימה לא ממוספרת התבליטים המוצגים בברירת המחדל מוצגים קצת שונה מן המקור.

רשימה מקוננת יכולה להכיל קינן של אלמנט **ul** בפריט ברשימה של אלמנט **ol** וכן להיפך.

פרק הלינקים (links)

תוכן:

1. מבוא על לינקים.
2. רשימת סוגי לינקים.
- 3 טיפים חשובים ביצירת לינקים.
4. לינקים לאתר אינטרנט אחר.
5. לינק לכתובות e-mail.
6. הבנת הספריות ומבנה הספריות.
7. הבנת כתובת URL.
8. כתובת Absolute URL וכתובת Relative URL
9. הסבר על האלמנט **<base>**
10. יצירת לינקים בתוך הדף עם אלמנט **<a>** ותכונת **href**
11. יצירת לינק לחלק אחר בדף בשימוש התכונות **id** ו **name**

12. תכונות נוספות של האלמנט <a>

מבוא על לינקים

לינקים (links) הם תכונה מוגדרת של האינטרנט היות והם מאפשרים לך לנוע מדף אינטרנט אחד לדף אינטרנט אחר – כאשר שזה עצם מה שמאפשר את הרעיון של דפדוף והגלישה באינטרנט.

הלינקים יכולים להיות ע"י מילים או ביטוי או תמונה.

רשימת סוגי לינקים

1. לינק מאתר אינטרנט אחד לאתר אינטרנט אחר ומכונה external link
2. לינק מדף אינטרנט אחד לדף אינטרנט אחר באותו אתר אינטרנט ומכונה internal link
3. לינק מחלק אחד של הדף אינטרנט לחלק אחר של אותו דף.
4. לינק שנפתח בחלון דפדפן חדש.
5. לינק שמפעיל את תוכנת המיילים שלכם וממענת אימייל חדש לאיזשהו כתובת המפורטת.

טיפים חשובים ביצירת לינקים

1. כאשר אתם יוצרים איזשהו לינק אתם צריכים לנסות לעשות זאת באופן יעיל ע"י שנשתמש במילים שנותנים למשתמשים לדעת מה הם יראו אם ילחצו על הלינק על מנת ליצור חווית משתמש הסיבה לחשיבות עשיית הדבר הנ"ל כאשר הלינקים בדרך כלל מוצגים בצבע שונה משאר הטקסט הסובב אותם וכך מבליט אותם יותר משאר הטקסט הסובב אותם. וכתוצאה מכך הרבה משתמשים סורקים בעיניהם את הדפים עבור מציאת לינקים כאשר הם רוצים להגיע לדף הבא ללא קריאת הדף בכללותו. סיבה אחרת עקב כך שמשתמשים נוטים יותר להשתמש באתר אינטרנט שלכם כאשר הלינקים קלים לקריאה שזה דבר טוב יותר שישנו הסבר על הלינקים מאשר הוראה לחץ כאן.
2. כאשר אתם משתמשים בתמונה בתור לינק אתם צריכים לייצור וודאות שהתמונה נותנת אינדיקציה ברורה של איפה הלינק שהתמונה משמשת בתור לינק לוקח את המשתמשים. על כן אתם יכולים להשתמש בתכונה **title** על הלינקים כאשר הערך של התכונה **title** צריך להיות התיאור של איפה הלינק הזה לוקח אותכם כאשר יוצג ב **tooltip** כאשר הסמן יהיה על גבי הלינק. וזה יכול להיות שימושי באופן אתם משתמשים בתמונה בתור לינק.
3. על מנת לייצור לינק טוב יש לחשוב לגבי המילים כאשר המשתמש עושה בהם שימוש כאשר מחפש בהקשר למקום שבו הלינק מתקשר איליו כלדוגמא אם יוצר לינק שמקשר אותך למסעדה איטלקית כלשהיא באיזשהו מקום אז יש לציין שהלינק הינו מסעדה איטלקית. באותו מקום ספציפי ולא ציון באופן כללי של רק מסעדה איטלקית.

לינקים לאתר אינטרנט אחר

הלינקים שנוצרים בשימוש האלמנט a יש תכונה הנקראת href. כאשר הערך של התכונה href הינו דף האינטרנט שאנשים רוצים לגשת אליו מתי שהם לוחצים על הלינק.

המשתמשים יכולים ללחוץ על התוכן כלשהו שמופיע בתוך האלמנט a וייקח אותם לדף האינטרנט המפורט בתכונה href.

כאשר אתם מתקשרים לאתר אינטרנט שונה הערך של התכונה href יהיה כתובת אינטרנט מלאה של אותו אתר עבור יצירת קישור לאתר כאשר הכתובת ידועה absolute URL.

הדפדפנים אמורים להציג במצב של ברירת מחדל את הטקסט של הלינקים בצבע כחול וקו תחתון מתחתיהם.

לינק זה מכונה external link

Absolute URL

URL הינו ראשי תיבות של Uniform Resource Locator כאשר בתרגום זה מען משאבים אחיד. כל דף אינטרנט יש את URL שלו. זה בעצם להיות כתובת האינטרנט שאתם מזינים לתוך הדפדפן אם אתם רוצים לבקר בדף אינטרנט ספציפי.

בכתובת אתר אינטרנט מתחיל עם שם דומיין עבור האתר ולאחר מכן מקליד את הקובץ שהינו נתיב לדף האינטרנט הספציפי ע"י הפירוט של אותו הדף המבוקש. אם אין דף אינטרנט מפורט איליו רוצים להגיע באתר אינטרנט אזי האתר יציג את הדף הבית.

הבנת הספריות ומבנה הספריות

ספרייה הינה שם אחר פשוט לתיקייה בתוך האתר אינטרנט: באותו דרך שהכונן דיסק קשיח שלכם מכיל תיקיות שונות כך ניתן לומר לגבי אתר אינטרנט שהינו מכיל תיקיות. בדרך כלל אתם מוצאים שאתר אינטרנט מכיל ספריות אחדות ושכל ספרייה מכילה חלקים שונים של האתר אינטרנט. לדוגמא אתר אינטרנט גדול עם חלקים משנים אחדים יכול ספריות נפרדות עבור כל חלק של האתר וגם ספריות שונות עבור סוגים שונים של קבצים (לדוגמא תמונות עשויים לשכון בתוך בספרייה אחת וגיליון עיצוב בספרייה אחרת).

באותו דרך שאתם קרוב לוודאי מארגנים את הקבצים בכונן דיסק קשיח שלכם לתוך תיקיות נפרדות, אתם צריכים לארגן את הקבצים באתר אינטרנט שלכם כך שאתם יכולים למצוא מה שאתם רוצים להסתכל בו ביותר קלות ושומרים על שליטה על כל הקבצים. כשאתם יכולים לדמיין שאם כל הקבצים שבשימוש באתר האינטרנט ישכנו בתוך אותו ספרייה, שהספרייה באופן מהיר תהפך לגדולה ומסובכת.

לדוגמא של מבנה ספרייה של אתר חדשות עם תיקיות נפרדות עבור כל חלק. החלק Music יש את התיקיות שלו לחלקים משניים שהם תכונות, MP3, Reviews. ובנוסף התיקייה הראשית יש תיקיות נפרדות עבור סוגים שונים של קבצים שנעשה בהם שימוש באתר: תמונות,סקריפט, וגיליון עיצוב.

כאשר אתם מתחילים לבנות אתר אינטרנט אתם צריכים לייצור מבנה ספרייה טוב שיכול להתמודד עם גידול. זה להיות מפתיע איך אתר אינטרנט קטן יכול באופן מהיר לגדול ולהכיל הרבה יותר קבצים מאשר אתם בתחילה מדמיינים.

ספריית root (או תיקיית root) היא הספרייה הראשית שמחזיקה את כל האתר אינטרנט שלכם במקרה הזה שבדוגמה שבתמונה זה נקרא exampleNewsSite.com .

ספרייה משנית היא ספרייה שבתוך ספרייה אחרת. כאן בדוגמה שבתמונה Film היא ספרייה משנית של Entertainment .

ספריית מוצא היא ספרייה אחת שמכילה ספרייה אחרת . כאן Entertainment היא ספריית המקור של Arts, Film, Music, TV /

לינק לכתובות e-mail

אתם קרוב לוודאי ראיתם דף אינטרנט עם כתובת e-mail שכאשר לוחצים על הכתובת, זה פותח אימייל חדש בתוך תוכנת e-mail שלכם, עם שדה שמאוכלס מראש כתובת אימייל.

כדי לייצור לינק לכתובת e-mail , אתם צריכים להשתמש בתחביר הבא כבדוגמא:
E-mail us.

כאן הערך של תכונת href המתחילה עם מילת המפתח mailto לאחר מכן נקודתיים ואחר כך כתובת המייל שאתם רוצים שהדואר אלקטרוני ישלח לה. הלינק עם mailto: באלמנט a הוא כמו שאר הלינקים האחרים שהתוכן של אלמנט a הינו חלק החזותי של הלינק המוצג בדפדפן . E-mail us .

ישנו חסרון אחד כתובת אימייל שלכם על הדף אינטרנט: שישנו פה התנהלות פחות קפדנית לגבי הכתובת אימייל כאשר נעשה שימוש באינטרנט בתוכנות קטנות שבאופן אוטומטי מחפשות כתובת אימייל באתרי אינטרנט שמכילים כתובות אימייל. אחרי שהם מוצאים את הכתובות אימייל באתר אינטרנט הם מתחילות לשלוח ספאם לכתובות אימייל הללו.

ישנו חלופות ליצירת לינק לכתובת אימייל:

1. משתמשים בטופס e-mail (טופס יצירת קשר) שהמבקרים באתר משלימים אותו כשיתרון בשימוש זה במקום לשים לינק לכתובת אימייל על דף הינו בגלל שהתוכנות המחפשות את הכתובות אימייל באתרי אינטרנט באופן אוטומטי אינם יכולים להשתמש בטפסי יצירת קשר על מנת לאסוף כתובות אימייל. החיסרון כאשר לטופס e-mail (טופס יצירת קשר) נדרש סקריפט לרוץ על שרת אינטרנט (הנכתב בשפות כגון ASP.NET או PHP).

2. כתבו את כתובת האימייל שלכם בתוך הדף האינטרנט בשימוש בJavaScript עם jQuery הרעיון שעומד מאחורי הטכניקה זו זהו היתרון שהתוכנות הסורקות את האינטרנט עבור

מציאת כתובות e-mail שאינם יכולים לקרוא כתובות e-mail בגירסת JavaScript

הבנת כתובת URL

Uni Resource Locator או בקיצור URL שמשמעותו מען משאבים אחדים כאשר מפרט איפה אתם יכולים למצוא את המשאב על האינטרנט כאשר אתם קרוב לוודאי רבים מכם המשתמשים חושבים על זה ככתובת אינטרנט. כאשר אתם נעים בסביבת אינטרנט אתם רואים URL של כל דף אינטרנט בשורת הכתובת בדפדפן שלכם.

כתובת URL מחולקת לשלושה חלקי מפתח:
scheme.1

host address .2

filepath .3

דוגמא לכתובת URL: <http://www.mako.co.il/news?Partner=sidebar>

ה- http:// הינו החלק ה-scheme (סכמה) המזהה את הדרך שהקובץ מועבר אליכם כאשר בעצם באמצעות איזה פרוטוקול מועבר אליכם המידע . מרבית דפי האינטרנט משתמשים במשהו שנקרא HyperText Transfer Protocol (HTTP) להעביר מידע לכם כאשר זה להיות הדרך שמרבית דפי אינטרנט מתחילים את הכתובת URL עם http:// . למרות שאתם עשויים להבחין שישנם תחיליות אחרות כגון https:// כאשר משמש כגון לשירותי בנקאות אונליין כאשר להיות מבנה של http יותר מאובטח. או ftp:// כאשר משמש להורדת קבצים גדולים.

ה- host address (כתובת המארח) להיות לרוב השם דומיין עבור האתר לדוגמא mako.co.il לעיתים קרובות אתם רואים www לפני שם הדומיין: למרות שזה לא להיות חלק ממשי של השם דומיין. כתובת המארח יכולה להיות מספרים הנקראים כתובת IP .

הערה: כל המחשבים המתקשרים לאינטרנט משתמשים בכתובת IP . כתובת IP הינה קבוצה שמורכבת מספרות שאורכן עד 12 ספרות המופרדות בסמלי נקודות. כאשר אתם מכניסים שם דומיין לתוך הדפדפן, הדבר שעומד מאחורי הכנסת שם הדומיין הינו שהשם מומר לתוך כתובת IP של המחשב שמאחסן את האתר האינטרנט. וזה מבוצע ע"י השגת מידע של הפרוטוקול Domain Name System DNS בתוך המידע של שמות דומיין וכתובות IP השמורים אצלו בספרייה ומתאים אותם לכתובת IP שלהם. מעניין לציין בנקודה אחת שבאינטרנט נגמרו כתובות IP כך ש- (IETF) הציגו Internet Protocol Version 6 (IPV6) כאשר להיות מערכת חדשה של כתובות IP שמאפשרות להרבה יותר כתובות זמינות.

ה- file path (נתיב של ספריות הקבצים) תמיד מתחיל עם תו של לוכסן קדמי ועשויה להכיל שמות של הספריות אחד מהם או יותר. (הספריות להיות רק שמות אחרים לתיקיות בשרת אינטרנט.) ה- file path עשוי להסתיים עם שם הקובץ בסוף. לדוגמא news/newsIsrael.html

ה- file path לרוב תואם למבנה הספרייה של האתר אינטרנט, כך במקרה של הדוגמא פה אנחנו יכולים למצוא את הדף אינטרנט newsIsrael.html בספרייה הנקראת news .

זה לא להיות רק הדפי אינטרנט שיש להם את הכתובת URL שלהם. כל קובץ באינטרנט יכול לכלול אף תמונות כאשר כל תמונה אף לה יש כתובת URL משלה. כך ששם קובץ יכול להיות תמונה יותר מאשר איזשהו דף HTML ואף רק קובץ או תוכנה.

אם שם הקובץ להיות לא נתון בכתובת URL אז השרת אינטרנט לרוב מבצע אחד משלושת הדברים (בהתאם על איך להיות מוגדר התצורה שלו)

1. השרת מסתכל בקובץ ברירת מחדל ומחזיר את הקובץ שהוא בתור שם הקובץ בכתובת URL. עבור האתרים הנכתבים ב-HTML הקובץ להיות לרוב index.html אם אין נתיב לקובץ מפורט בכתובת URL אז השרת מסתכל בתיקיית root של הספרייה או אם הספרייה מפורטת בכתובת URL אז השרת מסתכל בקובץ index.html

2. השרת מציע רשימה של קבצים בספרייה המפורטת בכתובת URL.

3. השרת מציג הודעה שאומרת שהדף אינו נמצא או שלא ניתן לדפדף בקובץ שבתקייה.

כאשר אתם מתקשרים לדף אינטרנט אחר באתר שלכם אתם לא צריכים להשתמש בכל שלושת חלקים של URL. אתם יכולים להשתמש רק ב- filepath ושם הקובץ.

כתובת Absolute URL וכתובת Relative URL

Absolute URL מכיל כל דבר שאתם צריכים להשתמש בו על מנת לזהות באופן ייחודי קובץ מסוים על האינטרנט. וזה להיות מה שאתם מקלידים בתוך שורת הכתובת שבדפדפן שלכם על מנת למצוא את הדף אינטרנט המבוקש. לדוגמא כדי לקבל את דף האינטרנט אודות סרטים באתר חדשות דמיוני שהינו

<http://www.exmappleNewsSite.com/Entertainment/Film/index.html>

Relative URL מצביע כאשר המשאב להיות קשור לדף האינטרנט הנוכחי כשאתם נמצאים בו לדוגמא באתר הקודם כאשר מתקשרים לדף אחר באותו ספרייה זה להיות Relative URL ואף גם להשתמש ב- URL relative לציין קבצים בספריות שונות. כאשר אתם רוצים להוסיף לינק לדפי index ספריות שונות כגון הספריות השונות בכתובת האתר הדמיוני שצינו Film,TV,,Art,Music עם הסיימת /index.html לדף הבית של האתר וזה יותר מהיר מאשר לכתוב את הכתובת המליאה ש Relative URL ניתן לשימוש בכל דף באותו אתר.

הערה: שרתי אינטרנט לרוב יש רעיון מושגי של מסמך שיוצג כברירת המחדל. זה מציין שאיזשהו בקשה עבור ספרייה להיות מעוברת למסמך הברירת המחדל המוגדרת בספרייה. בהרבה שרתים זה מוביל לשם הקובץ index.html שזה בעצם דף הבית באותו ספרייה כאשר ייצור רשימה של מאמרים שאורכם קל לקריאה

אתם עשויים להתעניין לדעת מדוע לגבי הדבר שדפדפן האינטרנט עדיין מבקש את הכתובת URL המליאה ולא את כתובת URL הקצר (Relative URL) אבל באופן ממשי הדפדפן מבצע עבודה של הפניית ה- Relative URL לתוך כתובת Absolute URL המליאה.

יתרון אחר לשימוש ב- Relative URL בתוך האתר אינטרנט שלכם הינו כך כשאתם יכולים לפתח את האתר שלכם על המחשב השולחני או הנייד ללא קניית שם דומיין. ואף אתם יכולים לשנות את השם דומיין שלכם או להעתיק את החלקים המשניים של אתר אחד לשם דומיין חדש ללא צורך של שינוי של כל הלינקים מאחר ומדובר בלינקים שהם קשורים לדפים אחרים באותו אתר ולא לינקים לאתר אינטרנט חיצוני בפני עצמו.

הערה: Relative URL עובד רק על לינקים בתוך אותו אתר אינטרנט: ואינכם יכולים להשתמש בהם כלינקים לשמות דומיין אחרים

החלקים המשניים באתר אינטרנט מספקים סוגים שונים של relative URL שאתם יכולים להשתמש.

1. **אותו ספרייה.**
2. **ספרייה משנית**
3. **ספריית המוצא.**
4. **תיקיית root**

אותו ספרייה

כאשר אתם רוצים לקשר או להכליל משאב מאותו ספרייה אתם יכולים רק להשתמש בשם של הקובץ שהוא שמאותו ספרייה. לדוגמא אם אתם רוצים לקשר מדף הבית (index.html) לדף contactus.html אתם צריכים להשתמש בשם קובץ contactUs.html מאחר והקובץ שוכן בתוך אותו ספרייה וכך כשאתם לא צריכים לציין איזשהו דבר אחר.

ספרייה משנית

שמות הקבצים שצינו קודם בדוגמא לאתר החדשות הדמיוני: Art, Music, TV, film הם ספריות משניות של ספריית entertainment. כאשר אם כותבים את הדף בתוך ספריית entertainment הם יכולים לייצור לינק לדף index של הספריות המשניות שנראה כך:

Film/index.html
TV/index.html
Art/index.html
Music/index.html

אתם צריכים לכלול את השם של הספרייה המשנית ולאחריו בתו של לוכסן קדמי והשם של הדף שאתם רוצה לקשר אליו.

עבור כל ספרייה משנית נוספת רק צריך להוסיף את שם של הספרייה ולאחריו תו של לוכסן קדמי. כך שאם אתם יוצרים קישור מהדף שבתיקיית root של האתר כגון דף הבית הראשי של האתר, אתם משמשים Relative URL לדוגמא את הדפים הבאים:

Entertainment/Film/index.html
Entertainment/TV/index.html
Entertainment/Arts/index.html
Entertainment/Music/index.html

ספריית המוצא

אם אתם רוצים לייצור לינק מספרייה אחת לספריית המוצא שלה(שזהו הספרייה שהיא נמצאת בתוכו) משתמשים בסימון של שתי נקודות ולאחר מכן תו של לוסן קדמי /.. לדוגמא מדף בספריית Music לדף בספריית entertainment אנחנו משתמשים ב- Relative URL הנראה כמו זה `../index.html` . וכאשר אתם רוצים לקשר מספריית Music לדף בספריית ה- root אתם צריכים לעשות זאת שוב פעם אחת באופן הנראה כמו זה `../../index.html` .

תיקיית root

אתם יכולים להצביע על קובץ הקשור לתיקיית root של האתר. כך שאם אתם רוצים לקשר את הדף לדוגמא `contactUs.html` מאיזשהו דף בתוך האתר עליכם להשתמש באופן זה בנתיב שלו כאשר לוסן נמצא קודם לנתיב כלדוגמא אם הדף `ContactUs` הינו בתוך תיקיית root, אז אתם צריכים להכניס באופן זה `/ContactUs.html` .

לחלופין אתם יכולים לקשר את לחלק Music של הדף `index` מאיזשהו מקום בתוך אותו האתר בשימוש באופן הבא `Entertainment/Music/index.html` / כאשר הלוסן האחורי הנמצא בתחילה מצביע על ספריית root ואחר כך הנתיב מהמקום שזה להיות מצוין שם.

<base> על האלמנט

כפי שהזכרנו מוקדם יותר כאשר הדפדפן נפגש בכתובת relative URL הוא למעשה מעביר את כתובת relative URL לכתובת המלאה. האלמנט base מאפשר לכם לציין כתובת URL בסיסית עבור הדף שכל relative url המצוינות בדף יהיו מתווספות לכתובת URL הבסיסית כאשר הדפדפן נפגש relative URL .

כאשתם מציינים את הכתובת URL הבסיסית עושים זאת כערך של התכונה href באלמנט base לדוגמא: `<base href=http://www.exampleSite.com/>`

האלמנט base צריך להיות בתוך האלמנט head, ואף יכול להופיע פעם אחת בדף.

יצירת לינקים בתוך הדף עם אלמנט <a> ותכונת href

כמו כל המסעות ללינקים יש גם נקודת פתיחה הידועה כ- source (מוצא) ונקודת הסיום הידועה כ- destination (יעד) . ב- HTML שתי הנקודות הללו נקראות anchor (עוגנים). כל הלינקים שאתם רואים בדף כאשר אתם יכולים לחוץ עליו הינו source anchor (עוגן המוצא) שנוצר באמצעות האלמנט a. אתם יכולים להשתמש באלמנט a ליצור סימון בחלקים של דפים שלכם שמאפשרים לכם ליצור לינק ישירות לחלק של הדף בתוך אותו דף. הסימונים הללו נקראים destination anchor (יעד).

יצירת source anchor עם התכונה href

ה - source anchor (מוצא) הינו מה שמרבית אנשים חושבים כאשר הם משוחחים אודות הלינקים באינטרנט בין שהלינק מכיל טקסט או תמונה. שזה להיות דבר כאשר אתם יכולים ללחוץ ושנו צפייה שייקח אותכם למקום אחר.

כפי שראיתם מקודם איזשהו טקסט המוכל בין התג פותח a ותג הסוגר a כאשר יוצר את חלק של הלינק כאשר המשתמש יכול ללחוץ. הכתובת URL שהמשתמש צריך להילקח בלינק הינו מצוין כערך בתכונת href.

כלדוגמא כאשר אתם לוחצים על המילים exmapleNewsSite אז הלינק מעביר אותכם לכתובת <http://www.exmapleNewsSite.com>.

```
<a href="http://www.exmapleNewsSite.com">exmapleNewsSite</a>
```

ואם הלינק ממוקם בתוך הדף הבית של האתר כלדוגמא באתר חדשות הנ"ל אז הלינק ייקח אותכם לדף האחר המצוין בלינק.

```
<a href="Entertainment/Film/index.html">Film Section</a>
```

אתם צריכים לציין destination anchor (יעד) רק אם אתם רוצים לעשות לינק לחלק אחר באותו דף.

יצירת לינק לחלק אחר בדף בשימוש התכונות id ו- name

אם יש לכם דף אינטרנט ארוך אתם עשויים לרצות לקשר לחלק ספציפי של הדף על מנת לחסוך למשתמש את הצורך לגלול למעלה או למטה את הדף על מנת למצוא חלק רלוונטי שהוא רוצה לראות. ה- destination anchor (יעד) מאפשר למחבר הדף לסמן נקודות ספציפיות בתוך הדף ש- source anchor (מוצא) יכולות להצביע לחלק המסוים הזה בדף.

דוגמאות נפוצות לשימוש בלינקים לחלקים ספציפיים של הדף שאתם עשויים לראות שנעשה בהם שימוש בדף האינטרנט כוללים:

1. חזרה לראש הדף הנמצא בתחתית דף ארוך.
2. רשימה של תוכן בדף שלוקחת את המשתמש לחלק הרלוונטי של אותו דף המצוין בתוכן.
3. לינקים בתוך הטקסט להערת שוליים או להגדרה.

כשאתם יוצרים destination anchor (יעד) שוב בשימוש האלמנט a , אבל כאשר הלינק מתנהג כ- destination anchor (יעד) אתם משתמשים בתכונה id בשונם כאשר אתם משתמשים רק בתכונה href.

אם אתם מסתכלים בקוד המקור של דפי אינטרנט ישנים יותר דהיינו שמשתמשים בגירסאות של HTML ישנות יותר, אתם עשויים לראות תכונה name שגם בה נעשה שימוש, או אפילו עשו בה שימוש במקום id בלינקים לחלק אחר באותו דף. נזכיר בקצרה

שהתכונות id ו- name הם תכונות המשתייכות לקטגוריית התכונות הגלובליות כאשר יכולים להתבצע בכל האלמנטים. התכונת id היא הדרך המועדפת ליצור destination anchor (יעד) כעת אבל זה הוצג אך ורק מהגרסה 4 של HTML ובמקום זה השתמשו בתכונה name שביצעה אותם פעולות בדיוק כמו התכונה id ביצירת destination anchor בגרסאות הקודמות של HTML.

בדרך של דוגמא נניח שיש לכם דף אינטרנט ארוך עם כותרת ראשית וכותרות משניות אחדות, כאשר כל הדף אינו מתאים לתצוגה במסך בפעם אחת, ועל כן המשתמש מוכרח לגלול את הדף כך כאשר אתם רוצים לחסוך מן המשתמש את הצורך לגלול אתם מוסיפים לינקים בראש העמוד שלוקח את הקוראים ישירות לכל אחד מן החלקים המודגשים בדף ההוא.

לפני שאתם יוצרים לינקים לכל אחד מן החלקים של הדף (בשימוש source anchor) אתם צריכים להוסיף destination anchors. פה אתם יכולים לראות שבתוך האלמנטים h2 שהינם אלמנטים של כותרות משניות ישנה תכונה של id כאשר בתכונה זו יש ערך המזהה כל חלק. יש לזכור כאשר לגבי הערך של תכונה id שאף פעם לא יכול להכיל שני ערכים אותו דבר באותו דף.

```
<h1>Linking and Navigation</h1>
```

```
<h2 id="URL">URLs</h2>
```

```
<h2 id="SourceAnchors">Source Anchors</h2>
```

```
<h2 id="DestinationAnchors">Destination Anchors</h2>
```

```
<h2 id="Examples">Examples</h2>
```

עם ה- destination anchors (יעד) במקומם, כעת אתם יכולים להוסיף source anchors (מוצא) לקשר לחלקים הללו בתוך הדף שמצוינים כ- destination anchors. על מנת לקשר לחלק מסוים בדף אז הערך של התכונה href בתוך ה- source anchors (מוצא) צריך להיות אותו דבר כערך של התכונה id ב destination anchor (יעד) התואם איליו מקשרים כאשר לפני הערך צריך להיות סולמית #. כלדוגמא:

```
<nav>
```

```
<p>This page covers the following topics:</p>
```

```
<ul>
```

```
<li><a href="#URL">URLs</a></li>
```

```
<li><a href="#SourceAnchors">Source Anchors</a></li>
```

```
<li><a href="#DestinationAnchors">Destination Anchors</a></li>
```

```
<li><a href="#Examples">Examples</a></li>
```

```
</ul>
```

```
</nav>
```

אם אנשים רוצים לקשר לחלק ספציפי בתוך דף אינטרנט מאתר אינטרנט אחר הם צריכים להוסיף את הכתובת URL המליאה עבור הדף ולאחר מכן עם סולמית ולאחר מכן אות ערך של התכונה id בדיוק כפי שנמצא באותו דף אינטרנט באתר אחר.

תכונות נוספות של האלמנט <a>

האלמנט a מבצע תכונות נוספות שעדיין לא נפגשו בהם.
האלמנט a תומך בכל התכונות הגלובליות והתכונות הבאות: accesskey ו- href ו- hreflang ו- rel ו- style ו- tabindex ו- target ו- type.

הגבר accesskey attribute

התכונה **accesskey** יוצרת קיצורי מקלדת שמאפשרים למשתמשים להפעיל את הלינק. לדוגמא אם אתם נותנים בתכונה accesskey ערך של t כאשר המשתמשים מקישים על הקליד t יחד עם המקש Alt או Ctrl (בהתאם למערכת הפעלה) ואז מפעיל את הלינק.

בדפדפנים מסוימים כגון-chrome כאשר הלינק מפעיל את הדפדפן באופן מיידי לאחר הפעלת הלינק במקשי הקיצור לקישור. ובדפדפנים מסוימים אחרים כגון IE הלינקים הינם רק מוצגים באופן מודגש והמשתמשים צריכים ללחוץ על המקש Enter לאחר לחיצת מקשי הקיצור עבור שהדפדפן יתחבר ללינק .

התכונה **accesskey** צריכה להיות מצוינת ב- source anchor (מוצא). לדוגמא אם אתם רוצים קישור לחלק שבראש הדף כאשר המשתמש לוחץ על המקש T בלוח מקשים שלו (יחד עם Alt או Ctrl) אתם צריכים להשתמש בתכונה **accesskey** באופן זה

```
<a id="bottom" accesskey="t">Back to top</a>
```

הדפדפן	מערכת הפעלה	מקש צירוף
IE	Windows	Alt+ accesskey ולאחר מכן נדרש ללחוץ Enter
	לא נתמך ב- linux	
Chrome	Windows	Alt+ accesskey
	Linux	Alt+ accesskey
Firefox	Windows	Alt+Shift+ accesskey

Alt+Shift+ accesskey	Linux	
----------------------	-------	--

הסבר hreflang attribute

התכונה hreflang מצביעה באיזו שפה הדף שאיליו אתם מקשרים נכתב. התכונה מעוצבת לשימוש כאשר ישנם קישורים לדפים שכתובים בשפות שונות מהמסמך הנוכחי, הערך של התכונה הינו שתי האותיות המשמשים כקוד השפה כלדוגמא:

```
<a href="http://www.amazon.co.jp/" hreflang="JA">Amazon Japan</a>
```

בדוגמא זו אנו רואים שימוש בקוד שפה של יפנית לערך של התכונה hreflang .

בתכונה hreflang ניתן לעשות שימוש רק אם מוגדר באלמנט a התכונה href .

יש להשתמש בתכונה זו כאשר אתה יוצרלינק לדף אינטרנט שהינו תרגום מלא של הדף שבו הלינק.

הסבר target attribute

התכונה target מפרטת איפה לפתוח את המסמך המקושר.

במצב ברירת מחדל כאשר אנו משתמשים באלמנט a ליצור לינק, המסמך שאתם מקשרים איליו נפתח באותו חלון בדפדפן, כאשר אתם רוצים שהדפדפן יפתח חלון חדש כשהקישור יופעל עליכם להשתמש בתכונה target עם הערך _blank וכך זה יראה לדוגמא:

```
<a href="Page2.html" target="_blank">Page 2</a>
```

פרק התמונות

תוכן:

- 1.בחירת תמונות עבור הדף אינטרנט שלך.
- 2.דרכים להוספת תמונות לדף אינטרנט.
- 3.הוספת תמונות בשימוש האלמנט
- 4.איפה למקם את התמונות בתוך הקוד שלכם.
5. אלמנט כ- replaced element .
6. שימוש התמונות כלינקים.
7. בחירת פורמט הנכון של תמונה.

8. פורמט לתמונה GIF.

9 אופן שימוש בתמונות ממספר משאבים

10. הסבר אלמנט <figure>.

בחירת תמונות עבור האתר שלך

תמונה שווה אלף מילים, ותמונות גדולות מסייעות ליצור הבדלים תדמיתיים בין אתר הנראה כאתר ממוצע לאתר אחד שנראה ממש מרתק.

בתמונות ניתן להשתמש בהן כדי להגדיר את הטון עבור האתר בפחות זמן מאשר זה לוקח לקרוא את התיאור. אם אין לכם תצלומים להשתמש בזה באתר אינטרנט שלכם ישנו לכך פתרון כאשר ישנם חברות כאשר מוכרות stock images (מלאי של תמונות) כאשר לגבי אותם תמונות אתם צריכים לשלם על מנת להשתמש בהם. זכרו שכל התמונות כפופות לזכויות יוצרים, ואתם יכולים לביא עליכם צרות משפטיות בגין לקיחת תצלומים פשוטים מן אתר אינטרנט אחר.

אם יש לכם דפים שמציגים תמונות אחדות (כגון תצלומים של מוצר או תצלומים של חברי צוות) אז אתם שמים אותם ברקע פשוט שכולו אחיד כאשר עוזר למשתמשים לראות את אותם באופן טוב יותר כקבוצה.

התמונות צריכות להיות:

1. רלוונטיות לגבי התוכן הדף שבהם הם מוצגים.
2. משמשות להעברת מידע.
3. צריכות להעביר הלך חשיבה נכון.
4. להיות מותאם לצבע התבנית.

דרכים להוספת תמונות לדף אינטרנט

- ישנם 4 דרכים להוספת תמונות לדף אינטרנט
1. **external link** (קישור חיצוני).
 2. **embedded images** (הטמעת תמונות).
 3. **background images** (תמונות רקע).
 4. **custom bullets** (תבליטים מותאמים אישית)

external link

מדובר כאשר בדף אינטרנט יש טקסט שמוטמע בתוכו לינק. כאשר המשתמש מקליק על הלינק, הדף אינטרנט מוחלף בתמונה. על מנת ליצור תמונה שתוצג ע"י לינק יש לעשות זאת באמצעות האלמנט a שמכיל קישור לתמונה.
לדוגמא:

Susan B. Constant

בדוגמא אנחנו רואים שהתמונה שוכנת כלינק רק ע"י האלמנט a שמכיל את שם קובץ התמונה עם סיומת שבו מצוין פורמט התמונה אך מומלץ שהתמונה תהיה ממוקמת בתוך השרת ווב שלכם ולא להסתמך מקישור מאתר אחר מאחר כאשר בעלים של אותו אתר יחליט להסיר את אותו תמונה זה יגרום שהלינק שלכם יהיה לינק שבור.

embedded images

זהו אופן כאשר התמונה מוטמעת לתוך הדף אינטרנט עצמו וזה נעשה באמצעות האלמנט .img

לדוגמא:

background images

זה אופן שהתמונה משמשת כרקע עבור הדף אינטרנט בכללותו או שבחלק מסוים בדף אינטרנט. בשביל זה נדרש מניפולציות מיוחדות על מנת ליצור אותם כמתאימים לשימוש כרקע.

custom bullets

זה מתייחס לתמונה קטנה שמכניסים אותה בתור תבליט לרשימה ממוספרת או רשימה לא ממוספרת כאשר זה מתבצע יחד עם CSS .

** הוספת תמונות בשימוש האלמנט**

התמונות נוספות לאתר אינטרנט בשימוש האלמנט img כאשר צריך לשאת בתוכו לפחות שני תכונות שהן src שהינה תכונה המצביעה על מקור התמונה, ותכונה שנייה alt שמספקת תיאור של התמונה.

לדוגמא בשורה הבאה אנו מוסיפים תמונה הנקראת logo.gif לתוך הדף.

האלמנט img תומך בכל תכונות הגלובליות ואף יכול לשאת בתוכו את התכונות שלו שהן src – alt – width – height – usemap – ismap – crossorigin .

האלמנט img הינו אלמנט ריק כאשר מכיל רק תג אחד

הסבר על תכונת src

התכונה src מספרת לדפדפן איפה ניתן למצוא את התמונה. הערך של התכונה הינו כתובת URL בדיוק כמו לינקים שדיברנו עליהם בפרק קודם הכתובת URL יכולה להיות Absolute URL או Relative URL.

באופן כללי כאשר מדברים על תמונות עבור האתר שלכם זה צריך תמיד להיות מאוחסן בשרת שלכם או בשרת אחר שיש לכם עליו שליטה. זה לא מנהג טוב לקשר תמונות שנמצאים על אתרי אינטרנט אחרים בגלל שאם הבעלים של אתרי אינטרנט אחר מחליט להעביר את התמונה מאותו מקום שאיליו קישרתם אז כתוצאה מכך המשתמשים שלכם כבר לא יראו את התמונה.

בגלל שהתמונות נמצאות על שרת אינטרנט שלכם כתוצאה מכך הכתובת URL קרוב יותר לוודאי שהינה Relative URL יותר מאשר Absolute URL.

מרבית מחברי דפי האינטרנט יוצרים ספרייה נפרדת(או תיקייה) בתוך האתר האינטרנט עבור תמונות. אם יש לכם אתר אינטרנט גדול אתם עשויים אפילו ליצור תיקיות שונות עבור סוגים שונים של תמונות. כלדוגמא אתם עשויים לשמור איזשהו תמונות שמשמשים עבור העיצוב של הממשק (כגון logos ו- button) בנפרד מהתמונות שעבור שימוש בתוך התוכן של האתר או תמונות של חדשות יהיו בתיקיית news וכך תמונות מוצרים יהיו בתיקיית product.

כאשר דף אינטרנט נטען באותו רגע הדפדפן מקבל את התמונה משרת ווב ומכניס אותו לתוך הדף. לכן יש ליצור וודאות שהתמונה שוהה באופן ממשי באותו מיקום בהקשר לדף האינטרנט, אחרת המבקרים שלכם באתר יקבלו אייקון של לינק שבור. האייקון של לינק שבור מוצג אם הדפדפן אינו יכול למצוא את התמונה.

הסבר על תכונת alt

התכונה alt צריכה להופיע בכל אלמנט של img וערך של התכונה צריך להיות תיאור טקסטואלי של התמונה.

חשיבות השימוש בתכונה alt על מנת לתאר באופן טקסטואלי את התמונה הינו מהסיבות הבאות:

1. כאשר אם הדפדפן אינו יכול להציג את התמונה אז הטקסט שבתכונה משמש כאלטרנטיבה לתצוגה במקום התמונה.
2. מסייע לנגישות האתר במקרה של כבדי ראייה כאשר התוכנה screen reader כאשר היא מגיעה לתמונה בדף אינטרנט היא אינה יכולה לראות את התמונה אלא משתמשת בתכונה alt על מנת לתאר את התמונה.
3. למרות שמנועי חיפוש הם חכמים הם אינם יכולים לתאר או למפתח את התוכן של תמונה עקב כך מספקים חלופה של טקסט כאשר מסייעת למנועי חיפוש למפתח את הדפי האינטרנט שלכם וכך לסייע למבקרים למצוא את האתר שלכם.

לפעמים תמונות אינם מספקות מידע כלשהוא והם משמשים רק לשיפור מתווה של הדף (לדוגמא התמונה עשויה להיות רק אלמנט דקורטיבי ולא מוסיף איזשהו מידע לדף) במקרה הזה המובא בדוגמא התכונה alt צריכה עדיין לעשות בו שימוש אבל שלא נתון בה איזשהו ערך. כלדוגמא

השימוש בתכונת title

ניתן לעשות גם בתכונת title עם האלמנט img על מנת לספק מידע נוסף אודות התמונה. מרבית הדפדפנים יציגו את התוכן של התכונה ב- tooltip כאשר המשתמש ירחף עם הסמל על גבי התמונה.

הסבר על תכונות width ו- height

התכונות width ו- height מפרטות את האורך והרוחב של התמונה. כאשר התכונה height מפרטת את אורכה של התמונה בפיסקלים. והתכונה width מפרטת את רוחבה של התמונה בפיסקלים.

דוגמא לשימוש בתכונות הללו

```

```

באופן טכני הערכים של התכונות הללו יכולים להיות אחוזים מהמסך של הדפדפן. או שהתמונות הינם בתוך אלמנט שמקבל רק חלק של הדף המוכר כ- containing element (אלמנט הכלה) והערכים יהיו האחוזים של גודל containing element (אלמנט הכלה). אם אתם עושים שימוש באחוז המספר יהיה מוצג עם סימן אחוז לאחריו, אבל עבור מרבית דפי האינטרנט זה נדיר לעשות שינויים בערכים של גודל התמונה מאחר והצגת תמונה כאשר שונה הגודל לאיזשהו גודל אחר שונה מן הגודל שבו התמונה נוצרה בהתחלה יכול לגרום לתמונה מעוותת ומטושטשת.

ציון הגודל של התמונה הינו מנהג טוב ושקול כשאתם לכן אתם צריכים לנסות להשתמש בתכונות הללו על איזשהו תמונה שאתם שמים בדף שלכם.

שימוש בתכונות הללו גם יכולות לסייע לטעינת הדף באופן מהיר יותר ואף באופן חלק יותר כאשר תמונות לעיתים קרובות זמן טעינתם ארוך יותר מאשר הקוד HTML שיוצר את שאר הדף. כך בעצם ציון גודל התמונה בשימוש התכונות הללו מסייע לדפדפן שיכול להגיש את השאר הטקסט שבדף באופן תקין בעוד שהוא משאיר את הכמות הנכונה של מרחב עבור התמונה שעדיין נטענת ע"י הגודל המצוין של התמונה בתכונות הללו.

גם אם עבור איזשהו סיבה התמונה לא נטענת הדפדפן יצייר תיבה ריקה באורך והרוחב הנתונים עם הטקסט של alt בתוכו ע"י שימוש בתכונות הללו.

למרות שאתם יכולים להורות לדפדפן להציג תמונות קטנות יותר או גדולות יותר מאשר גודלם האמיתי (ע"י הוראה לדפדפן שהרוחב והאורך יהיו שונים מגודלם האמיתי) אתם צריכים להימנע מלעשות זאת בגלל שהתמונה השלכם תהיה לא ברורה. אלא אתם צריכים לשאוף ליצור גרסאות של תמונות באותו גודל שאתם תשמשו איתם בדפי אינטרנט שלכם. תוכנות כ- PhotoShop Elements, PhotoShop Pro, Paint Shop Pro, או GIMP יכול לעזור לכם לעשות זאת.

זה גם חשוב לא להשתמש בתמונות גדולות יותר מאשר הגודל שניתן לראות במסך(לדוגמא אתם צריכים לא להשתמש בתמונה בגודל 800 פיקסלים על 800 פיקסלים כי אם אתם תראו את התמונה זה יראה רק בגודל של 100 פיקסלים על 100 פיקסלים שניתן לראות

במסך) בגלל שתמונה בגודל קטן יותר הגודל של הקובץ הינו קטן יותר מאשר התמונה הגדולה יותר (במושגים של קילובייט). וכאשר גודל הקובץ של התמונה קטן יותר התמונה נטענת מהר יותר בתוך הדפדפן. גם להשמת תמונות קטנות יותר בגודל מתמונות הגדולות באתר אינטרנט שלכם עבור אחרים שיראו זאת באתר, זה עשוי לחסוך לכם כסף בגלל שאתם מחויבים לעיתים קרובות ביחס לגודל הכולל של כל הקבצים שאתם שולחים לאנשים כאשר מבקרים באתר שלכם כאשר תמונות וש"כ כשהתמונות גדולות ואז זה תופס שטח אחסון גדול לקבצים כאשר לעיתים נדרש עוד שטח אחסון גדול לכלל הקבצים מחמת סיבה זו.

כמו כן זה חשוב לא להראות תמונות שהן גדולות יותר מאשר שהן באופן אמיתי. אם יש לכם תמונה קטנה (נאמר כלדוגמא 100 פיקסלים) ולנסות להציג את זה הרבה יותר גדול (נאמר כלדוגמא 300 פיקסלים) אז התמונה תופיע מחוספסת.

למרות זה לא רעיון טוב לעשות כך כאשר אתם מציינים רק את התכונה width או רק את התכונה height, הדפדפן שלכם יציג את התמונה עפ"י קנה מידה. נניח למשך רגע שיש לכם תמונה ברוחב של 200 פיקסלים על גובה 100 פיקסלים. אם אתם רק מציינים את הרוחב של התמונה כ- 200 פיקסלים אזי הדפדפן ינסה להציגו בגודל התקין של רוחב 200 פיקסלים על גבי גובה 100 פיקסלים. אולם אם אתם אמרתם שהתמונה הייתה ברוחב כ- 100 פיקסלים ולא ציינתם את האורך, הדפדפן ינסה ליצור שגובה התמונה יהיה כ- 50 פיקסלים. בגלל שהתמונה הינה 50 אחוז מרוחב התמונה המקורית על כן הדפדפן יציג את התמונה ב-50 אחוז של אורך המקורי של התמונה. במילים אחרות זה מקיים את Aspect ratio של תמונה כאשר הרוחב של התמונה מתחלק באורך התמונה.

אתם יכולים אפילו לעוות תמונות באמצעות קביעת רוחב שונה ביחס לאורך.

אמנם שימוש בגודל התמונה נעשה ב-CSS יותר מאשר ב-HTML

איפה למקם את התמונות בתוך הקוד שלכם

איפה התמונה ממוקמת בתוך הקוד ישפיע על איך התמונה תהיה מוצגת.

פה אנו נציג שלוש דוגמאות של מיקום תמונה שמייצר תוצאות שונות

1. לפני הפסקה: כאשר הפסקה מתחילה בשורה חדשה לאחר התמונה.
2. בתוך תחילת הפסקה: השורה הראשונה של הטקסט מסתדרת בשורה עם תחתית התמונה.
3. באמצע הפסקה: התמונה ממוקמת בין המילים של הפסקה שבה היא מופיעה.

אלמנט `img` הינו מוגדר כאלמנט מקטגוריית `inline element` וזה שכפי רואים במופרש בשני הדוגמאות שצינו בתחילת הפסקה ובאמצע הפסקה כאשר האלמנט `img` בתוך הטקסט של האלמנט `p` כשאר האלמנטים `inline level` ואף בדוגמא הראשונה כאשר התמונה לפני הפסקה מדובר כאשר האלמנט `img` פועל בתוך האלמנט `div` שהינו אלמנט מקטגוריית `block level`.

אלמנט כ- replaced element

האלמנט img הינו אלמנט המשתייך לאלמנטים מקטגוריית replaced element.

Replaced element הינם אלמנטים שהתוכן שלהם מסופק עי משאבים אחרים. כך אלמנט img כאשר התוכן שלו שהינם התמונות המוטמעות בדף אינטרנט בשימוש התכונה src כאשר מבצעת את הטמעת התמונות מאותם משאבים אחרים ואף ישנו אורך ורוחב המוגדרים אף ללא CSS.

שימוש התמונות כלינקים

זה קל מאוד להפנות תמונה לתוך לינק. יותר השמת טקסט בין התג הפותח של אלמנט <a> לתג הסוגר של אלמנט כאשר אתם פשוט שמים תמונה בתוך התגים הללו. כך תמונות משמשות כלינק או כפתור גרפי להעברה לדף אחר לדוגמא:

```
<a href="www.israelisraeli.com"
 </a>
```

בדוגמא זו של קוד אנו רואים שתוך האלמנט a השתמשנו במקום בטקסט השתמשנו בתמונה שמשמשת כלינק.

יש לציין שכדאי על מנת להגביר את חוויית המשתמש באתר כאשר משתמשים בתמונה כלינק יש לשים תמונה משייצגת דבר שנראה בברור שהוא משמש כלינק כלדוגמא מסעדה צרפתית כלשהי כאשר משתמשים בתמונה של לוגו של אותה מסעדה ואף בתכונה title שתספק מידע למשתמש על אודות הדבר איליו התמונה מקשרת ב- tooltip שיוצב ע"י ריחוף סמן של עכבר על גבי התמונה זו.

בחירת פורמט הנכון של תמונה

תמונות והגרפיקה יכולות להביא לאתר שלכם חיות. אבל אתם מוכרחים ללמוד איך להכין תמונות לאתר אינטרנט שלכם. שאם לא כן הם יכולים להגדיל משמעותית את הזמן שזה לוקח לדף להיטען.

כאשר כותבים אתרים במחשב השולחני או הנייד, אתם עשויים לא לתפוס כמה זמן דף ייקח לו להיטען, מאחר שקבצים שיושבים במחשב שלכם ייטען הרבה יותר מהר מאשר אם הם יהיו באינטרנט. עקב כך בחירת הפורמט התמונה הנכון ושמירת התמונות שלכם באופן תקין

יכולה להבטיח שאתר שלכם לא ייטען בטעינה איטית ומיותרת והתוצאה תהיה מבקרים מאושרים יותר מן האתר שלכם.

הערה: pixel או בשמו האחר picture element הינו ריבוע זעיר המייצג יחידת מידע קטנה ביותר ומוצג כנקודה או כריבוע בתוך רשת כאשר עם פיקסלים נוספים יחדיו ברשת הם יוצרים תמונה באמצעים הדיגיטליים.

מרבית התמונות הסטטיות באינטרנט מסווגות כ-bitmapped images (תמונות שמתוארות באמצעות מפת סיביות). Bitmapped images מחלקות תמונה לתוך רשת של פיקסלים ומציינות צבע של כל פיקסל באופן פרטני. אם אתם מסתכלים מקרוב על המסך מחשב שלכם, אתם עשויים לראות את הפיקסלים שיוצרים את התמונה במסך שהינם הריבועים כאשר נראים שמגדילים את התמונה בגודל שרואים את הרבועים הזעירים. ישנם פורמטים אחדים ושונים של bitmapped שהנפוצים ביותר בהם הם GIF, JPEG, TIFF, PNG, BMP שמכונה גם bitmap בשמו המלא.

המספר של פיקסלים בכל רבוע אינטש של תמונה הינו מתייחס לרזולוציה של התמונה. באופן נורמלי שומרים את התמונות שיהיו בשימוש באינטרנט ברזולוציה של 72 פיקסלים לכל אינטש בגלל שזה מתאים עם מספר של פיקסלים לכל ריבוע אינטש במסך מחשב שלכם. לעומת זאת תמונות שהם שימוש להדפסה לרוב הם מסופקות למדפסות ב- 300 נקודות לכל אינטש (או 300 פיקסלים לכל אינטש).

המספר יותר של פיקסלים או נקודות לכל אינטש תמונה מכיל גודל גדול יותר של הקובץ ב-(בקילובייט), כאשר אז לוקח זמן ארוך יותר להעביר את התמונות על גבי האינטרנט. עקב כך כל תמונה כלשהי שאתם משתמשים באינטרנט אתם צריכים לשמור ברזולוציה של 72 נקודות או פיקסלים לכל אינטש.

הערה: למרות שבקלות אתם יכולים לשמור תמונה שהינה 300 פיקסלים לכל אינטש ב- 72 פיקסלים לכל אינטש עבור שימוש בזה באינטרנט, אך בתמונה שהיא 72 פיקסלים לכל אינטש אינכם יכולים פשוט רק להגדיל את התמונה מ 72 פיקסלים לכל אינטש ל- 300 פיקסלים לכל אינטש בגלל שאתם לא יודעים מה הצבע 228 שחסר בו הפיקסלים שצריכים להיות עבור השלמת הצבע. ואם אתם רק מנסים להגדיל את הרזולוציה של התמונה זה לעיתים קרובות זה יראה כמחוספס. עקב כך כאשר יש לכם רזולוציה גבוהה של כ- 300 פיקסלים לכל אינטש בתמונה, אתם צריכים ליצור העתק של התמונה עבור שימוש באינטרנט ולשמור את הגירסה המקורית בנפרד.

פורמט לתמונה GIF

פורמט GIF הינו ראשי תיבות של Graphics Interchange Format (פורמט המרת גרפיקה) של תמונות נוצר בשימוש ב-palette (לוח צבעים) עד לכמות של 256 צבעים וכאשר כל פיקסל של התמונה הינו אחד מן 256 צבעים הללו. כל תמונה שונה בפורמט GIF יכולה להיות palette שונה של כ- 256 צבעים שנבחרו מטווח של מספר צבעים שמונה

יותר מ 16 מיליון צבעים. וזה נעשה ע"י התוכנה ששומרת את התמונה שגם בוחרת את ה-palette שמציגה הכי טוב את התמונה.

הקובץ GIF מאחסן את ה-palette של הצבעים בתוך color lookup table (טבלת חיפוש) וכל פיקסל מפנה את מידע על הצבע למידע ששוכן בתוך color lookup table לא שהפיקסל עצמו יציין מידע לגבי הצבע שהוא מייצג. היתרון של טכניקה זו שאם הרבה פיקסלים משתמשים באותם צבעים התמונה לא תחזור על אותו מידע של צבעים וכך תוצאה מכך גודל הקובץ יהיה קטן יותר מאשר אם יחזור על אותו מידע של אותם הצבעים שהפיקסלים עושים בהם שימוש כאשר אם הפיקסלים בעצמם היו מציגים את המידע של הצבעים שהם משתמשים. זה יוצר שתמונות בפורמט GIF יותר מתאימות לשימוש בגרפיקה(ששם ישנם לעיתים קרובות שטחים שהם באותו צבע המוכר לנו כ-flat color) ופחות מתאים לתצלומים(ששם ישנם לעיתים קרובות הרבה צבעים שונים).

הדרך לאחסן את תמונות זה ב-indexed color format.

אם GIF מכיל פחות מאשר 16 צבעים אך מכיל מספר צבעים שגדול מן 8 צבעים (ובמקרה זה, התייחסות איליו היא כ- 4 bit GIF), גודל הקובץ של התמונה יהיה פחות ממחצית גודל המקסימלי הקובץ של התמונה הפורמט GIF המשתמשת ב-256 צבעים (8 bit GIF).

הסבר אלמנט <figure>

פרק וידאו

תוכן:

1. הוספת וידאו באמצעות האלמנט <video>
2. הסבר על התכונות של אלמנט <video>
3. הסבר על האלמנט <source> ותכונותיו בוידאו
4. הסבר על אלמנט <track>

הוספת וידאו באמצעות האלמנט <video>

הדרך החדשה ביותר להטמעת וידאו לדף אינטרנט ואף הינו סטנדרט בכלל להטמעת וידאו הינו באמצעות האלמנט video שהינו אלמנט חדש שנוסף ב-HTML 5 כך שאינו תומך בגירסאות קודמות של HTML וייתמך רק ע"י הדפדפנים החדשים שבהם החל תמיכה ב-HTML 5 כאשר ב-IE כלדוגמא זה תומך החל מ-IE9 ומעלה.

ההוספה עם האלמנט video צריך שיכיל בתוכו לפחות את התכונה src המציינת את הכתובת URL שבה שוכן אותו סרט וידאו ואף מומלץ שגם התכונות width ו-height שיציגו את גובה ורוחב של חלון הנגן של הסרט שיוצג בדף אינטרנט בפיקסלים.

כלדוגמא:

```
<video width="720" height="480" src="central.mp4">
```

בדוגמא אנחנו רואים שימוש באלמנט video עם התכונות width ו-height שמציינות את רוחבו וגובהו של החלון וידאו שיוצג בדף אינטרנט ותכונת src המפרטת את הכתובת URL שהסרט וידאו שוכן בו כאשר כולל שם הקובץ של הסרט ופורמט הסרט שפה בדוגמא הינו פורמט של MP4.

האלמנט video במצב ברירת מחדל יוצר מעצמו נגן וידאו

היתרון של אלמנט video

היתרון של האלמנט video הינו בכך שבעצם מאפשר לנו הטמעה ישירות לדף אינטרנט של תוכן וידאו ובכך יצר סטנדרט של הוספת וידאו לדף אינטרנט כמו הוספת תמונות הנוסף בסטנדרט של אלמנט img ולא כפי שהיה לפני עידן HTML 5 שלא היה סטנדרט להטמעת תוכן וידאו אלא היו משתמשים באמצעות ח-plug-in (תוסף) חיצוני כגון flash כדי להטמיע תוכן ואף ללא צורך בשימוש בג'אווה סקריפט ובכך האלמנט video יצר סטנדרט שמונע את מורכבות המיותרת כאשר למימוש התכונה זו של תוכן וידאו בקוד מינימאלי.

הסבר על התכונות של אלמנט <video>

הסבר על התכונה src

התכונה src מפרטת את כתובת URL שבו תוכן הוידאו שוכן שזה יכול להיות בשני אופנים URL relative כאשר זה שוכן באותו שרת או URL absolute כאשר מתייחס שהקובץ וידאו שוכן במקום אחר שהינו שרת אינטרנט שבו שוכן תוכן וידאו.
לדוגמא:

```
<video src="http://www.example.com/movie.ogg"> </video>
```

```
<video src="movie.ogg"> </video>
```

בדוגמא אנו רואים בתחילה URL absolute ולאחר מכן בשורה מתחתיו URL relative

בתכונה אף כולל כאשר שם הקובץ וידאו נכתב עם הפורמט שלו כאשר בדוגמא הפורמט וידאו הינו ogg .

הסבר על התכונות width ו-height

התכונות width ו-height מציינים את גודלו של החלון של הנגן כאשר התכונה width מציינת את רוחב החלון של הנגן וידאו והתכונה height מציינת את אורך החלון של הנגן וידאו.

הערך של התכונות הללו הינו פיקסלים אך לא ניתן לציין כמו ב-CSS בסיומת px אלא מספיק רק בערך מספרי שמציין שזהו מספר הפיקסלים של הגודל ביחס לרוב או לאורך.

חשוב לציין את התכונות הללו באלמנט video על מנת שהדפדפן ידע מראש בזמן הטעינה לפרוס שטח בגודל המצוין של נגן הוידאו בחלון הדפדפן וכך ייצור טעינה מהירה יותר של הדף אינטרנט. לולא כן טעינת הדף אינטרנט תארך זמן רב יותר מחמת כאשר הדפדפן אינו יודע את גודל החלון של הנגן וידאו המתווה הדף משתנה בזמן הטעינה עד שיוצג החלון הוידאו.

הסבר על התכונה controls

התכונה controls מגדירה כאשר פקדי וידאו יוצגו הפקדים שיוצגו הינם:

1. פקד play (לנגן) שהינו הפקד המאפשר לנגן את תוכן הוידאו.
2. פקד pause (להפסיק) הינו פקד המאפשר לעצור תוכן הוידאו המתנגן על מנת להפסיק לגמרי או להפוגה זמנית את התוכן וידאו המתנגן.
3. פקד seeking (חיפוש) הפקד כאשר מציג את הסרגל שבו רואים הטווח הזמן כשכעת התוכן המתנגן אוחד וניתן להזיז אותו לחלק ספציפי שרוצים לראות באותו תוכן וידאו.
4. פקד volume (עוצמת הקול) הפקד המציג את עוצמת הקול של תוכן הוידאו המתנגן בסרגל שניתן להזיז את הפקד על גבי הסרגל זה על מנת לווסת את עוצמת הקול.
5. פקד Fullscreen toggle (החלפת מצב למסך מלא) הפקד זה מאפשר שהחלון הנגן יעבור לתצוגה במצב של מסך מלא או להחזרו למצב קודם שאינו במסך מלא.

אלו פקדים שיוצגו במצב ברירת מחדל ע"י התכונה controls. אך ישנם עוד שני פקדים שאינם מוצגים במצב ברירת מחדל ויוצגו באופן כאשר הם זמינים בנגן שהינם:

1. פקד caption/subtitle (כתוביות) זה פקד המאפשר תצוגת כתוביות על גבי התוכן וידאו המתנגן.

2. פקד track (הרצה) זה פקד המאפשר הרצה של תוכן הוידאו במהירות גבוהה או נמוכה יותר.

במצב ברירת מחדל כאשר לא מוגדר הצגת פקדים בנגן וידאו בדפדפן, אלא ניתן להציגם ע"י הדפדפן באמצעות תפריט הנפתח בדפדפן ע"י לחיצת קליק בלחצן הימני מאפשר הצגת פקדים.

התכונה controls הינה תכונה בוליאנית מאחר ומניתן לעשות בה שימוש רק ע"י שם התכונה וערך הוא רק שם התכונה כאשר ב-html בתכונה בוליאנית אין חובה לכתוב את הערך. כאשר בעצם תכונה בוליאנית אין שום רלוונטיות לערך.

לדוגמא:

```
<video controls src="movie.ogg"> </video>
```

הסבר על התכונה autoplay

התכונה autoplay מגדירה שתצוגת התוכן הוידאו תתנגן באופן אוטומטי מייד שיופיע נגן הוידאו בדף האינטרנט ללא עצירה.

התכונה autoplay הינה תכונה בוליאנית מאחר ומניתן לעשות בה שימוש רק ע"י שם התכונה וערך הוא רק שם התכונה כאשר ב- **html** בתכונה בוליאנית אין חובה לכתוב את הערך. כאשר בעצם תכונה בוליאנית אין שום רלוונטיות לערך.

לדוגמא:

```
<video autoplay src="movie.ogg"> </video>
```

הסבר על התכונה loop

התכונה loop מגדירה שתצוגת התוכן של הוידאו לאחר שמסיים את ניגון התוכן וידאו באורך הזמן של התוכן יחל מחדש את הפעלה של התוכן באופן אוטומטי.

התכונה loop היא בעצם לולאה אינסופית כאשר הוא מפעיל שוב ושוב מחדש את הנגן וידאו לאחר שהתוכן וידאו הסתיים באורך הזמן שהתוכן אמור להיות מוצג.

התכונה loop הינה תכונה בוליאנית מאחר ומניתן לעשות בה שימוש רק ע"י שם התכונה וערך הוא רק שם התכונה כאשר ב- **html** בתכונה בוליאנית אין חובה לכתוב את הערך. כאשר בעצם תכונה בוליאנית אין שום רלוונטיות לערך.

מומלץ בכל התכונות הבוליאניות שפירטנו או שנפרט על מנת שיהיה קריא לכתוב את שם התכונה וסימן השוויון וצד גרשיים ללא ערך ביניהם או עם ערך של שם התכונה עצמה.

לדוגמא:

```
<video loop src="movie.ogg"> </video>
```

הסבר על התכונה muted

התכונה muted מגדירה שהפלט אודיו של תוכן הוידאו יהיה מושתק במצב של ברירת מחדל.

התכונה muted הינה תכונה בוליאנית מאחר ומניתן לעשות בה שימוש רק ע"י שם התכונה וערך הוא רק שם התכונה כאשר ב- **html** בתכונה בוליאנית אין חובה לכתוב את הערך. כאשר בעצם תכונה בוליאנית אין שום רלוונטיות לערך.

לדוגמא:

```
<video muted src="movie.ogg"> </video>
```

הסבר על התכונה preload

התכונה preload הינה תכונה המפרטת איך יוצר דף האינטרנט חושב לגבי אופן שבו טעינת הוידאו תתבצע במקביל שדף האינטרנט בכללותו נטען.

התכונה preload מאפשרת ליוצר הדף אינטרנט לספק לדפדפן רמז לגבי החשיבה שלו מה מוביל את חוויית המשתמש המיטבית ביותר ביחס לטעינת נגן וידאו של HTML 5 כך שהוא מגדיר לדפדפן איך הוא רוצה שטעינת הוידאו תתבצע במקביל שדף אינטרנט נטען.

ישנם שלשה ערכים אפשריים בתכונה **none, auto, metadata**

none

באמצעות נתינת הערך none בתכונה preload אנו מגדירים לדפדפן שלא יטעין את הוידאו כאשר הדף אינטרנט נטען עד שהמשתמש ילחץ על play.

auto

באמצעות נתינת הערך auto בתכונה preload אנו מגדירים לדפדפן שהוא צריך להטעין וידאו בכללותו לדף אינטרנט כאשר הדף אינטרנט נטען.

metadata

באמצעות נתינת הערך metadata בתכונה preload אנו מגדירים לדפדפן שהוא צריך להטעין את ה-metadata שהינם נתוני מידע כל הנגן וידאו כאשר הדפדפן אוסף את המידע הזה שכולל כאת גודל חלון של נגן הוידאו, משך הזמן של תוכן הוידאו, ורשימת רצועות של תכני הוידאו שצריכים להתנגן.

התכונה preload אינה יכולה לפעול יחד עם התכונה autoplay מאחר שאז הדפדפן מתעלם מהתכונה preload.

התכונה preload אינה נתמכת בדפדפן IE.

```
<video controls="" preload="auto" src="movie.ogg"> </video>
```

הסבר על התכונה poster

התכונה poster מגדירה שיוצג תמונה בחלון הנגן הוידאו כאשר מורד הקובץ וידאו לנגן של דף אינטרנט עד כאשר המשתמש ילחץ play.

הערך שהתכונה מכילה הינו כתובת URL של התמונה שזה יכול להיות בשני אופנים

URL relative כאשר זה שוכן באותו שרת או URL absolute כאשר מתייחס שהקובץ התמונה שוכן במקום אחר שהינו שרת אינטרנט שבו שוכן התמונה.

לדוגמא:

```
<video controls src="movie.ogg poster="star.png" > </video>
```

הסבר על האלמנט <source> ותכונותיו בוידאו

האלמנט source משמש לציון משאבי מדיה מרובים תחת האלמנט video או האלמנט audio. משאבי המדיה המרובים מתייחס לכך כאשר ישנם מספר קבצי וידאו שלרוב מדובר באותם קבצים בעלי פורמט שונה כשאז מקבצים אותם שכל קובץ אחד הינו תחת האלמנט source וביחד שוכנים בתוך האלמנט video.

היתרון שאלמנט זה נותן הוא בכך כאשר שדרך כלל שמים שם אותם קבצים אך בעלי פורמט שונים כאשר יוצר אלטרנטיבה באופן שהדפדפן לא תומך באחד מן הפורמטים של הקובץ זה מאפשר לדפדפן לבחור בקובץ עם פורמט עם קידוד כשהוא תומך בו

הסבר על התכונה src

התכונה src מפרטת את כתובת URL שבו תוכן הוידאו זה שוכן שזה יכול להיות בשני אופנים URL relative כאשר זה שוכן באותו שרת או URL absolute כאשר מתייחס שהקובץ וידאו שוכן במקום אחר שהינו שרת אינטרנט שבו שוכן תוכן וידאו.

לדוגמא:

```
<video controls>  
<source src="horse.ogg" type="video/ogg">  
<source src="horse.mp4" type="video/mp4">  
</video:>
```

בדוגמא אנו רואים שני קבצי וידאו שהם אותם קבצים אך בעלי פורמט שונה כאשר הראשון הינו בפורמט ogg ופורמט שני הינו MP4.

הסבר על התכונה type

התכונה type מפרטת את Internet media type (סוג פורמט הקובץ שנמצא על האינטרנט) של משאבי מדיה כאשר כאן אנו מתייחסים למשאבי מדיה שהם קבצי וידאו.

ערכים של התכונה הם כאשר אנו משתמשים בקבצי וידאו הינם הפורמטים של קבצי הוידאו כאשר התחביר הינו `source type="video/media type`.

לדוגמא:

```
<video controls>
  <source src="horse.ogg" type="video/ogg">
  <source src="horse.mp4" type="video/mp4">
</video:>
```

בדוגמא אנחנו רואים שהתכונה `type` מגדירה שמדובר באמצעי מדיה שהינו וידאו בפורמט MP4 או בקובץ האחר שמופרט שגם הוא מצעי מדיה שהינו וידאו בפורמט OGG.

הסבר על אלמנט `<track>`

האלמנט `track` מציין רצועת טקסט

פרק הטבלאות

תוכן:

1. מבוא לטבלאות.
2. מהי טבלה.
3. מבנה הבסיסי של הטבלה.
4. הסבר על אלמנט `th`.
5. הסבר על התפשטות העמודות בטבלה.

מבוא לטבלאות

ישנם סוגים אחדים של מידע שאתם צריכים להציג אותם כרשת או בשמו האחר טבלה. לדוגמא תוצאות ספורט, דו"ח מניות, ולוח זמני נסיעה של רכבת.

מהי טבלה

טבלה מציגה מידע בתוך פורמט של רשת. דוגמאות של טבלאות כוללות דו"ח פיננסי, לוח שידורים, ותוצאות ספורט.

הטבלאות מאפשרות לנו להבין נתונים מורכבים ע"י הפניית המידע בתוך שני צירים של אורך ורוחב.

כל ריבוע בתוך הטבלה מתייחסים איליו כתא בתוך הטבלה. בתוך מסמך HTML הטבלה נכתבת כשורה אחר שורה ולא כתצוגה ויזואלית של טבלה בתוך המסמך HTML עצמו.

מבנה הבסיסי של הטבלה

לצורך יצירת הטבלה ישנם ארבעה אלמנטים בסיסיים האלמנט `<table>` ו- `<tr>` ו- `<td>` ו- `<th>` כאשר אלמנטים הללו יוצרים מבנה הבסיסי של טבלה הכוללת את השורות ועמודות וכותרת הטבלה.

הסבר על אלמנט `<table>`

האלמנט `table` משמש ליצירת טבלה. כאשר התוכן של האלמנט זה נכתב שורה אחר שורה.

האלמנט כולל בתוכו האלמנטים הבסיסיים שציינו קודם שנחוצים ליצירת הטבלה

הסבר על אלמנט `<tr>`

האלמנט `tr` משתמשים בו להצביע על שורה חדשה בטבלה ע"י התג הפותח `<tr>`, (ה- `tr` הינו ראשי תיבות של `table row` (שורה בטבלה)).

האלמנט `tr` מכיל בתוכו את האלמנט `td` כאשר את האלמנט זה כותבים לאחר שכתבנו את האלמנט `tr`, (האלמנט `td` מציין תא בטבלה אחד באותה שורה).

בסוף כל שורה אנחנו משתמשים בתג הסוגר `</tr>`.

הסבר על אלמנט `<td >`

כל תא בטבלה מיוצג בשימוש באלמנט `td`. (האלמנט `td` הינו ראשי תיבות של `table data` (נתונים בתוך הטבלה))

בסוף כל תא אתם משתמשים בתג הסוגר `</td>`.

האלמנט `td` מייצג רק תא אחד בשורה, כאשר אתם רוצים מספר כלשהו של תאים בטבלה יש לבצע באותה שורה את האלמנט `td` כמספר הפעמים שאתם רוצים להוסיף.

את האלמנט `td` יש לכתוב בהזחה בתול האלמנט `td` לדוגמא:

`<table>`


```

<tr>
  <td>15</td>
  <td>15</td>
  <td>30</td>
</tr>
<tr>
  <td>45</td>
  <td>60</td>
  <td>45</td>
</tr>
<tr>
  <td>60</td>
  <td>90</td>
  <td>90</td>
</tr>
</table>

```

בדוגמא זו אנו רואים כאלמנט `table` מגדיר את התוכן שבו הינו טבלה ולאחר מכן שלוש פעמים נעשה שימוש באלמנט `tr` שמציין שישנם 3 שורות בטבלה ובכל שורה נעשה שימוש 3 פעמים באלמנט `td` המציין שישנם 3 תאים בכל שורה בטבלה.

הסבר על אלמנט `th`

האלמנט `th` משמש ליצירת תא בטבלה כאלמנט `td` אך בשונה ממנו האלמנט `th` משמש ליצירת תא של כותרת לעמודה בטבלה או שורה בטבלה ואף מוצג בצורה מודגשת. שם האלמנט `th` הינו ראשי תיבות של `table heading` (כותרת בטבלה).

אפילו אם אין תוכן בשורה כלשהי בטבלה אתם צריכים להשתמש באלמנט `td` או האלמנט `th` אף ללא תוכן ביניהם על מנת להציג את התאים הריקים בטבלה כי לולא כן הטבלה תוגש לתצוגה בדף באופן לא תקין.

האלמנט `th` אף מסייע לתוכנות קוראי מסך כאשר קוראים את תוכן האלמנט בהדגשה ככותרת של העמודה או שורה.

האלמנט `th` אף מסייע למנועי חיפוש במפתוח הדפים שלכם ע"י שימוש בכותרות העמודות בטבלה כחלק מן מאפייני התוכן שבאותו דף ע"י שכותרות הללו משמשות מילות מפתח בדף.

האלמנט `th` גם מאפשר לכם לשלוט טוב יותר על ההופעה של הטבלה כאשר אתם מתחילים לעשות שימוש ב-CSS כדי לעצב את תצוגת הטבלה מאחר והדגשה שהאלמנט `th` נותן

מסייע מבנה ברור וטוב של טבלה שאיתו העיצוב ה-CSS נהיה פשוט יותר כאשר מבנה הטבלה ברור לחלוטין.

הסבר על התפשטות העמודות בטבלה

התפשטות העמודות בטבלה משמעותו שתא בטבלה משתרע יותר מעל פני עמודה אחת בטבלה. זה נעשה בשימוש בתכונה colspan שקיימת באלמנטים td ו-th שמייצרים את תאי הטבלה הינו באופן שאנחנו צריכים שהתא בטבלה ישתרע על פני יותר מעמודה אחת בטבלה. הערך של התכונה מציין כמה עמודות בטבלה התא משתרע עליהם.

לדוגמא:

```
<table>
  <tr>
    <td colspan="2">20</td>
  </tr>
```

בדוגמא זו התכונה colspan מציינת שהתא בטבלה משתרע על פני 2 עמודות בטבלה.

פרק form (טופס)

תוכן:

1. מבוא ל- form
2. הסבר על אלמנט <form> ותכונותיו
3. הסבר על האלמנט <input>
4. הסבר על תכונת type באלמנט <input>
5. הסבר על תכונת placeholder באלמנט <input>
6. הסבר על תכונת maxlength באלמנט <input>
7. הסבר על תכונת minlength באלמנט <input>
8. הסבר על תכונת multiple באלמנט <input>
9. הסבר על תכונת pattern באלמנט <input>

מבוא ל- form

form (טופס) הינו מושג ב-HTML המתייחס לאלמנטים שונים שמאפשרים לכם לאסוף מידע מהגולשים עבור האתר שלכם.

שימוש ידוע בטופס כלדוגמא הינו תיבת החיפוש של גוגל שבו אתם מזינים ערכי חיפוש שזהו בעצם פעולת איסוף מידע מהגולש שמזין ערכים וכך ע"י זה מבוצע חיפוש ע"י הרובוט של גוגל באותו מידע שנאסף מהגולש עבור אתר.

הסבר על controls (פקדים) של form

ב- form על מנת לאסוף מידע מן גולשי האתר עבור אתר שלכם ישנם סוגי controls (פקדים) שונים שבאמצעותם אתם יכולים לעשות שימוש עבור איסוף מידע מן המשתמשים עבור אתר שלכם.

סוג ראשון של פקדים הינו בקטגוריית הוספת טקסט (adding text) כאשר הינם פקדים שאוספים את טקסט שהוזן ע"י המשתמש עבור אתר שלכם כאשר בקטגוריה זו אנו מוצאים:

1. **Text input** מדובר בשורה יחידה של תיבת טקסט המאפשרת הזנת קלט של טקסט כשיכול לשמש ככתובת מייל או שם או שאר מילוי פרטים אישיים.
2. **Password input** כמדובר בשורה יחידה של תיבת טקסט המאפשרת הזנת קלט של טקסט בשורה יחידה אך זה מסווה את התווים שבטקסט שמוזנים בו ומשמש לכתיבת סיסמאות.
3. **Text area** מדובר בתיבת טקסט המאפשרת הזנת קלט של טקסט בתוך שטח התיבה המשתרע יותר מן שורה אחת, ומשמש לתגובות או הערות ושאר מסרים.

סוג שני של פקדים הינו בקטגוריית יצירת בחירות (making choices) כאשר הפקדים אוספים נתונים לגבי האפשרויות שאותם סימנו המשתמשים בין אפשרויות הבחירה שהוצגו בפניהם, כאשר בקטגוריה זו אנו מוצאים:

1. **Radio buttons** זה בעצם כפתור שנועד לשימוש כאשר על המשתמש צריך לבחור אפשרות אחת מתוך מספר אפשרויות כאשר לכל אפשרות ישנו radio button כפתור שהמשתמש מסמן בכפתור תחת האפשרות הרצויה מבין האפשרויות.
2. **Checkboxes** זה בעצם תיבת סימון שנועד לשימוש כאשר על משתמש מוצג מספר אפשרויות ויש צורך לאפשר לבחור יותר מן אפשרות אחת או שאין חובה כלל לבחור לכך, ישנו את ה- Checkboxes שניתן לסמן וי בתיבה אפשרויות הרצויות.
3. **Drop-down-boxes** זה בעצם תפריט נשלף שנועד לשימוש כאשר על המשתמש צריך לבחור אפשרות אחת מתוך מספר אפשרויות כאשר כל אפשרות עולה בתיבה ע"י דפדוף כאשר אותה אפשרות שהמשתמש משאיר שתוצג בתיבה הינה האפשרות שהוא בחר.

סוג שלישי של פקדים הינו בקטגוריית הגשת הטופס (submitting forms) כאשר הפקדים משמשים להגשת הטופס לאחר מילוי הפרטים הנחוצים למילוי בטופס וע"י שימוש בפקדים הללו נשלח הטופס עם המידע שהזין המשתמש לאתר שלכם דהיינו לבסיס הנתונים של האתר שלכם. בקטגוריה זו אנו מוצאים:

1. **Submit button** זה בעצם כפתור שע"י המשתמש לוחץ עליו נשלח הנתונים מן הטופס

שהמשתמש הזין בטופס לאתר שלכם דהיינו לבסיס הנתונים באתר שלכם ולאחר מכן מכן מעביר אותכם לדף אינטרנט אחר או שרק מציג הודעה שהנתונים התקבלו.

2. **Image button** זה בעצם כפתור שע"י המשתמש לוחץ עליו נשלח הנתונים מן הטופס שהמשתמש הזין בטופס לאתר שלכם דהיינו לבסיס הנתונים באתר שלכם אך בשונה מן הכפתור submit שמוצג רק כפתור רגיל כאן הכפתור של הגשת הנתונים נעשה שימוש בתמונה למטרה זו.

סוג רביעי של פקדים הינו בקטגוריית העלאת קבצים (uploading files) כאשר הפקדים נותנים למשתמש יכולת לצרף קובץ ולשולח לבסיס נתונים של האתר שלכם. בקטגוריה זו אנו מוצאים:

1. **File upload** זה בעצם כפתור המאפשר למשתמש ע"י לחיצה בכפתור זה לצרף קובץ שיכול להיות מסמך או תמונה ושאר סוגי קבצים שונים.

איך forms (טפסים) פועלים

הפעילות של from (טופס) מתבצעת על פני 4 תהליכים:

1. הזנת תוכן.
2. שליחת הטופס לשרת.
3. עיבוד נתונים ואחסון.
4. קבלת תשובה מהשרת.

הזנת תוכן

התהליך הראשון המתקיים בטופס הינו כאשר המשתמש מזין תוכן הנדרש להזנה בטופס, ולוחץ Submit לשליחת הנתונים לשרת.

שליחת הטופס לשרת

הטפס נשלח לשרת עם שם הטופס יחד עם ערכים שהמשתמש הזין או בחר.

עיבוד נתונים ואחסון

השרת מעבד את הנתונים שהתקבלו מן הטופס, כאשר השרת משתמש על מנת לעבד נתונים באמצעות שפות התכנות כגון - PHP, VB.net, C#, Java. ולאחר מכן מאחסן את הנתונים בבסיס נתונים.

קבלת תשובה מהשרת

השרת יוצר דף חדש ושולח אותו בחזרה לדפדפן שהינה תוכנת הלקוח (המשתמש) כאשר נתונים הקיימים באותו דף קשורים למידע שהתקבל מן המשתמש כלדוגמא נשרת שולח דף עם תשובה כשהמידע נשלח והתקבל.

הסבר על אלמנט <form> ותכונותיו

הטפסים שוכנים בתוך האלמנט form. האלמנט form יכול להכיל גם אלמנטים אחרים כגון אלמנטים של פיסקאות, כותרות וכו'.

האלמנט form אינו יכול להכיל אלמנט form אחר בתוכו כך שבדף אינטרנט כאשר אתם רוצים ליצור טפסים שונים שיכולים להכיל באותו דף אינטרנט כגון login form, search form וטופס מנוי לניוזלטר אתם צריכים ליצור זאת ע"י אלמנטים של form בנפרד. כך כאשר יש לכם יותר מטופס אחד באותו דף המשתמש יכול לשלוח נתונים לשרת מן כל טופס בנפרד בפני עצמו.

בכל דף אינטרנט שיש בו את האלמנט form האלמנט זה צריך להכיל לפחות שתי תכונות שהם action ו- method.

הסבר על התכונה action

התכונה action מצביעה לגבי מה שמתרחש לגבי אותם נתונים לאחר שנשלחו. כאשר הערך של התכונה מצביע לגבי איפה הנתונים האלו נשלחים ע"י נתינת כתובת URL שאליה הנתונים נשלחים כאשר הכתובת URL יכול להיות absolute URL או relative URL כשבכתובת זו שאליה נשלחים הנתונים יכול להיות דף או תוכנה בשרת שמקבל את הנתונים. לדוגמא:

```
<form action="http://www.example.org/membership/login.aspx">
```

בדוגמא אנו רואים כאשר יש לנו login form (טופס כניסה למערכת) המורכב משם משתמש וסיסמא, אנו רואים שהפרטים שהמשתמש הזין מועברים לדף הנכתב ב- ASP.NET השוכן בשרת ווב ונקרא login.aspx כפי שמובא בערך של התכונה action.

הסבר על התכונה method

התכונה method מפרטת איך לשלוח את נתוני הטופס לאותו דף בשרת המפורט בתכונה action כאשר ישנם שני דרכים לשליחת נתוני הטופס באמצעות הפרוטוקול HTTP:

1. **get**

2. **post**

הפרוטוקול HTTP הינו בשימוש כאשר אתם רוצים להשיג יכולת לעשות שימוש במשאב של תוכן על גבי האינטרנט דף HTML אזי הדפדפן שולח בקשה באמצעות הפרוטוקול HTTP ע"י כתובת URL שאותו מצינים וכך גם השרת שולח בחזרה את הדף או תוכן המבוקש באמצעות הפרוטוקול HTTP. הבקשה הנשלחת ב- HTTP מורכבת משני חלקים header ו-body כאשר header שמכיל נתונים כוללים אודות הדפדפן הכוללים סוג הדפדפן וגרסתו שממנו נשלח הבקשה ויכולתו של הדפדפן לקבל את הנתונים מן השרת. חלק שני body הינו גוף הבקשה של המידע המבוקש שבו בעצם מופרט המידע המבוקש מן השרת. כך גם בטופס שנשלח כבקשה ע"י הדפדפן בכתובת URL איליו הטופס נשלח באמצעות הפרוטוקול HTTP באמצעות שני דרכים get או post.

הסבר על get

ה- get מתייחס לכך ששיטת שליחת נתוני הטופס הנשלחת מהדפדפן לשרת הינה כאשר הנתונים נשלחים כחלק מ- URL בתור פרמטרים ב- URL וזה גם מצב ברירת המחדל של שיטת שליחת נתוני הטופס.

ניתן לתאר את שיטת שליחת הנתונים get כאשר הדפדפן שולח בקשה לשרת שהוא רוצה לקבל יכולת של שימוש במשאב על גבי הרשת וכאשר גוף הבקשה ריקה ולכן הנתונים בטופס שנשלחים לשרת נוספים לכתובת URL.

נתוני הטופס נוספים בתוך ה- URL פרמטר עם name (שם) ו- value (ערך) כאשר סימן שוויון מפריד בין השם הפרמטר והערך שלו או בצמד של שמות וערכים כלדוגמא שני פרמטרים של user name ו- password כשבהם יש ערכים שבין ובין שמות הפרמטרים וערכים מפריד סימן שוויון וכך גם סימן & המציין הפרדה בין שני הפרמטרים הנשלחים. הפרמטרים הנשלחים בכתובת URL נקראים query string כאשר בין הפרמטרים לכתובת URL ישנו ? מפריד.

לדוגמא:

www.example.com/?userName=dsds@email.com&password=b6h42

בדוגמא אנו כיצד מבוצעת שליחת נתונים של טופס login בשיטת get.

מאחר ונתוני הטופס נשלחים בשיטת ה- GET בכתובת URL על כן אין לשלוח נתונים רגישים בשיטת GET מאחר כשהם נשלחים כחלק מ URL הם גלויים לעיני כל. מסיבה זו שפירטנו כעת לגבי נתונים רגישים על כן פחות מומלץ בכלל להשתמש בשיטת GET בשליחת הנתונים.

האורך התווים של כתובת URL מוגבל בערך לכ- 3,000 תווים שזה יכול להיות עוד סיבה שפחות מומלץ להשתמש ב-GET כאשר זה יוצר מצב שכמות הנתונים הניתנת לשליחה היא די מוגבלת שעומדת כ- 1024 תווים לשימוש ב- query string מתוך הכמות התווים הכוללת שהינה בערך 3000 תווים. על כן גם שימוש בנתונים בעברית בשיטת GET עוד פחות מומלץ מפני שזה יוצר הגבלה די משמעותית מאחר ואותיות בעברית תופסות גודל של כבערך 2-3 תווים דבר שמביא את כתובת URL לאורך שמוגבל בערך בין 1,000 ל- 2,000 תווים כך שכמות הנתונים ב query string קטנה יותר ומוגבלת לחלוטין.

מגבלת הנתונים נוספת כאשר לא ניתן לשלוח בשיטת GET לשרת מידע בינארי שזה כולל מסמכי וורד, קבצי טקסט ותמונות.

בשיטת get לא ניתן להעלות כלל קבצים מאחר ולא ניתן להעביר קבצים דרך URL שזו הדרך של העברת הנתונים בשיטת get.

שימוש בשליחת נתונים בשיטת GET שא זה יכול להיות שימושי באופן כאשר המשתמש רוצה ליצור סימניה בתוצאות של הנתונים של הטופס שהוא שלח מאחר והנתונים מוזכרים

ב- query string כחלק מכתובת URL מחמת כך ניתן ליצור סימניות מהתוצאות של הנתונים בקלות.

ניתן למצוא שימושיות בשליחת נתוני טופס בשיטת GET בטפסים קצרים כלדוגמא תיבת חיפוש מאחר וכמות הנתונים מוגבלת בשליחת נתוני טופס בשיטת GET

הסבר על post

ה- post מתייחס לכך ששיטת שליחת נתוני הטופס הנשלחת מהדפדפן לשרת הינה כאשר הנתונים שהם בעצם הפרמטרים המכילים את המידע נשלחים בתוך גוף הבקשה ב- HTTP

יתרונותיה של שיטת post לשליחת נתוני טופס הינה בכך שאין הגבלה בגודל הנתונים וניתן לשלוח נתונים רגישים מאחר והנתונים נשלחים בתוך הבקשה (request) ב- HTTP כך שזה לא נראה בגלוי לעיני כל כבאופן שזה היה נשלח כחלק מ-URL.

ניתן אף להעלות קבצים כלדוגמא תמונות ומסכי וורד ושאר קבצי טקסט מאחר ושאינן הגבלת בגודל הנתונים ואף ניתן לשלוח נתונים בינאריים הכוללים את הדוגמאות הנ"ל שהוזכרו לגבי קבצים שניתן להעלות.

חסרונו רק שלא ניתן ליצור סימניה מתוצאות הנתונים שנשלחו מאחר וזה נשלח בגוף הבקשה (request) ב- HTTP.

הסבר על התכונה id באלמנט - form

התכונה id מאפשרת לזהות באופן ייחודי את האלמנט form בדף, כאשר זו בדיוק עצם שימוש בתכונה זו כאתם יכולים להשתמש עימה לזהות באופן ייחודי איזשהו אלמנט בדף.

השימוש בתכונה id עבור כל אלמנט של form לתת לו ערך של מזהה ייחודי זה הרגל טוב מפני אופן שישנם טפסים רבים בדף ורבים מהם נדרש שיעשה בהם שימוש ב-עיצוב ב-CSS או שימוש בסקריפטים כך שיתכן באופן שישנם מספר טפסים יידרש שימוש בתכונת id על מנת לזהות את הטופס שבו צריך לעשות שימוש ב-CSS או בסקריפטים.

הערך של התכונה id צריך להיות ייחודי בתוך הדף ולא יהיו אחרים בעלי אותו ערך של id.

לדוגמא:

```
<form id="login-form"
action="http://www.example.org/membership/login.aspx"method="post">
```

הסבר על התכונה name באלמנט - form

התכונה name מציינת את השם של הטופס (form) כמו בתכונה id עם שם ייחודי כך שניתן יהיה לזהות האלמנט form בדף וניתן לעשות בו שימוש בתכונה של name על מנת לזהות את הטופס שצריך לעשות בו שימוש בג'אווה סקריפט אך כיום שינו את תכונה id שמבצעת בדיוק אותו דבר זה אינו רלוונטי אע"פ שהתכונה זו של name נתמכת ב-HTML 5 היא איבדה את הרלוונטיות שלה לגבי שם טופס באלמנט form עצמו אך הרלוונטיות שלה הינו כתכונה באלמנט input.
לדוגמא:

```
<form name="login-form"
action="http://www.example.org/membership/login.aspx" method="post">
```

הסבר על התכונה enctype

באופן שאתם משתמשים בשיטת post לשליחת נתונים לשרת אזי ניתן להשתמש בתכונה enctype לציין איך הדפדפן מקודד את הנתונים לפני שהוא שולח לשרת. הדפדפנים תומכים בשלושה סוגי קידוד שהן:

1. **application/x-www-form-urlencoded**
2. **multipart/form-data**
3. **text/plain**

חשיבות שימוש בתכונה enctype הינו בכך כאשר מונע מנתונים ללכת לאיבוד כנשלחים לשרת מחמת שהשרת אינו מצליח לקרוא את תווים שמכילים הנתונים וכך ע"י קידוד אנו מאפשרים לשרת לקרוא את אותם נתונים.

הסבר על application/x-www-form-urlencoded

שיטת הקידוד application/x-www-form-urlencoded הינה שיטת הקידוד הסטנדרטית שמרבית הטפסים נעשה שימוש ע"י הדפדפנים בשיטה קידוד נתונים זו שהיא אף ברירת המחדל כאשר לא מוגדר שום שיטת קידוד של הנתונים הקידוד בדרך הזו. הדפדפנים משתמשים בשיטה זו כברירת מחדל מפני שתווים כמרווחים וסימן פלוס ושאר התווים שהם אינם אלפאנומריים שהם אינם יכולים להישלח לשרת ווב. ובמקום זה התווים הללו מומרים לערכי ASCII HEX שהינם ערכים אלפאנומריים שהם מחליפים את אותם תווים.

הנתונים של הטופס בשיטה זו מקודדים את הנתונים לתוך URL של ה-form על כן לא ניתן בשיטה זו להעלות קבצים עם שיטת קידוד זו מאחר ולא ניתן לשלוח דרך URL

לדוגמא:

```
<form action="demo_form.asp" method="post" enctype="applicarion/x-www-
form-urlencoded">
E-mail: <input type="email" name="user_email">
<input type="submit">
</form>
```


הסבר על multipart/form-data

שיטת הקידוד multipart/form מאפשרת לנתונים להישלח בחלקים לשרת ולא שכל הנתונים ישלחו בבת אחת, החלקים נשלחים באופן רציף כאשר כל חלק נתונים הוא מהווה נתונים של פקד בטופס וכך שהחלקים של נתונים הללו שליחתם הם בסדר שהפקדים מופיעים בטופס.

בכל חלק של נתונים שנשלח בשיטת הקידוד multipart/form-data מופיע ב- HTTP header של אותו חלק שנשלח פרטים על סוג התוכן הנתונים בחלק ההוא כאשר מצביעים על סוג הנתונים של אותו פקד בטופס שהחלק מהווה נתונים שלו.

נעשה שימוש בשיטת קידוד זו כאשר ישנו צורך להעלות קבצים כגון תמונות לשרת מאחר ואינו מקודד את הנתונים לתוך URL וזה שמאפשר קידוד של נתונים רבים.

לדוגמא:

```
<form action="demo_form.asp" method="post" enctype="multipart/form-  
data">  
E-mail: <input type="email" name="user_email">  
<input type="submit">  
</form>
```

הסבר על text plain

שיטת הקידוד text-plain שולחת את הנתונים כטקסט רגיל שלא נעשה בו שום שינוי בתווים שאינם אלפאנומריים חוץ מן רווחים שממיר אותו לסימן פלוס. על כן לא מומלץ להשתמש בשיטת קידוד זו.

לדוגמא:

```
<form action="demo_form.asp" method="post" enctype="text-plain">  
E-mail: <input type="email" name="user_email">  
<input type="submit">  
</form>
```

השימוש בקידוד text plain זה טוב כאשר שולחים במייל את נתוני הטופס כאשר הקידוד text plain הופכים את נתוני הטופס לקריאים יותר

לדוגמא:

```
<form action="mailto:myforms@example.com" enctype="text/plain" >
```

באופן זה המוצג בדוגמא מומלץ להשתמש בקידוד של text plain מאחר שלולא השימוש בקידוד text plain הערכים ישלחו במייל כמצב ברירת המחדל שהינו מקודד - application/x-www-form-urlencoded ואזי הנתונים של הטופס ישלחו דרך הכתובת URL באופן שהוא אינו קריא ויראה כך כלדוגמא:
Name=mark&Email=mark@yahoo.com&Address=&Submit=Submit

על כן שימוש בקידוד textplain מומלץ מאחר והופך את הנתוני הטופס הנשלחים ללמייל קריאים כאשר יראו כך כלדוגמא:

```
Name=mark  
Email=mark@yahoo.com  
Address=  
Submit=Submit
```

הסבר על התכונה novalidate

התכונה novalidate הינה תכונה חדשה ב-HTML 5 שמתייחסת לכך שהטופס לא צריך לעבור ולידציה ע"י הדפדפן לפני שהטופס נשלח לשרת על מנת לבדוק האם טופס הוזן באופן הנדרש.

התכונה הזאת היא תכונה בוליאנית כאשר אין בה ערך אלא רק התכונה עצמה שניתן לפרט את התכונה כערך או לכתוב כלום או לשים רק שני גרשיים באופן כזה "" על מנת שיהיה קריא יותר.

לדוגמא:

```
<form action="demo_form.asp" novalidate>  
  E-mail: <input type="email" name="user_email">  
  <input type="submit">  
</form>
```

התכונה זו נתמכת בדפדפן IE רק מ- IE10

הסבר על האלמנט <input>

האלמנט input הינו אלמנט השוכן בתוך האלמנט form כאשר מגדיר את פקדי הקלט כאשר המשתמשים יכולים להזין בהם נתונים בתוך הטופס בכך שהאלמנט input מפרט את השדות שהמשתמשים יכולים להזין בהם נתונים.

באלמנט input הינו אלמנט ריק כאשר אינו מכיל תוכן בתוך האלמנט מלבד הערכים בתוכנות שלו.

הסבר על תכונות type - ו- name - ו- value באלמנט

<input>

מהי תכונת type

תכונת type באלמנט input מציינת את סוג הפקד של האלמנט input שיוצג בטופס.
הערך של התכונה type במצב ברירת מחדל כאשר לא מוגדר התכונה type אזי יהיה text

מהי תכונת name

התכונה name הינה מצביעה על שמו של האלמנט input שזה בעצם שמו של הפקד קלט.

הערך של תכונה name עם שם הפקד נשלח לשרת כאשר אתה מבצע פעולת שליחת הטופס ותפקידו להיות כמזהה של הפקד עבור השרת.

בתכונה name לא יכול להיות ערך ריק.

מהי תכונת value

התכונה value מפרטת את הערך ברירת מחדל של האלמנט input כלומר הערך של הפקד קלט(input).

ישנם שימוש שונה בערכי value של הערכים השונים ב type:

עבור ערכים של button -ו- reset -ו- submit הערך שבתכונה value מגדיר את הטקסט שנמצא על כפתור הפקד.

הסבר ערך value עבור text input -ו- hidden input

עבור ערכים של text -ו- password ושאר ערכי text input השונים שהינם שדות טקסט וגם הערך hidden שבתכונה value מגדיר את הערך הראשוני שהינו ערך ברירת המחדל של שדה הפלט שהינו הערך הנראה למשתמש כאשר הטופס נטען למעט הערך hidden כאשר פקד זה אינו נראה אלא רק בהצצה בקוד מקור ניתן לראות את ערך שלו, אע"פ ש

בערכי text input ניתן לעשות שימוש בתכונה value כאשר הערך מורה למשתמש לגבי מה צריך לכתוב בשדות הללו שהם הוא מופיע אך פחות מומלץ לעשות שימוש בכך מאחר וזה מצריך שערך זה יוצג בתוך השדה בצורה מפורשת כאשר ניתן לעשות שימוש בתכונה placeholder כאשר מרמז באופן ברור ע"כ ככאשר לא נעשה שימוש ממש בתור ערך מפורש הכתוב בתוך השדה אלא ברמז הנותן הוראה מה לכתוב.

הערך של value כאשר מופיע כערך ראשוני בתכונות של פקדי text input אינו נשלח לשרת מאחר והוא מציין את הערך שהינו ברירת המחדל.

ערכי value של text input נשלחים לשרת אך לא הערכים שבהם המצוינים כברירת מחדל. רק הערכים שהוזנו ע"י המשתמשים ב- text input נשלחים לשרת.

הסבר על ערך value עבור radio button ו-check box

עבור ערכים הללו מדובר בערכים שהינם ערכים של הכפתורים כאשר כל ערך בכפתור מייצג את האפשרות שבאותו כפתור כך כלדוגמא ב-radio button כאשר מציגים מספר אפשרויות וכל ערך של radio button מייצג ערך של אפשרות שמציג אותו radio button .

הערך של value נשלח לשרת יחד עם ערך של name בשונה מן value ב-text input כאשר אינו נשלח לשרת.

הסבר על ערך value עבור button controls

עבור ערכים של button control כאשר מדובר ב-submit ו-image ו-restart הערכים הינם מה שכתוב על כפתור עצמו

הערכים של ה-value הללו נשלחים לשרת יחד עם ערך של name.

הסבר על ערך value עבור file

הערך של value ב-file בברירת מחדל הינו ריק מחמת סיבות אבטחה מאחר והערך אמור להיות קובץ כאשר הערך היה אמור לציין את מיקום הקובץ בתוך המחשב שממנו מועלה הקובץ על כן לא מצויין.

הסבר על ערך text

text הינו ערך של התכונה type המגדיר ששדה הקלט יהיה שדה טקסט של שורה אחת שלא ניתן לרדת יותר משורה אחת ואורך התווים במצב ברירת מחדל הינו כ-20 תווים כאשר לא מוגדר התכונה maxlength המגדירה את אורך התווים בשדה טקסט.

text הינו מצב ברירת המחדל של התכונה type כאשר לא מוגדר התכונה type באלמנט Input.

שדה טקסט מכונה זה ניתן לכנותו גם כתיבת טקסט בעלת שורה אחת או כ-text input .

בתיבת טקסט בעלת שורה אחת ניתן להשתמש לשדה הזנה של פרטים האישים הכוללים שדות של שם פרטי, משפחה כתובת, עיר ומס' דירה. אך ישנם פרטים אישים כתאריך וכתובת אימייל שלהם וטלפון ישנם שדות מיוחדת.

אך באופן שייתמך בדפדפני IE בגירסאות קודם ל-IE10 וב-Chrome בגירסאות הקודמות ל-5.0 chrome וב-Firefox בגירסאות הקודמות ל-4.0(2.0) Firefox מס' טלפון וכתובות אימייל היו ממלאים את זה בשדה טקסט של שורה אחת כאשר מוגדר כערך של התכונה type=text.

הערך file 74 הינו מקטגורייה של text input controls

הסבר על הערך password

password הינו ערך של התכונה type המגדיר שדה קלט שיהיה בדומה לשדה טקסט בעל שורה אחת אף הוא שדה טקסט בעל שורה אחת אך השוני בינו לשדה טקסט בעל שורה אחת הוא בכך כאשר השדה מכסה את התווים שהמשתמש מזין לתוכו כך שהתווים אינם יכולים להיראות על גבי המסך.

בשדה password (הסיסמה) מוצג בשדה הזה נקודות או כוכביות כאשר המשתמש מזין לתוך השדה תווים במקום התווים כאשר המשתמש מזין לתוך השדה password (סיסמה) הכוכביות או הנקודות שמוצגות במקום התווים מוצגים כמספר התווים שהמשתמש הזין לתוך התיבה.

שדה זה מכונה שדה password (סיסמה) מאחר והוא משמש להזנת סיסמאות.

בשדה password (סיסמה) נעשה שימוש להזנת סיסמאות ב- login form (טופס כניסה למערכת), או כאשר מזינים נתונים רגישים כגון מספרי כרטיס אשראי.

בשדה password רצוי להשתמש בהגבלת מס' התווים שניתן להזין לתוך השדה הסיסמה שזה נעשה ע"י תכונה maxlength.

הערך password הינו מקטגורייה של text input controls

הסבר על הערך checkbox

checkbox הינו ערך של התכונה type המגדיר checkbox (תיבת בחירה) שניתן בעצם להציג את תיבה זו כמו מתג ש לחשמל להפעלת אור כאשר ישנם שני מצבים on ו-off כלומר שהמתג מופעל ע"י לחיצה או שמכובה שהמתג לא לחוץ כלל או שמכובה ע"י הסרת הליחס, כך גם ב checkbox (תיבת בחירה) ישנו שני מצבים לבחירה של on ו-off כאשר המשתמש בוחר בתיבה ע"י לחיצה על כפתור של התיבה נמצאת במצב on כאשר הוא לא בוחר כלל את התיבה או ע"י לחיצה בכפתור שמבטלת את הבחירה שעשה קודם לכן אזי התיבה נמצאת במצב off.

checkbox (תיבת בחירה) יכולה להופיע באופן פרטני כאשר כל תיבה יש לה שם משלה בפני עצמה כלדוגמא: תיבת בחירה האם יש לך רישיון נהיגה כאשר שם התיבה רישיון נהיגה ותיבת בחירה שנייה שתופיע בדף האם יש לך רכב כאשר שם התיבה הינו רכב. אפשרות שנייה לתצוגת התיבות בחירה שניתן להציגם שיופיעו כקבוצה של תיבות בחירה אשר לפקד של התיבות בחירה הללו ישנו שם משותף והשתמש צריך לבחור בין הערכים של התיבות שהינם אפשריות הניתנות לבחירה לאותו מאפיין כלדוגמא ארבע תיבות בחירה כאשר הינם תיבות בחירה של מצב אישי כאשר ארבע תיבות הללו מציגות אפשרויות רווק, נשוי, גרוש ואלמן.

השימוש ב- checkbox (תיבת בחירה) הינו אידיאלי לשימוש בטופס באופנים הבאים:

1. אספקת תגובה כן ולא עם תיבה אחת לאפשריות כמו של קבלת תנאים והתניות למיניהם כגון כתנאי שימוש או התניה של אפשרות לקבלת פרסום מן החברה אם תסומן תיבת הבחירה .
2. בחירת סעיפים אחדים מתוך רשימת אפשרויות המתאפשרות לבחירה כלדוגמא כאשר אנחנו רוצים שהמשתמש יצביע ע"י בחירתו מתוך רשימת אפשרויות של מיומנויות בשפות תכנות מהם המיומנויות שהם הוא שולט.

הסבר על ערך radio

radio הינו ערך של התכונה type המגדיר radio button (כפתור בחירה) שדומה ל- check box (תיבת בחירה) ישנם שני מצבים לבחירה on ו-off אך ישנם שני הבדלים:

1. הינו כאשר יש לכם קבוצה של radio buttons (כפתורי בחירה) שיש להם שם משותף אזי רק אפשרות אחת יכולה להיבחר. וגם לאחר שנבחרה אפשרות ב- radio button מתוך האפשרויות כאשר המשתמש ילחץ על אפשרות אחרת אזי האפשרות החדשה היא הנבחרת והבחירה באפשרות הישנה שנבחרה מתבטלת. משא"כ ב- checkbox המשתמש יכול לבחור יותר מן אפשרות אחת בנוסף לאפשרות שבחר קודם.
2. אתם צריכים לא להשתמש ב- radio buttons (כפתורי בחירה) כפקד יחיד המצביע כעל שתי ערכים של on ו-off דהיינו כן ולא לגבי אפשרות אחת כלדוגמא אפשרות הם היש לך רישיון נהיגה והערכים הינם radio button כאשר אתם לוחצים אזי כן ואם לא לוחצים אזי לא מאחר שבכפתור בחירה יחיד לאחר שנבחר לא ניתן לבטל את הבחירה כלל ללא כתיבת סקריפט שניתן לבטל זאת. משא"כ ב- checkbox ניתן לבטל אף באפשרות אחת להחליט אם כן או לא דהיינו on ו-off.

עקב כך עפ"י מה שראינו בשני ההבדלים אזי השימוש בקבוצה של - radio buttons (כפתורי בחירה) הינו אידיאלי כאשר אתם רוצים לספק למשתמש מס' של אפשרויות שהוא יצטרך לבחור אחת מן האפשרויות. במצב זה ישנו גם אלטרנטיבה לעשות שימוש ב-- drop-down box (תיבה נפתחת) שמאפשרת למשתמשים לבחור אפשרות אחת מתוך אפשרויות אחדות. לגבי החלטה שלכם האם להשתמש ב- drop-down-box (תיבה נפתחת) או בקבוצה של radio buttons (כפתורי בחירה) בהתאם לשלושה דברים:

1. **ציפיות המשתמשים:** הינו כאשר אם הדגם שהטפסים שלכם מתבסס על טופס נייר כאשר שלמשתמשים הטופס זה יהיה מוצג עם תיבות בחירה שניתן לבחור רק אפשרות אחת אזי אתם צריכים להשתמש בקבוצה של radio buttons (כפתורי בחירה) שאז ניתן לבחור אפשרות אחת מתוך האפשרויות.
2. **ראיית כל האפשרויות:** כאשר אם המשתמשים ייהנו מהצורך שכל האפשרויות יהיה לפניהם בזמן שעדיין לפני שהם בחרו אפשרות אחת, אזי אתם צריכים להשתמש בקבוצה של radio button (כפתורי בחירה) שבו מוצג לפני המשתמשים כל האפשרויות.

3. **מקום:** באופן שאתם מתייחסים אודות למרווחים , לדוגמא בתקן של מובייל שהשטח התצוגה קטן אזי נשתמש drop down-box select (תיבה נפתחת לבחירה) שתופס פחות מקום מאשר רשימה של radio buttons (כפתורי בחירה).

הסבר על הערך file

file הינו ערך של התכונה type המגדיר file select box (תיבת בחירה של קובץ) כאשר אם אתם רוצים להעלות קובץ לאתר אינטרנט שלכם מהמחשב שלכם אזי אתם צריכים להשתמש file upload box (תיבת העלאה של קובץ) שידוע גם בשמו file select box (תיבת בחירה של קובץ).

file select box (תיבת בחירה של קובץ) צריך להיות בטופס שנשלח בשיטת post או בשיטה אחרת של שליחה שזה לא נעשה דרך כתובת URL כאשר לא מאפשר שליחת קבצים מחמת המגבלות הקיימות בגודל המוגבל הנתונים שניתן לשלוח ואף שלא קורא מידע בינארי ואף לכן קידוד מידע של הנתונים צריך לא להיות אף דרך ה- URL כאשר מדובר על שיטת קידוד multipart/form-data.

על מנת להגדיר את סוג הקבצים שניתן להעלות באמצעות file select box (תיבת בחירה של קובץ) יש לעשות שימוש בתכונה accept שמגדירה איזה סוג קבצים ניתן לעלות באמצעות תיבת הבחירה של הקובץ.

הערך file הינו מקטגורייה של text input controls

הסבר על ערך hidden

הערך hidden בתכונה type מתייחס כאשר לפעמים שאתם רוצים להעביר מידע בין דפים ע"י שליחת מידע זה לשרת ללא שהמשתמשים רואים שישנו אפשרות העברת מידע אזי אתם משתמשים ב- hidden form controls (פקד מוסתר של טופס). למרות שהמשתמשים אינם יכולים לראות אותם בתוך הדף אינטרנט עצמו , אם יסתכלו בקוד המקור של הדף אינטרנט הוא הם יוכלו לראות את הערך hidden עם ציון הערך שהוא מעביר בתוך קוד המקור. עקב כך hidden form control (פקד מוסתר של טופס) מומלץ לא לעשות שימוש עבור העברת מידע רגיש שאתם לא רוצים שהמשתמש יראה.

בשימוש ב- hidden הינו פקדים שנעשה בהם שימוש עבור אופן העברת מידע של נתוני הטופס כלדוגמא שבשדות הנתונים שבתוך השרות נוצרים באמצעות ג'אווה סקריפט.

הסבר על ערך email

ערך email בתכונה type מגדיר שדה קלט כאשר אינו שונה משדה טקסט רגיל אך המאפיין שהינו עבור הזנת כתובת e-mail.

הערך email הינו ערך חדש החל מ- HTML 5 הינו נתמך בדפדפן IE10 ומעלה ובדפדפן chrome 5 ומעלה ובדפדפן firefox 4.0(2.0) ומעלה.

יתרון השימוש בערך email הינו בכך שמציג יותר סמנטיקה כאשר הערך email מציג שדה טקסט הדומה במבנהו לשדה טקסט רגיל אך המשמעות הסמנטית שתיבת טקסט של אימייל

הערך email מספק אף יתרון כאשר יוצר חוויית משתמש במובייל כאשר המשתמש במובייל מזין טופס אזי כאשר הסמן של התו נמצא בתוך שדה טקסט של אימייל עם ערך email אזי מוצג במקלדת הסימן @ והסיומת .com.

הערך email mail הינו מקטגורייה של text input controls

הסבר על ערך search

ערך search בתכונה type מגדיר שדה טקסט כאשר שדה בעצם תיבת טקסט אך משמשת להזנת ערכי חיפוש ושונה מקצת באופן עיצובי מתיבת הטקסט הרגילה ושימושו כאשר אתם רוצים למקם חיפוש בטופס שלכם.

הערך search הינו ערך חדש החל מ- HTML 5 הינו נתמך בדפדפן IE10 ומעלה ובדפדפן chrome 5 ומעלה ובדפדפן firefox 4.0(2.0) ומעלה.

מצוי שימוש בך באתרי e-commerce (סחר ומכר אלקטרוני) לשים תיבת חיפוש בטופס. עבור חיפוש כלדוגמא של מוצרים הנמצאים בחנות.

הערך search הינו מקטגורייה של text input controls

הסבר על ערך month

הערך month בתכונה type מגדיר שדה של תיבה נפתחת לבחירה כאשר נותן שתי אפשרויות בחירה של תפריט שנפתח של חודש בלבד אפשרות בחירה שניה של תפריט שפותח לוח שנה שיוצג במצב ברירת מחדל החודש הנוכחי עם השנה הנוכחית, הדפדוף בלוח שנה מאפשר לבחור חודש כולל עם שנה.

הערך month מאפשר לנו לעשות שימוש ליישם זאת בטופס כאשר אנחנו רוצים ליצור שדה של חודש כלדוגמא בפרטי כרטיס אשראי לציון בשדה את חודש שהינו תוקף כרטיס האשראי או בביצוע הזמנה של כרטיס טיסה להציג את חודש שהזמנה מתייחסת אליו.

יתרון השימוש בערך month כאשר מאפשר לנו לבחור חודש ולהציגו ע"י תיבת בחירה נפתחת בלא שימוש בספריות ב- javascript כ-JQuery, Dojo או YUI אלא באפן פשטני ע"י HTML 5 בלבד.

הערך month הינו ערך חדש החל מ- HTML 5 והינו נתמך בדפדפן Chrome 5 ומעלה אך לא נתמך ע"י IE ולא ע"י Firefox.

הערך month הינו מקטגורייה של text input controls

הסבר על הערך week

הערך week בתכונה type מגדיר שדה של תיבה נפתחת לבחירה כאשר נותן שתי אפשרויות בחירה של תפריט שנפתח של מס' שבוע בלבד כאשר בדפדוף ניתן לבחור מס' השבוע בשנה או אפשרות בחירה שנייה של תפריט שפותח לוח שנה שיוצג במצב ברירת מחדל החודש והיום בחודש הנוכחים והשנה הנוכחית, וע"י דפדוף בלוח שנה ניתן לבחור את השבוע בשדה.

הערך week מאפשר לנו לעשות שימוש ליישם זאת בטופס כאשר אנחנו רוצים ליצור שדה של שבוע בשנה כלדוגמא בהזמנת מנוי למשך מס' שבועות בשנה כאשר שדה זה יכול להכיל השבוע שבו החל המנוי ושדה נוסף של שבוע המציין את השבוע שמסתיים זמן המנוי.

הערך week הינו ערך חדש החל מ- HTML 5 והינו נתמך בדפדפן Chrome 5 ומעלה אך לא נתמך ע"י Firefox אך בדפדפני IE זה לא נתמך אלא ניתן להציגו רק בהורדת תוכנה nightly build

יתרון השימוש בערך week כאשר מאפשר לנו לבחור שבוע בשנה ולהציגו ע"י תיבת בחירה נפתחת בלא שימוש בספריות ב- javascript כ-JQuery , Dojo או YUI אלא באפן פשוטני ע"י HTML 5 בלבד.

הערך week הינו מקטגורייה של text input controls

הסבר על ערך time

ערך time בתכונה type מגדיר שדה של תיבה נפתחת לבחירה כאשר נותן אפשרות בחירה של תפריט שנפתח של שעה ומס' הדקות.

הערך time במצב ברירת מחדל לא מציג כלל שעה.

הערך time מאפשר לנו לעשות שימוש ליישם זאת בטופס כאשר אנחנו רוצים ליצור שדה של שעה כלדוגמא כאשר אנחנו רוצים לפרט בטופס של הזמנת השכרת רכב שבו אנו מפרטים את השעה שבו אנו רוצים שהרכב יהיה מושכר למזמין השכרה.

יתרון השימוש בערך time כאשר מאפשר לנו לבחור שעה ודקות ולהציגו ובתיבת בחירה נפתחת באתר בלא שימוש בספריות ב-javascript כ-jQuery , Dojo או YUI אלא באפן פשטני ע"י HTML 5 בלבד.

הערך time הינו ערך חדש החל מ-HTML 5 והינו נתמך בדפדפן Chrome 5 ומעלה אך לא נתמך ע"י Firefox אך בדפדפני IE זה לא נתמך אלא ניתן להציגו רק בהורדת תוכנה nightly build

הערך time הינו מקטגורייה של text input controls

הסבר על ערך date

הערך date בתכונה type מגדיר שדה של תיבה נפתחת לבחירה כאשר נותן שתי אפשרויות בחירה של תפריט שנפתח של יום בחודש וחודש ושנה כאשר הבחירה נעשית בתפריט זה נעשה ע"י דפדוף בתפריט שנפתח בכל חלק של תבנית התאריך כאשר הסמן נמצא בחלק ההוא בתוך התבנית כפי שמוצגת בשדה שהינה dd/mm/yyyy או אפשרות שנייה של תפריט שנפתח בלוח שנה כאשר במצב ברירת מחדל מציג את התאריך הנוכחי כאשר בחירה מתוך לוח השנה בוחר ישירות את התאריך במלואו.

התבנית תאריך המוצגת בשדה date הינו dd/mm/yyyy כאשר בצד שמאל זה מתייחס ליום בחודש ובאמצע מתייחס לחודש ובצד ימין מתייחס לשנה. תבנית זו מצגת עם הכיתוב dd/mm/yyyy במצב ברירת מחדל בשדה קודם שהוגדר תאריך בשדה.

הערך date מאפשר לנו לעשות שימוש ליישם זאת בטופס כאשר אנחנו רוצים ליצור שדה של תאריך וניתן לעשות שימוש כתאריך לידה או כתאריך הזמנה וכו'.

הערך date הינו ערך חדש החל מ-HTML 5 והינו נתמך בדפדפן Chrome 5 ומעלה אך לא נתמך ע"י Firefox אך בדפדפני IE זה לא נתמך אלא ניתן להציגו רק בהורדת תוכנה nightly build

יתרון השימוש בערך date כאשר מאפשר לנו לבחור תאריך ולהציגו ע"י תיבת בחירה נפתחת בלא שימוש בספריות ב-javascript כ-jQuery , Dojo או YUI אלא באפן פשטני ע"י HTML 5 בלבד.

הערך date הינו מקטגורייה של text input controls

הסבר על ערך datetime

הערך datetime בתכונה type מגדיר שדה של שתי תיבות נפתחות כאשר התיבה הראשונה פותחת תפריט של תאריך ותיבה שנייה פותחת תפריט של זמן וליד התיבה השנייה מופיע כיתוב שבו כתוב UTC הזמן המתייחס לזמן של השעון הבסיסי שעליו מתבסס הזמן בעולם הידוע יותר כשעון גריניץ.

בבחירת התאריך ניתן ע"י הזנה עצמאית או פתיחת הפריט כאשר פותח לוח שנה כאשר התאריך הנוכחי מוצג כברירת מחדל.

הערך `datetime` הינו ערך חדש החל מ-HTML 5 אך הוסר במשך הזמן בשימוש ב-HTML 5 ובמקום זאת עושים שימוש בערך `datetime-local`. לא נתמך ע"י IE ואף לא ע"י Chrome ואף לא ע"י Firefox. אך נתמך ע"י הדפדפן Opera בגירסת 10.62 .

הערך `datetime` הינו מקטגורייה של `text input controls`

הסבר על הערך `datetime-local`

הערך `datetime – local`

הסבר על תכונת `placeholder` באלמנט `<input>`

התכונה `placeholder` מפרטת רמז תמציתי המתאר את הערך המצופה של קלט השדה כאשר יכול להיות ערך פשוט כלדוגמא ערך בתוך שדה של טקסט כאשר התכונה מציינת שהערך הקלט פשוט שיש להזין לתוך השדה הטקסט תחת השם שם פרטי הינו הכנס שם פרטי או שהתכונה מציינת ב את הפורמט שהערך צריך להיות כאשר כלדוגמא בשדה `password` כתוב הזן סיסמה שתכיל אותיות וספרות.

הרמז התמציתי מוצג בשדה הקלט לפני שהמשתמש מזין את הערך לתוך שדה הקלט.

התכונה `placeholder` פועל על שדות פלט מסוגים הבאים כאשר השדות הן מוגדרים בתכונה `type` : `text`, `search`, `url`, `tel`, `email`, `password` .

התכונה זו של `placeholder` נתמכת ע"י הדפדפנים ובדפדפן IE החל מ-IE 10.

יתרונה של התכונה `placeholder`

התכונה `placeholder` הינה תכונה חדשה החל מ-HTML 5 כאשר יתרונה הינו כאשר על מנת ליצור רמז תמציתי בשדה הקלט של הטקסט לגבי מה הערך שיש להזין או באיזה פורמט יש להזין את ערכים בשדות הקלט של טקסט אין צורך ליצור סקריפטים על מנת לעשות זאת כפי שהיה קודם ל-HTML 5 אלא מספיק להשתמש בתכונה `placeholder` ב-HTML 5 על מנת ליצור זאת.

הסבר על תכונת `maxlength` באלמנט `<input>`

התכונה `maxlength` מציינת את המספר המקסימאלי של התווים המאופשרים להזנה בתוך השדה הקלט המוגדר באלמנט `input` כלדוגמא בשדה `password` התכונה `maxlength` מגבילה את אורך התווים בשדה זה של סיסמה ב- 10 תווים.

הערך המקסימאלי של מספר התווים בשדה קלט במצב ברירת מחדל הינו 524288 תווים

התכונה `maxlength` פועלת על שדות הקלט מסוגים הבאים כאשר השדות הן מוגדרים בתכונה `type`: `text`, `search`, `url`, `tel`, `email`, `password`.

התכונה `maxlength` לא מאפשרת כלל להזין תווים נוספים מעבר למה שהוגדר בתכונה.

הסבר על תכונת `minlength` באלמנט `<input>`

התכונה `minlength` מציינת את מספר המינימאלי של התווים ההכרחיים להזנה שיהיו בשדות הקלט המוגדרים באלמנט `input` כלדוגמא בשדה `password` התכונה `minlength` מגבילה את האורך המינימאלי של התווים הנדרש בשדה זה של סיסמה הינו 4 תווים.

התכונה `minlength` פועלת על שדות הקלט מסוגים הבאים כאשר השדות הן מוגדרים בתכונה `type`: `text`, `search`, `url`, `tel`, `email`, `password`.

התכונה `minlength` הינה תכונה חדשה החל מ-HTML 5 ונתמכת רק בדפדפן Chrome החל מגרסה 40 ומעלה

הסבר על תכונת `multiple` באלמנט `<input>`

התכונה `multiple` מציינת כאשר המשתמש יכול להזין יותר מערך אחד בשדה קלט המיוצג ע"י האלמנט `input`.

התכונה `multiple` הינה תכונה בוליאנית דהיינו שהערך הינו שם התכונה בעצמו או שלא צריך לכתוב כלל את הערך התכונה מאחר וערך התכונה הינו התכונה עצמה.

התכונה `multiple` פועלת על שדות קלט שהינם שדה קלט של `file upload` או שדה קלט של `email` דהיינו כאשר הערכים של `type` הינם: `type="file"` או `type="email"`

כאשר התכונה `multiple` פועלת בשדה קלט של `file` על מנת לבחור קבצים מרובים יש לחוץ הקליד `CTRL` או `SHIFT` כאשר בוחרים קובץ נוסף להעלות.

כאשר התכונה `multiple` פועלת בשדה קלט של `email` כל כתובת אימייל אתם צריכים להפריד עם פסיקים כלדוגמא: `gdgd@gmail.com`, fdd@gmail.com

התכונה `multiple` הינה תכונה חדשה החל מ-HTML 5 ונתמכת ע"י הדפדפנים ובדפדפן IE החל מ-IE10

הסבר על תכונת pattern באלמנט <input>

מהי תכונת pattern

תכונת pattern באלמנט input מפרטת את regular expression (ביטוי רגולרי) שמוגדר בתוך שדה הקלט כאשר הערך של השדה הקלט צריך להיות בפורמט המוגדר ב- Regular expression (ביטוי רגולרי) כך שהערך של שדה קלט המוגדר באלמנט input ייבדק ע"י הדפדפן על מנת לאמת האם הערך השדה עומד בפורמט המוגדר ב- regular expression (ביטוי רגולרי).

תכונת pattern הינה תכונה חדשה החל מ- HTML 5 נתמכת ע"י הדפדפנים למעט הדפדפן safri ובדפדפן IE זה נתמך רק מ- IE10 ומעלה.

תכונת pattern הינה פועלת רק על שדות קלט מסוגים הבאים: text,search,url,tel,email ו- passwrod.

כלדוגמא:

```
<form action="demo_form.asp">
Country code: <input type="text" name="country_code"
pattern="[A-Za-z]{3}" title="Three letter country code">
<input type="submit">
</form>
```

בדוגמא אנחנו רואים שהערך של התכונה pattern הינו כאשר regular expression (ביטוי רגולרי) של שדה הקלט של קוד מדינה כאשר נתוני השדה יכולים להיות מוצגים באורך של 3 תווים בלבד וכאשר התווים מורכבים מאותיות גדולות או אותיות קטנות.

Regular expression (ביטוי רגולרי) שבתכונה pattern מבוסס על regular expression (ביטוי רגולרי) שבג'אווה סקריפט.

חשיבות נתינת מידע למשתמש בשימוש בתכונה pattern

כאשר אתם עושים שימוש בתכונה pattern מומלץ לעשות שימוש בתכונה title כאשר ייצור tooltip שיציג את הפורמט המבוקש בשדה כלדוגמא באמצעות ה- tooltip אנו נציג כאשר הסמן ירחף על גבי שדה קלט זה כאשר השדה צריך להכיל כ-10 תווים של אותיות גדולות או קטנות. ובכך יסייע למשתמש לדעת מהו הפורמט של נערכים ששדה קלט שאותו הוא רוצה להזין.

לדוגמא:

```
<form action="demo_form.asp">
Country code: <input type="text" name="country_code"
```

```
pattern="[A-Za-z]{3}" title="Three letter country code">
<input type="submit">
</form>
```

בדוגמא אנחנו רואים שימוש בתכונה title כאשר מספק מידע למשתמש ששדה country code מורכב מ-3 תווים של אותיות.

מהו regular expression (ביטוי רגולרי)

regular expression (ביטוי רגולרי) הינו תבניות טקסטואליות שמסייעות לכם להתאים טקסט הרצוי כאשר לדוגמא שדה קלט יכיל ערכים בתבנית המוגדרת באמצעות regular expression (ביטוי רגולרי).

regular expression (ביטוי רגולרי) יוצר תבניות טקסטואליות שמסייעות לגבי מיקום הטקסט כיצד ימוקם כפי שרואים לדוגמא כאשר מספר טלפון מוגדר מיקומו של טקסט כאשר 3 מספרים ראשונים מייצגים מיקומו של הקידומת ולאחר מכן קבוצה של 4 מספרים המייצגים מיקום של חלק הראשון של מספר טלפון עצמו ולאחר מכן קבוצה של 3 מספרים המייצגים מיקום חלק האחרון של המספר טלפון.

regular expression (ביטוי רגולרי) יוצר תבניות טקסטואליות שמסייעות לגבי ניהול הטקסט בכך שניתן להתאים לטקסט הרצוי ואף להגדיר מיקומו של הטקסט.

ערכי regular expression בהקשר ל pattern

bracket(סוגריים מרובעות)

בסוגריים המרובעות נעשה שימוש על מנת להגדיר טווח של ערכי טקסט. כאשר בסוגריים מרובעות ישנם רצף של תווים ללא מקף כלדוגמא [abc] או [123] אזי המשמעות שערך הטקסט יכול להיות a או b או c וכך גם במספרים יכול להיות 1 או 2 או 3.

כאשר ערכים בסוגריים מרובעות מופיע מקף בין שני התווים כלדוגמא [0-9] כאשר המשמעות שניתן לעשות שימוש בטקסט של המספרים הכוללים את המספרים בין ל 0-9 ואף את המספרים 0 ו 9 - עצמם.

כאשר מופיע באותיות [a-z] כלומר הטקסט יכול הכיל את הערכים של אותיות הקטנות בין a ל-z הכולל את האותיות a ו-z עצמם. כאשר בסוגריים המרובעות שמים את האותיות a-z באותיות גדולות [A-Z] אזי כנ"ל הערכים כוללים את האותיות הגדולות ואף את האותיות הקצה שבטווח A ו-Z. אופן נוסף עם אותיות [A-z] או [a-Z] מציין שהטקסט יכול להכיל אותיות בין a ל-z בין אותיות גדולות ובין אותיות קטנות למעט A האות הראשונה כאשר היא גדולה ניתן להוסיפה לטקסט רק כגדולה ו-z כאשר היא קטנה רק ניתן להוסיפה לטקסט כקטנה וכך גם להפך כאשר בהתחלה אות קטנה ובסוף אות גדולה אזי יהיה ניתן להוסיף כנ"ל את הטקסט בטווח הערכים אך את האות הראשונה והאות האחרונה כפי הגודל שהם צוינו.

כאשר מופיעים בסוגריים המרובעות גם התווים הללו שהם אינם אלפאנומריים כמו הללו [!#\$%&'()*+,-./:;<=>?@^_`{|}~]. ושאר התווים שהם אינם אלפאנומריים מציין שניתן להוסיף לטקסט את אחד מהם או כמה מהם או כולם יחדיו מן הסמלים שהם אינם אלפאנומריים לטקסט שמשמעות כאשר הם מופיעים יחד עם ערכים אלפאנומריים וכאשר הם לא מופיעים עם ערכים אלפאנומריים אזי ניתן להוסיף לטקסט רק את הסמלים הללו המצוינים.

Quantifiers(מכמת)

CSS

פרק התחביר ורקע על CSS

תוכן:

1. מהו להיות CSS
2. למה להשתמש ב-CSS.
3. הסבר על CSS rules.
4. תחביר ה-CSS.

מהו להיות CSS

CSS הינו ראשי תיבות של Cascading Style Sheet בתרגום עברי גיליון סגנון מדורג. כאשר היא נחשבת שפת סגנון שמגדירה את תצוגת המסמך.

CSS מגדיר את תצוגת דף האינטרנט כלומר האופן שהאלמנטים שב-HTML יוצגו. כלדוגמא הפונטים, הצבעים, מרווחים, ועוד מאפיינים אחרים כאשר כולם יוצרים את מבנה התצוגה הייחודי של הדף אינטרנט.

למה להשתמש ב-CSS

בראשית נציין שעצם השימוש ב-CSS חסך בעיה גדולה כאשר במפרט של HTML 3.2 נוספו אלמנטים כ- font ותכונות כ- color דבר אשר התחיל להיות סיוט עבור מפתחי web. כשפיתוח של אתר אינטרנט גדול כאשר המידע על font - color שציינו למעל כדוגמא ושאר הדברים שהם אלמנטים של תצוגה ותכונות של התצוגה שהיו צריכים להוסיף לכל דף יחיד דבר אשר הפך את תהליך הפיתוח לארוך ויקר.

בעיה זו נפתרה החל מ-HTML 4 כאשר אז הכניסו לשימוש את ה-CSS כאשר חוסך הרבה עבודה כאשר על ידי CSS ניתן להגדיר לכל הדפים דברים שקשורים בתצוגה אחידה לכל הדפים במקום להוסיף מחדש בכל ודף בפני עצמו.

הסבר על CSS rules

ה-CSS rule הינם הכללים כאשר הם משמשים את אופן הפעולה שבו CSS משתמש באמצעותם לשייך לאלמנטים של HTML כללים איך הם יהיו מוצגים.

כלל ב-CSS (rule) הינו בעצם הכינוי לכל הדבר שבו אנו כותבים ב-CSS עם השיוך לאותו אלמנט של HTML שאנו משתמשים עמו להגדיר את תצוגתו בשימוש ב-CSS.

תחביר ה-CSS

מבנה התחביר של הכלל (rule) ב-CSS מורכב משני חלקים ה-selector ו-block declaration.

ה-selector מצביע על איזה אלמנט של HTML אנו רוצים לעצב כהצהרה.

ה-declaration block כאשר מכיל בתוך סוגריים מסולסלות declaration אחת או יותר כאשר מפרידים ע"י נקודה ופסיק כל declaration (הצהרה) נוספת או אף ישנו הצהרה אחת בלבד זה מסתיים בנקודה פסיק ;

כל declaration (הצהרה) מכילה property name (שם תכונה) ו-value (הערך) כאשר נקודתיים מפרידים בין ה-property name ל-value.

לדוגמא:

```
p {  
  font – family: Arial;  
}
```

בדוגמא זו ראינו שהאלמנט p שלמעלה משמש כ-selector שאותו אנו רוצים לעצב. ובסוגריים המסולסלים שהינם declaration block שבהם אנו מפרטים את הצהרה כשאנו רוצים לעצב את האלמנט p, בהצהרה אנו רואים את החלק הראשון שהינו property name המציין ששם התכונה אנו רוצים לעצב הינו font –family דהיינו איזה סוג של משפחת פונטים יוצגו באלמנט p כאשר לאחר הנקודתיים ישנו את הערך שמצביע שמשפחת הפונטים שיוצגו באלמנט p הם Arial .

Selector כאשר מצביע על איזה אלמנט מיושם ה-declaration (הצהרה) למטרת העיצוב אם ישנו יותר מ-selector אחד באופן זה נפריד בין ה-selector הנוספים בפסיק.

לדוגמא:

;


```
h1,h2{
  font-family: Arial;
}
```

בדוגמא ראינו שישנם שני selector שהינם h1 ו-h2 כאשר פסיק מפריד ביניהם.

פרק document tree

תוכן:

1. הסבר על document tree (עץ מסמכים)
2. הסבר על ancestor element
3. הסבר על descendant element
4. הסבר על parent element
5. הסבר על child element
6. הסבר על sibling element

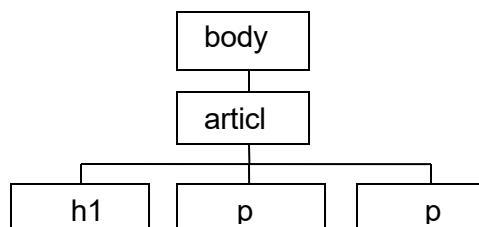
הסבר על document tree (עץ מסמכים)

בכל מסמכי HTML ישנו עץ, שניתן להגדירו כהיררכיה כאשר כל שלב של העץ שהינו מתואר באותו צורה כמו עץ משפחתי אנושי, עם ancestor (אב הקדמון), descendants (צאצאים), parents (אבות), children (ילדים), sibling (אחים). ה- rule ב- CSS מבוססים על document tree ועל ישנו חשיבות להבין את document tree וע"י כך נבין בקלות את selector ב- CSS ונוכל ליישם זאת בקלות.

הסבר על ancestor element

ancestor (אב קדמון) מתייחס לאיזשהו אלמנט שמכיל בתוכו אלמנטים שהינם Child element of child element בעץ או בהיררכיה דהיינו שאין בהכרח חיבור באופן ישיר בין ancestor element לאלמנטים השוכנים בתוכו. לדוגמא האלמנט body הינו ancestor element של כל האלמנטים הנמצאים בתוך body והוא מכיל גם אלמנטים שהם child element of child element כאשר אינם מחוברים אליו ישירות.

הדוגמא:



הסבר על descendant element

Descendants (צאצאים) מתייחס לאלמנטים שהינם שוכנים בתוך אלמנט כאשר בין שהם child של אלמנט או child element of child element ישירות ושאינם מחוברים ישירות בעץ או בהיררכיה דהיינו שאין בהכרח חיבור ישיר בין האלמנטים לאלמנט שהם שוכנים בתוכו. לדוגמה אלמנט article שמכיל בתוכו את האלמנט section שבתוכו שוכנים האלמנטים h2 ו-p.

הסבר על parent element

Parent element מתייחס לאיזשהו אלמנט שמכיל בתוכו אלמנט אחד או יותר כאשר הם מחוברים אליו ישירות בעץ או בהיררכיה. כלדוגמה האלמנט header כאשר מכיל בתוכו את האלמנט h1 כאשר לא שוכן בתוך אלמנטים האחרים המופיעים בתוכו אזי באופן האלמנט header נחשב כ- parent element של האלמנט h1.

הסבר על child element

Child element מתייחס לאיזשהו אלמנט אחד או יותר ששוכן בתוך אלמנט עם חיבור ישיר אליו בעץ או בהיררכיה. כלדוגמה האלמנט h1 שציינו למעלה ששוכן בתוך האלמנט header ולא שוכן בתוך אלמנטים אחרים המופיעים בתוך האלמנט header נחשב כ- child element של האלמנט header.

הסבר על sibling element

Sibling element מתייחס לאלמנטים שהינם child element ששוכנים יחד בתוך אלמנט עם חיבור ישיר אליו בעץ או בהיררכיה. כלדוגמה האלמנטים li ששוכנים בתוך האלמנט ol נחשבים כ- sibling element.

פרק ההורשה ב CSS

תוכן:

1. ההורשה ב- CSS

2 הסבר על parent element and child element

בהקשר להורשה.

3. איך ההורשה פועלת.

4. הורשת האלמנט body

5. הסבר מילת המפתח inherit

ההורשה ב-CSS

הורשה מתוארת כמצב כאשר אלמנטים של HTML יורשים מאפייני עיצוב שהינם ה-property מ-parent element (אלמנט אב) על ידי שאין ערך CSS ספציפי עבור child element, וכך ש-child element עשוי באותו מצב לרשת את הערך ה-CSS הנתון של parent element. כאשר ה-CSS עושה פעולה של הגדרת rule באלמנט אב שמכיל בתוכו אלמנטים כאשר אותם אלמנטים (child element) יורשים.

ההורשה היא פעולה מבורכת מחד ומאידך פעולה שאינה מבורכת כלל, מאחר וההורשה הינה מתודולוגיה בעלת עוצמה רבה כך שלעיתים קרובות לא מבינים אותה כראוי ביחס לגבי השפעת ההורשה על child element. וזה יכול לגרום לבלבול ע"י כאשר נתקלים ב-style sheet כפולים שמרכזים אותם ב-parent element תוצאה מכך שהם חלים אף על אותם אלמנטים ששוכנים בתוכו של אותו parent element שאיליו הם מוגדרים בחיבור ישיר או שלא ישיר בייחוד כאשר אתם עושים פעולת ניפוי הבאגים (debugging) ב-CSS שלכם וכך שלא מצליחים להבין איזה מ style sheet (ההצהרה של ה-rule) לא תקין ומשפיע על style sheet באלמנטים וגורם לבאגים ועל כן שימוש בהורשה כבר בתחילה שבמבצעים את יישום של אותם style sheet שהינם מאפייני העיצוב צריכים להיות מודעים לגבי שיוכם של אותם style sheet, כאשר איזה אלמנטים מושפעים מהם על מנת לדעת על איזה מאלמנטים שיוורשים את ה-rule כאשר ה-rule לא תקין ומשפיע על אחד מן האלמנטים שיוורש זאת באופן של תצוגה לא תקינה. אך הצד המבורך שבהורשה היא בכך שפעולה זו יכולה גרום לנו לאמץ הקטנת גודל של style sheet (מאפייני העיצוב) ע"י הקטנת היקף התיגו של style sheet בפעולה זו של הורשה ואף מאפשר בכך שינויים בקנה מידה גדול להרבה כללים של CSS עם מינימום עבודה ע"י שמשפיע בהורשה על אלמנטים רבים ששוכנים באותו parent element. באופן כללי הורשה תמיד מתרחשת בהקשר לאלמנטים של HTML שלכם שמושפעים ממנה, ובמרבית המקרים ההתערבות בהורשה היא חיונית רק כדי לשלוט על הערכים שמורשים בגלל סיבה ספציפית כלשהיא.

הסבר על parent element and child element

בהקשר להורשה

על מנת להבין את נושא ההורשה אנחנו נציין מהו parent element ו child element כאשר איזשהו אלמנט של HTML נחשב כ-parent והאלמנטים שהם מכילים בתוכם נחשבים כ-children. ההורה הוא בעלים של הילד ומעביר מה שהוא יודע או שהיא יודעת אל הילד. ב-CSS ההורשה פועלת בדרך דומה, מלבד שסגנון והערכים והעצות שהוא מעביר לא על חינוך או שאר ידע שקשור לבני אדם עצמם.

העברה באמצעות תגי HTML הינה ברורה כאשר איזשהו child element (כאשר כל האלמנטים של html הינם child element של האלמנט HTML שמכיל את כל האלמנטים) מתנהג כ-parent element ל-child elements אחרים ששוכנים בתוכו וכך הלאה, וכך הוא מפתח את היררכיית הכלה. היררכיית הכלה זו גם ניתן להתייחס אליה כעץ.

איך ההורשה פועלת

נמחיש את פעולת ההורשה עם אלמנט של `h1`. ה `h1` <h1> לדוגמא כאשר מסוגן עם rule (כלל) פשוט מאוד:

```
/* Top-level heading */
h1{
  color: #333;
}
```

הכלל הינו פשוט למדי ואתם בדיוק מקבלים את הכותרת שתוגש בצבע אפור כהה. כעת בואו ונניח שהתג ב-HTML נותן פקודה שמעט מן המילים צריך להיות מודגש בשימוש האלמנט ``

```
<h1>This is the greatest heading <em> in the world</em></h1>
```

בשלב הזה אין rule (כלל) מפורש ספציפי שמתקיים על מנת להשפיע על אלמנט ``. עקב כך, הטקסט בתוך האלמנט `` יירש את הצבע מן rule (כלל) `h1` (שזה parent element שלו) ויהיה עקב כך גם בצבע אפור כהה. על מנת לנטרל את ההורשה הצבע זו ניתן להגדיר את `` באופן פשוט בתוך (rule)style sheet

```
/* Make emphasized text shine brightly */
em {
  color:#F00;
}
```

כעת כל הטקסט המודגש לרובו של האתר יוגשו בצבע אדום, והאלמנט כבר לא יורש את הצבע שלו מאיזשהו parent element. נבחין שלולא אם היה מוגדר בתוך ה- selector `em` שלכם הצבע שהוגדר בתוכו, ההצהרה אחרת של הצבע שנמצאת ב- parent element עדיין תוריש ל- child element האלמנט `em` את הצבע שהוגדר ב selector שמייצג את parent element. כך אם אתם רוצים רק שהטקסט המודגש שלכם שיהיה פעמיים ה- font-size (גודל האות) יהיה בגודל השווה פעמיים הגודל אות הנוכחי ע"י שאנו מצהירים ש- font-size (גודל האות) ספציפי בתוך ה- rule (כלל) `em` שלכם. אם אתם מגדירים את ה- rule (כלל) הזה בתוך rule (כלל) `p`, כל הטקסט יגדל לגודל של פונט השווה לפעמיים הגודל הנוכחי.

הורשת האלמנט body

זאת המלצה לגבי עיצוב ב- CSS כאשר המלצה זו עדיין בתוקף שמתייחסת לכך שכל עיצובי CSS הרציניים מתחילים עם הצהרה לגבי אלמנט `body` בתוך החלק של style sheet הראשי. השימוש בהצהרה באלמנט `body` הינו יותר מאשר רק דרישה של דף HTML מעוצב היטב, זה גם משמש כ- parent element דהיינו ancestor של כל האלמנטים

שנמצאים בחלק החזותי של דף אינטרנט עמו בתבנית(ולא בתוך האלמנט head שעוסק במידע על גבי המידע וסעיפים אחרים שלא מוצגים ע"י הדפדפן) וכל האלמנטים הללו יכולים לרשת ממנו מסיבה זו.
לדוגמא:

```
<body>
<h1>Absolutely everything else!</h1>
<p>Yep, every visible element is contained within the body.</p>
<p>And so on.</p>
</body>
```

עקב כך לכן זה הגיוני שיש להגדיר את כל default (מצב של ברירת מחדל* ב-CSS בשימוש ה-body כ-selector הראשון בתחילת העיצוב של ה style sheet מאוחר יותר אנחנו נראה ש-selector של body יהיה בשימוש להגדיר את האלמנטים העיקריים של הדף כגון margin (שוליים) עבור הדף, וצבע רקע או tiled image (תמונה המשמשת כרקע) הפונט שמוגדר כברירת מחדל בדף וצבע הפונט וכו'.

לדוגמא:

```
/* Define all main values for the web site */
body {
margin:10px;
font-family:Helvetica,Arial,sans-serif;
background:#CCC;
color:#000;
}
```

כתוצאה הצהרות הללו כל כללים אחרים ב-CSS ירשו את הערכים אלא כן יצוין אחרת. כל הכותרות, הפסקאות, רשימות ואלמנטים אחרים יהיו מוגשים עם טקסט בצבע שחור(#000) בשימוש הפונט הראשון שזמין מאפשריות שמוצעות במחשב של המשתמש קצה מ-selector שהוגדר עבור אחד מ-child element מצוין אחרת, או ש-child element שוכן בתוך parent element נוסף (כגון עמודה או אלמנט של הכלה) שהערך הורשה של אותו אלמנט סותר את הערך שמוריש ה-body. נבחין שאיזשהו תכונות עיצוב לא יורשות מ-parent element התכונה background שנמצאת בדוגמא שציינו כאן (background: #CCC); רקע בצבע אפור בהיר, ה-child element לא יורש את זה למעשה, אבל הרקע של parent element מופיע כברירת מחדל. במילים אחרות ה-child element יורש את הערך של font color אבל זה לא צריך להניח שיש לו את אותו צבע הרקע מאחר שאינו יורש את הערך של צבע רקע(background)

הערה: כדאי לזכור שכל האלמנטים הרקע שבהם שקוף אלא אם כן אתם תציינו אחרת.

הסבר מילת המפתח inherit

מילת המפתח inherit נעשה בה שימוש להורשה של תכונות ל-child-element כאשר באותם תכונות לא מתקיים הורשה במצב ברירת מחדל על שתתקיים הורשה באותם תכונות צריך לשים כערך באותם תכונות את מילת המפתח inherit ואז תתקיים הורשה אך ההורשה תתקיים בתנאי כאשר אותם תכונות קיימות ומגודרות ב-parent element

אפשרות שימוש במילת המפתח inherit התקיים החל מ-CSS 3.

מילת המפתח inherit נתמכת בכל הדפדפנים אך באינטרנט אקספלורר החל מ-IE8

פרק ID ו class ב-CSS

תוכן:

1. מבוא

2. הסבר מהו id

3. יישום תכונת id ב-CSS

4. id בצירוף עם selectors

5. מתי להשתמש בתכונת id

מבוא

עד כה היה נראה שרק ב-selector הבסיסי ניתן לבצע עיצוב. כאשר ה-selector הבסיסי לוקח תג HTML בסיסי ומגדיר מחדש חלק או כל מאפייני עיצוב בברירת מחדל במטרה לעצב כל אלמנט ב-HTML כגון rule (כלל) המגדיר שפיסקאות יהיו בצבע

```
P{color:#f00;}
```

למרבה המזל, CSS מאפשר לכם להגדיר selectors מותאם אישית שמוכרים כ-selector ID ו-class selector. ID ו-Class מיושמים באלמנטים של HTML כתכונות פשוטות שמספקות שליטה הדוקה ויותר תשומת לב לעיצוב שלכם.

לאורך ההיסטוריה של התכונות class ו-id רבים מן המפתחים מבלבלים את id עם ה-class, או שנכשלים לנצל לממש את המטרה האמתית שלהם או פשוט משתמשים באחר דהיינו לדוגמא במקום class אני משתמש ב-id

הערה: בהגדרה פשוטה id נחשיב אותו כמספר זיהוי. מספר זיהוי שלכם שהינו ייחודי ולא ניתן לשתף אותו עם איזשהו אחרים. Class הינה שונה בכך שיכולים להיות הרבה

אנשים באותה כיתה זה בבית ספר, בחברה או בכל מקום שהוא. ובהקבלה בהקשר ל- id css יכול להופיע רק פעם אחת בכל דף, ואילו מחלקות ניתנים לשימוש למספר בלתי מוגבל של פעמים.

הסבר מהו id

id יכול להיות בשימוש רק פעם אחת בכל דף, וזה הינו מזהה ייחודי לאלמנט. בדרך כלל id הוא בשימוש עבור איזשהו אלמנט ייחודי בדף כגון header ו- main navigation ו- footer או מרכיבי מפתח אחרים של ממשק המשתמש.

יישום תכונת id ב- CSS

הדרך הנפוצה ביותר ליישום תכונת id הינו ביחס ל-HTML כאשר אנחנו משתמשים בתכונה id בתוך התג בפותח של האלמנט.

לדוגמא:

```
<p id="highlight">This paragraph has red text.</p>
<p id="default">This paragraph has dark gray text.</p>
```

פה בדוגמא יש לנו שני id שאחד שמו highlight ושני שמו default כאשר שניהם מיושמים בפיסקאות כאשר הם משמשים id באלמנט p שהינו האלמנט המגדיר שהתוכן שבו הינו פיסקה.

ב- CSS כאשר רוצים ליישם את האלמנט id למטרת עיצוב את אותו אלמנט עם id הייחודי אנחנו משתמשים ב- rule בתחילת ה- selector בתו סולמית # המציין שמודבר ביישום id ייחודי כאשר לאחר הסולמית אנו כותבים את הערך של השם של id ולאחר מכן יישום ה- rule באופן הדומה לאחרים.

כלדוגמא:

```
*/Define highlighted text/*
#highlight {
color:#F00;
}
*/Define default text/*
#default {
color:#333;
```

;

```
}
```

פה בדוגמא אנו רואים כהמשך לדוגמא שציינו בתחילה בפיסקאות את ה-id כאשר כאן אנו מיישמים זאת ב-rule ב-CSS.

Id בצירוף עם selectors

התכונה id יכול להיות מצורף עם סלקטורים אחרים כלומר עם שם אלמנט והתכונה id שלו יחדיו כסלקטור במבנה element#id באופן שאנחנו רוצים להגדיר שאלמנט עם ערך של id יוגדר העיצוב שלו שיוצג באופן שמוגדר ב-rule כפי שנסביר את זה בדוגמא:

```
/* Basic heading style */  
h2 {  
  color:#333;  
  font-size:16px;  
}
```

פה אנו רואים בדוגמא rule של CSS בסיסי שמגדיר את כל האלמנטים של h2 heading התוכן שלהם יהיה בצבע אפור כהה וגודל הפונט יהיה 16 פיקסלים. האופן הגדרה בסיסי זה טוב כאשר אנו רוצים לשייך לשימוש בכל האלמנטים של h2, אבל כאשר שאנו רוצים ש-h2 הראשי בדף שלנו (כותרת של מאמר) יהיה מודגש עם צבע שונה. באופן זה אנחנו קוראים ל-rule חדש כאשר הסלקטור מוגדר במבנה של אלמנט יחד עם הערך של id שיראה כך element#id בדוגמא זו:

```
/* Adjust the color of h2 when used as a title */  
h2#title {  
  color:#F00;  
}
```

כפי שאנחנו כעת רואים בדוגמא ש-rule החדש שיישמנו יעקוף את הצבע שמוגדר כברירת המחדל באלמנט h2 שהינו צבע אפור כהה color:#333; עם שימוש בצבע אדום שמוגדר בו color:#f00; כאשר האלמנט h2 מזוהה עם id שהינו title ב-HTML. ה-rule החדש לא מגדיר מחדש את גודל הפונט, כך שגודל הפונט יוצג בו ללא שינוי. בפשטות מדובר שהוספנו id ייחודי בדף לאלמנט h2 שנראה כך בדוגמא: <h2 id="title">Title Of My Article</h2> זכרו שערך title של id הינו מזהה ייחודי, כך שלא ניתן להשתמש בו שוב בתוך אותו תבנית של הדף. שאר האלמנטים של h2 שללא id יוצגו באופן שהם מוגדרים כברירת המחדל ב-rule שמגדיר את האלמנט h2 כיצד יוצג בדף.

מתי להשתמש בתכונת id

;

מתי שאנחנו רוצים להשתמש במרכיב אחד בודד בדף באופן ייחודי כאשר בכל דף יכול להיות מעוצב ע"י כל id ייחודי, ועקב כך id צריך להישמר כייחודי, לשימוש במרכיבי הדף כגון header (החלק הפותח של הדף) או sidebar (החלק הצדדי של הדף) או ניווט הראשי או חלק הסיום של הדף. השימוש ב-id יוצר אפשרות של סריקת הקוד קלה יותר שניתן להתחקות אחר הפעולות הקוד כאשר כל id מציין תחום תוכן ייחודי או איזור מיוחד בדף, אמנם גם מספק גמישות גדולה יותר ליישום של CSS שהינו מורכב ע"י הציון באופן ייחודי של כל מרכיבים.

פרק הסבר על pseudo-class ו-pseudo element

תוכן:

1: הסבר מהו pseudo-class selector

2. הסבר מהו pseudo-element selector

הסבר מהו pseudo-class selector

pseudo-class (מחלקה מדומה) מתייחס למילת מפתח שמוסיפים ל-selector כאשר מציין מצב מיוחד של אלמנט שנבחר ב-selector עבור עיצוב האלמנט זה. לדוגמא hover: הינו pseudo-class המשמש למטרת יישום של עיצוב כאשר המשתמש מרחף עם הסמן של העכבר על גבי האלמנט המצוין ב-selector.

התחביר של pseudo-class יראה כך:

```
selector:pseudo-class {  
  property: value;  
}
```

הסבר מהו pseudo-element selector

pseudo-element (אלמנט מדומה)

פרק מידות (measurement) ב- CSS

תוכן:

1. מבוא ל- css measurements

2. הסבר על Absolute Measurements

מבוא ל- css measurements

ב-rules (כללים) ב- css ניתן על ידי הצהרות לשלוט על אורך הטקסט, מרווח בטקסט, רוחבי הגבולות, מרווח בין איזשהו אלמנט לאלמנט השכן, ע"י הגדרת המידות הנצרכות לכם באמצעות יחידות מידה עם ערכים מספריים.

ערכים המספרים של המידות יכולים להיות מספרים שלמים או מספרים עשרוניים (דצימליים).

ישנם שתי אפשרויות של ערכי מידות שזמינות ב- CSS שאחת הינה absolute (מוחלט) ושנייה relative (יחסי).

הסבר על Absolute Measurements

Absolute measurements מתייחס לערכים מוחלטים שהינם מידות ספציפיות וקבועות. הם נותנים לכם בדיוק את המידה הממשית וכך על פי זה ניתן לשלוט על התצוגה של הדף אינטרנט.

השימוש ב- absolute measurements (מידות מוחלטת) הינו רק כאשר הדפדפן יודע בדיוק כמה פיקסלים ישנם במסך וכמה גודל המסך, שלולא כן ה- absolute measurement ייתן תוצאה לא רצויה שלא יראה טוב בכך שיציג בגודל שאינו נראה באפן תקין. ועל כן פחות מומלץ להשתמש ב- absolute measurement מאחר שאיננו תמיד יודעים בדיוק את גודל המסך כאשר משתנה ממסך למסך וכך מוצג על כן השימוש ב- absolute measurement זה רק בתקנים בעלי רזולוציה גבוהה ששם בעיקר כמדפסות ששם לרוב הרזולוציה גבוהה. כאשר הרזולוציה גבוהה הערך המדידה המדויק שניתן לא מוצג באופן לא תקין מחמת הרזולוציה הנרחבת בה מוצג כאשר אז אפשרויות התצוגה יותר נרחב משא"כ ברזולוציה נמוכה שערך מדויק לא מותאם יראה לא ברור מחמת אפשרויות התצוגה הנמוכות.

מדפסת בעלת רזולוציה גבוהה נחשבת כשמתחילה ברזולוציה של כ- 300dpi. ומסמך מחשב נחשב בעל רזולוציה גבוהה כשמתחיל ב- 200 dpi.

הערכים המוחלטים הם: סנטימטר, מילימטר, אינטש, נקודות, pica, פיקסלים.

1. in מתייחס לאינטש..
2. cm מתייחס לסנטימטר.
3. mm מתייחס למילימטר.
4. pt מתייחס לנקודות.
5. px מתייחס לפיקסלים.
6. pc מתייחס ל-pica.

הסבר על המידות המסורתיות in ו-cm, mm, בשימוש כיחידות מידה ב-CSS

אתם יכולים להשתמש בסנטימטר ובאינטש או במילימטר כיחידות מידה לגודל הפונטים, אבל זה לחלוטין לא מעשי לעשות בזה שימוש. הסיבה לכך נתאר זאת שנניח שיש לנו דף אינטרנט שמוצג במסך מחשב שלכם וגם במקרן. אינטש אחד במסך שלכם יראה כעשרה אינטשים במקרן. לכן השימוש ביחידות מידה שמשמשים בהם בדברים פיזיים אין להם משמעות עבור הדף אינטרנט. השימוש בהם טוב לגודל פונטים הינו לגבי הדפסה כאשר אתם רוצים להדפיס איזשהו דבר ויש לכם ידע לגבי איך המדפסת מוגדרת טכנית לגבי גודל ההדפסה כאשר בהדפסה צריך להגדיר את המידה באופן מדויק אך באופן זה יותר טוב להשתמש בתוכנות שנעשה בהם שימוש להדפסה כמעבד התמלולים וורד מאשר HTML.

הסבר על שימוש במידה pc בשימוש כיחידת מידה ב-CSS

אתם יכולים להשתמש ב-pica (פיקה) שהינה יחידת מידה טיפוגרפית לגודל פונטים אך מאותו סיבה שפירטנו קודם לכן זה לא מומלץ.

הסבר על שימוש במידה pt בשימוש כיחידת מידה ב-CSS

אתם קרוב לוודאי מכירים points(נקודות) בעיבוד תמלילים שהינה יחידת מידה טיפוגרפית המוודדת את גודלו של הפונט. אתם יכולים להשתמש בקיצור pt להצביע שהמידה נעשית ביחידת מידה של points

כלדוגמא:

```
P {
  Font-size:14pt;
}
```

לרוע המזל points(נקודות) לא נחשב כיחידת מידה אפקטיבית לדפי אינטרנט. Points(נקודות) כקנה מידה אבסולוטי זה שימושי עבור הדפסה, אבל הם לא אמינים על גבי דפי אינטרנט בגלל שאתם לא יודעים מה הרזולוציה של המסך של המשתמשים. כלדוגמה מידה של 12 pt גודל הפונט שעשוי להיראות גדול יותר מהמידה שצוינה או קטן יותר משצוינה במסכי המחשב השונים.

הסבר על שימוש במידה px בשימוש כיחידת מידה ב-CSS

פיקסלים מתייחס לנקודות קטנות שעל המסך. אתם יכולים לעשות שימוש בפיקסלים כיחידת מידה לאורכים ב-CSS ואף יחידת המידה שבה נעשה השימוש הנפוץ ביותר באורכים ב-CSS

פרק בניית box model ב- CSS

תוכן:

1. מבוא ל box model
2. הסבר מהם התכונות border
3. כתיבת תכונת border באופן מקוצר
3. מהי תכונת padding.
4. מהי תכונת margin.
5. חשיבות רווח אנכי בין התיבות.
6. מימדים של content box (תיבת התוכן)

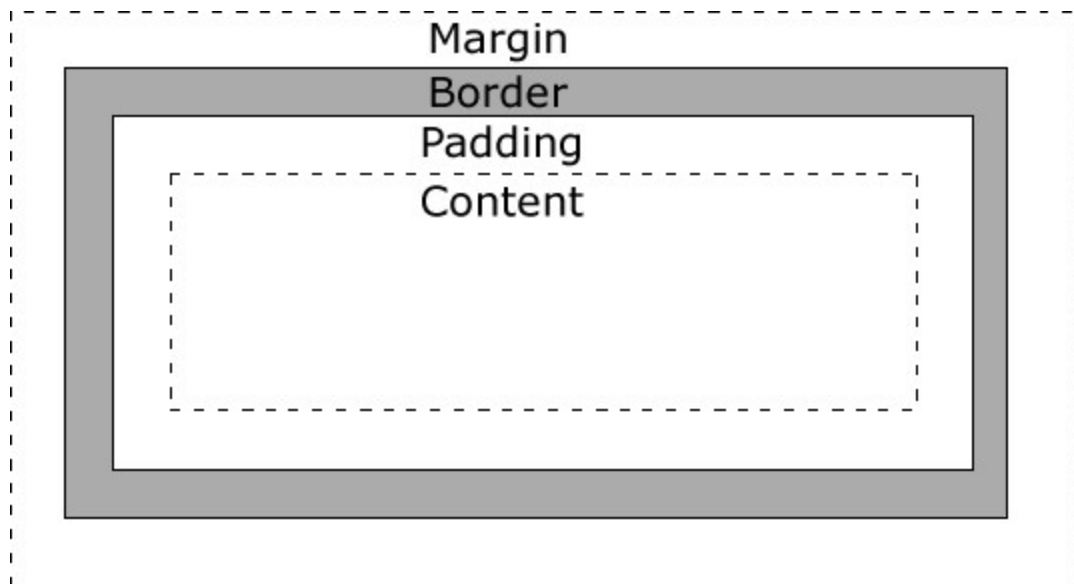
מבוא ל box model

במסמך כל אלמנט מוצג כתיבה מלבנית. פעולת הגדרת הגודל והתכונות עיצוב של התיבה (האלמנט) כמו הצבע של, רקע, והיבט הגבולות של התיבה ומיקומה מבוצע ע"י תוכנת rendering engine שבתוך הדפדפן שמגדירה את התצוגה שזאת המטרה של תוכנה זו שבדפדפן.

ב- CSS ההתייחסות לאותם תיבות מלבניות נעשה בשימוש box model שמחמת כך נקרא בשמו הזה ואף הינו מושג חשוב מאוד בכך שהוא מגדיר איך האלמנטים ימוקמו בחלון הדפדפן. ה-box model מתאר את התוכן של המרחב שבו שוכן האלמנט.

כל תיבה המייצגת אלמנט ישנם 4 מאפיינים שהינם margin (שוליים), border (גבול), padding (מרווח) content (תוכן).

דוגמא למודל:



הסבר מהם התכונות border

ה-`border` (גבול) ישנו בכל תיבה אפילו אם אינך יכול לראות זאת. כאשר הגבול הינו בעצם יוצר את הפרדה מן פאות של תיבה אחת לפאות של תיבות האחרות הסובבות אותו

התכונות (`border` property) מאפשרת לכם לציין את התצוגה שאיך יראה הגבול של התיבה המייצגת את האלמנט.

ה-`border` (גבולות) לא משנים באופן ממשי את מתווה הדף, אבל הם מוסיפים הפרדה ויזואלית שמיושמת, ובייחוד כאשר המתווה הדף שלכם יותר מורכב.

רשימת שלושת התכונות border שאתם יכולים לשנות

1. `border-color` – התכונה שמצביעה על הצבע שהגבול צריך להיות.
2. `border-style` - התכונה שמצביעה באיזה סגנון צריך שהגבול יראה בסגנון כלדוגמא: `solid`, או-`dashed` וכן עוד ערכים. כערכים הללו `solid`, או-`dashed` וכן עוד ערכים.
3. `border-width` – התכונה שמצביעה על רוחב שהגבול צריך להיות.

הסבר על תכונת border-color

תכונת `border-color` מאפשרת לכם לשנות את הצבע של הגבול הסובב את התיבה וניתנת ליישום על כל האלמנטים.

לדוגמא: p

```
border-color : #ff0000;  
}
```

בדומא אנחנו רואים הגדרת צבע של הגבול כאשר כוללת את כל הגבול הסובב את התיבה(האלמנט) מכל צדדיה כאשר הצבע המדובר בדוגמא של הגבול של האלמנט p הינו אדום.

כתיבת התכונה border-color צריכה להיכתב לאחר התכונה border-style

כתיבת תכונת border-color באופן פרטני

אתם יכולים להשתמש בתכונת border-color באופן פרטני שמאפשר לכם לשנות את צבע הגבול בצד אחד שאתם מפרטים באופן פרטני השימוש הוא כך:

border-top-color כאשר מציין שינוי של צבע הגבול העליון בלבד.
border-right-color כאשר מציין שינוי של צבע הגבול הימיני בלבד.
border-bottom-color כאשר מציין שינוי של צבע הגבול התחתון בלבד.
border-left-color כאשר מציין שינוי של צבע הגבול השמאלי בלבד.

ובכך ניתן ליצור צבע גבול שונה לכל אחד מן ארבע הגבולות הסובבים את האלמנט או שינויים אחידים בעלי אותו ערכים אך שנעשה באופן פרטני בכל צד.

כתיבת border-color באופן מקוצר

אתם יכולים להשתמש בתכונת border-color באופן מקוצר כאשר ניתן לבצע זאת באופן זה כאשר נותנים ערך אחד של צבע כאשר יגדיר שזהו הצבע של ארבע גבולות הסובבים אותו התחביר יראה כך:

```
selector{  
border-color: value of color;  
}
```

דרך שנייה באופן מקוצר כאשר מפרטים את כל צבעים של ארבע גבולות בתכונה border-color התחביר יראה כך

```
selector{  
border-color: top right bottom left;  
}
```

סדר צבעי הגבולות שהם מפורטים יחדיו הם של צבע הגבול למעלה(top) לאחמ"כ(right) ולאחמ"כ (bottom) ולבסוף(left).

דרך שלישית הוא שנותנים שני ערכים כאשר הערך הראשון מציין את צבע הגבולות top ו- bottom והערך השני מציין את צבע גבולות ה- left ו- right

אפשרויות לערכים של צבע הגבול

אפשרויות לערכים של צבע הם או ע"י:

שם צבע: כאשר כותבים את שם הצבע של הגבול.

ערך הקס דצימלי : כאשר כותבים את הערך הקס דצימלי של הצבע כלדוגמא #000000 המייצג את הצבע שחור.

ערך RGB: כאשר כותבים את הערך ה- RGB של הצבע כלדוגמא: (0,0,0) המייצג את הצבע שחור.

ערך RGBA: כאשר כותבים את ערך ה- RGBA של הצבע כלדוגמא: (0,0,0,1) המייצג את צבע שחור.

ערך HSL: כאשר כותבים את ערך ה- HSL של הצבע כלדוגמא: (0%,0,0%) המייצג את צבע שחור.

ערך HSLA: כאשר כותבים את ערך ה- HSLA של הצבע כלדוגמא: (1,0%,0,0%) המייצג את צבע שחור.

ערך transparent: הינו ערך של צבע שהינו שקוף

כלל ליישום התכונה זו:

את התכונה של border-color לא ניתן ליישם זאת לבד אלא אם כן ישנו קודם את התכונות border-style ו- border-width .

הסבר על תכונת border-style

תכונת border-style מאפשרת לכם להגדיר את כיצד יהיה מעוצב הקו של הגבול וניתנת ליישום על כל האלמנטים.

לדוגמא:

```
p {  
    style-color : solid;  
}
```

בדוגמא אנחנו רואים הגדרת הסגנון של הקו של הגבול כאשר כוללת את כל הגבול הסובב את התיבה(האלמנט) מכל צדדיה כאשר סגנון הגבול המדובר בדוגמא של הגבול של האלמנט p הינו solid.

רשימת הערכים של border-style

- none** זו ערך של התכונה זו במצב ברירת מחדל שלא מוגדר ערך של עיצוב הינו none כאשר המשמעות שאין שום קו גבול ונתן לציין זאת אף במפורש כ- border-style:none; ואף הערך של התכונה border-width באופן זה יהיה 0.
- solid** ערך של התכונה שמשמעותו בתרגום המילה solid הינו מוצק כאשר התכונה

- מציינת שמדובר בקו של גבול המעוצב כקו מוצק שהינו קו יחיד וישר.
3. **dotted** ערך של התכונה שמציין שקו הגבול יהיה סדרה של נקודות שאף יתכן שיהיו סדרה של ריבועים אם תגדירו את הרוחב של הגבול כ- 2px.
 4. **dashed** ערך של התכונה שמציין שקו הגבול יהיה סדרה של קווים קצרים דהיינו קו מרוסק.
 5. **double** ערך של התכונה שמציין שקו הגבול יהיה שני קווים מוצקים יחד עם השימוש התכונה של border-width זה יוצר את הסכום של שני הקווים והמרווח ביניהם.
 6. **groove** ערך של התכונה שמציין שקו הגבול יהיה קו עם הצללה שמדמה אותו כחוקק לתוך הדף.
 7. **ridge** ערך של התכונה שמציין שקו הגבול יהיה קו עם הצללה שמופיע בולט על גבי הדף באופן תלת מימדי .
 8. **inset** ערך של התכונה שמציין שקו הגבול יוצר תיבה עם הצללה באופן שמדמה את זה לכפתור לחוץ באופן תלת מימדי כמו דבר שנוצר מקנבס שהינו תלת מימדי.
 9. **outset** ערך של התכונה שמציין שקו הגבול יוצר תיבה עם הצללה שנראית באופן המדמה לכפתור בולט מן הדף עצמו כמו דבר שנוצר מקנבס והינו תלת מימדי .
 10. **hidden** ערך של התכונה זו מציין שאין קו גבול בדומה לתכונה none ואף הערך של התכונה border-width באופן זה יהיה 0. ההבדל שה- hidden הוא נועד לשימוש עבור border-conflict resolution for table elements

כתיבת border-style באופן מקוצר

אתם יכולים להשתמש בתכונת border-style באופן מקוצר כאשר ניתן לבצע זאת באופן זה כאשר נותנים ערך אחד של סגנון כאשר יגדיר שזהו הסגנון הקו של ארבע גבולות הסובבים אותו התחביר יראה כך:

```
selector{
border-style: value of border- style-;
}
```

דרך שנייה באופן מקוצר כאשר מפרטים את כל הסגנונות של ארבע גבולות בתכונה border-style התחביר יראה כך

```
selector{
border-style: top right bottom left;
}
```

סדר סגנונות הגבולות שהם מפורטים יחדיו הם של סגנון הגבול למעלה(top) לאחמ"כ(right) ולאחמ"כ (bottom) ולבסוף(left).

דרך שלישית הוא שנותנים שני ערכים כאשר הערך הראשון מציין את סגנון הגבולות top ו- bottom והערך השני מציין את סגנון גבולות ה- right ו- left

כתיבת תכונת border-style באופן פרטני

אתם יכולים להשתמש בתכונת border-style באופן פרטני שמאפשר לכם לשנות את סגנון קו הגבול בצד אחד שאתם מפרטים באופן פרטני השימוש הוא כך:

border-top-style כאשר מציין שינוי של סגנון קו הגבול העליון בלבד.

border-right-style כאשר מציין שינוי של סגנון קו הגבול הימיני בלבד.
border-bottom-style כאשר מציין שינוי של סגנון קו הגבול התחתון בלבד.
border-left-style כאשר מציין שינוי של סגנון קו הגבול השמאלי בלבד.

ובכך ניתן ליצור סגנון קו גבול שונה לכל אחד מן ארבע הגבולות הסובבים את האלמנט או שינויים אחידים בעלי אותו ערכים אך שנעשה באופן פרטני בכל צד.

הסבר על תכונת border-width

תכונת border-width מאפשרת לכם להגדיר את רוחב של קו גבול שלכם. יחידת המידה שלרוב נעשה בה שימוש כערך זה פיקסלים.

כלדוגמא:

```
p}  
border-style : solid;  
border-width : 4px;  
}
```

בדוגמא אנחנו רואים שקו הגבול בפסקאות יוצג בסגנון solid ורוחב קו גבול יהיה 4 פיקסלים.

הערכים של התכונה border-width לא יכולים להיות נתונים באחוזים, למרות שאתם יכולים להשתמש באיזשהו יחידת מידה absolute או יחידת מידה relative או אחד מן הערכים הללו:

1. **thin** – המציין שקו הגבול שיוצג יהיה דק כאשר הינו פחות ממעובי שמוגדר כברירת מחדל.

2. **medium** המציין שקו הגבול שיוצג יהיה בעובי המוגדר כברירת מחדל.

3. **thick** המציין שקו הגבול שיוצג יהיה בעובי יותר מהעובי המוגדר ברירת מחדל

4. **initial** המציין שקו הגבול יוצב בעובי המוגדר כברירת מחדל בדפדפן

שלושת הערכים שמופרטים מעל thin, medium, thick ב-CSS עצמו אין שום איזכור לקנה מידה מדויק של עובי קו הגבול ומידת העובי תלוי באמצעות איזה דפדפן הוא מוצג.

כתיבת תכונת border-width באופן פרטני

אתם יכולים להשתמש בתכונת border-width באופן פרטני שמאפשר לכם לשנות את רוחב קו הגבול בצד אחד שאתם מפרטים באופן פרטני השימוש הוא כך:

border-top-width כאשר מציין שינוי של רוחב קו הגבול העליון בלבד.
border-right-width כאשר מציין שינוי של רוחב קו הגבול הימיני בלבד.
border-bottom-width כאשר מציין שינוי של רוחב קו הגבול התחתון בלבד.
border-left-width כאשר מציין שינוי של רוחב קו הגבול השמאלי בלבד.

ובכך ניתן ליצור רוחב קו גבול שונה לכל אחד מן ארבע הגבולות הסובבים את האלמנט או שינויים אחידים בעלי אותו ערכים אך שנעשה באופן פרטני בכל צד.

כתיבת border-width באופן מקוצר

אתם יכולים להשתמש בתכונת border-width באופן מקוצר כאשר ניתן לבצע זאת באופן זה כאשר נותנים ערך אחד של רוחב כאשר יגדיר שזהו רוחב הקו של ארבע גבולות הסובבים או ארבע אותו התחביר יראה כך:

```
selector{  
border-width: value of border- width-;  
}
```

דרך שנייה באופן מקוצר כאשר מפרטים את כל רוחב של ארבע גבולות בתכונה border-width התחביר יראה כך

```
selector{  
border-width: top right bottom left;  
}
```

סדר עובי הגבולות שהם מפורטים יחדיו הם של עובי הגבול למעלה(top) לאחמ"כ(right) ולאחמ"כ (bottom) ולבסוף(left).

דרך שלישית הוא שנותנים שני ערכים כאשר הערך הראשון מציין את עובי הגבולות top ו-bottom והערך השני מציין את עובי גבולות ה- right ו- left

כתיבת תכונת border באופן מקוצר

בתכונת border ניתן ליישם את כל התכונות של border הכוללות border style ו- border width ו- border-color בשימוש רק בתכונה border כאשר בתחילה אנו מציינים זאת בסדר הבא כאשר בתחילה הערך של border-style ולאחמ"כ הערך של border-width ולבסוף הערך של border-color כאשר התחביר יראה כך לדוגמא

```
P{  
border: solid 2px #000000  
}
```

בדוגמא זו ראינו ששינוי כל בתכונות של border נעשה באופן מקוצר כאשר אנו בתחילה border-style שהינו solid בדוגמא ולאחמ"כ border-width שמצוין בדוגמא ברוחב 2px ולבסוף מוגדר border-color שדוגמא מצוין בערך הקס דצימלי שצבע הגבול הינו שחור.

כתיבת תכונת border באופן פרטני

אתם יכולים להשתמש בתכונת border- באופן פרטני שמאפשר לכם לשנות את הגבול בצד אחד בכך שאתם מפרטים באופן פרטני בשימוש שהוא כך:

border-top- כאשר מציין שינוי של קו הגבול העליון בלבד.
border-right כאשר מציין שינוי קו הגבול הימני בלבד.
border-bottom כאשר מציין שינוי קו הגבול התחתון בלבד.
border-left כאשר מציין שינוי של קו הגבול השמאלי בלבד.

ובכך ניתן ליצור רוחב קו גבול שונה לכל אחד מן ארבע הגבולות הסובבים את האלמנט או שינויים אחידים בעלי אותו ערכים אך שנעשה באופן פרטני בכל צד.

מהי תכונת padding

תכונת (property) padding (מרווח) מאפשרת לכם לציין כמה מרווח צריך להופיע בין התוכן של האלמנט והגבול שלו. וחשיבות שימוש בתכונה כך שיוצר מרווח בין האלמנט לתכונה הופך את התוכן לקריא יותר.

לדוגמא:

```
td {  
  Padding: 5px;  
}
```

הערך של התכונה מצוין לרוב כפי שבדוגמא מצוין בפיקסלים אך ניתן לציין זאת אף באחוזים או ב-em. כאשר נעשה שימוש באחוזים זה מציין שגודל המרווח הוא בערך של אחוזים ביחס לחלון הדפדפן או גם של תיבה באופן שהיא מוכלת בתוך תיבה אחרת ואז מיוחס האחוזים ביחס לתיבה שבה היא מוכלת כלדוגמא אם יצוין הערך באחוזים 10% אז בכל צד שבתיבה יהיה מרווח של 5% מן הגבול שבחישבו כולל של המרווח זה הינו כ- 10%

בתכונה padding אינה יורשת כ- child element מאלמנטים האחרים שבהם היא שוכנת כ- parent element כאשר בתוכם מוגדר התכונה padding את ערכי התכונה הזו כך שצריך להגדיר עבור כל אלמנט את התכונה padding. כך כאשר לדוגמא כאשר אתם מגדירים באלמנט body את התכונה padding עם ערך כלדוגמא 40px זה לא מתקיים בשאר האלמנטים השוכנים בתוכו באוטומטי אלא אם כן אנחנו מפרטים בתוך אותו אלמנט ששוכן בתוכו את התכונה padding עם ערך של מילת המפתח inherit שרק אז אנחנו יכולים לרשת את אותו ערך שמצוין ב- parent element

כתיבת התכונה padding באופן פרטני

- התכונה padding ישנם אפשרות של פירוט הצדדים באופן פרטני שהן:
1. padding-top המתייחס למרווח העליון בין התוכן של האלמנט לגבול העליון.
 2. padding right המתייחס למרווח בצד ימין בין התוכן של האלמנט לגבול הימני.
 3. padding bottom המתייחס למרווח התחתון בין התוכן של האלמנט לגבול התחתון.
 4. padding left המתייחס למרווח בצד שמאל בין התוכן של האלמנט לגבול השמאלי.

ובכך ניתן ליצור מרווח שונה לכל אחד מן ארבע הצדדים הסובבים את האלמנט או שינויים אחידים בעלי אותו ערכים אך שנעשה באופן פרטני בכל צד.

כתיבת התכונה padding באופן מקוצר

אתם יכולים להשתמש בתכונה padding באופן מקוצר כאשר יכולים לתת ערך אחד שיציג מרווח אחיד לכל הצדדים לדוגמא:

```
Td {  
  padding: 5px;  
}
```

בדוגמא אנחנו רואים מרווח אחיד של 5 פיקסלים בין התוכן בתא הטבלה לגבול מכל ארבעת צדדיו.

אופן שני הינו כתיבה של כל ארבע צדדים באופן מקוצר בסדר הבא top (למעלה) right (ימין) bottom (למטה) left (שמאל) התחביר יראה כך:

```
selector {  
  padding: top right bottom left;  
}
```

דרך שלישית הוא שנותנים שני ערכים כאשר הערך הראשון מציין את padding ה-top ו-bottom והערך השני מציין את padding ה-right ו-left

מהי תכונת margin

תכונת margin (property) (שוליים) שולטת על רווח בין התיבות.

לדוגמא:

```
p {  
  margin: 10px;  
}
```

הערך של התכונה מצוין לרוב כפי שבדוגמא מצוין בפיקסלים אך ניתן לציין זאת אף באחוזים או ב-em.

בתכונה margin אינה יורשת כ- child element מאלמנטים האחרים שבהם היא שוכנת כ- parent element כאשר בתוכם מוגדר התכונה margin את ערכי התכונה הזו כך שצריך להגדיר עבור כל אלמנט את התכונה margin. כך כאשר לדוגמא כאשר אתם מגדירים

באלמנט body את התכונה margin עם ערך כלדוגמא 40px זה לא מתקיים בשאר האלמנטים השוכנים בתוכו באוטומטי אלא אם כן אנחנו מפרטים בתוך אותו אלמנט ששוכן בתוכו את התכונה padding עם ערך של מילת המפתח inherit שרק אז אנחנו יכולים לרשת את אותו ערך שמצוין ב- parent element

אם תיבה אחת יושבת בתוך ראש של תיבה אחרת אז באופן זה השוליים נעלמים כאשר המשמעות שהשוליים הגדולים יותר של התיבה מבין שתי השוליים הקיימים שהינה התיבה שהגדולה יהיה בשימוש להציגו כשוליים ויחושב והקטן ביותר מבין השוליים שהינו בין התיבה שמוכל בתוך התיבה עצמה לתיבה שבו הוא מוכל הדפדפן יתעלם ממנו.

כתיבת התכונה margin באופן פרטני

התכונה margin ישנם אפשרות של פירוט הצדדים באופן פרטני שהן:

1. margin-top המתייחס לשוליים העליונים בין התיבה לתיבה אחרת שמעליה.
2. margin-right המתייחס לשוליים בצד ימין בין התיבה לתיבה אחרת שמימינה.
3. margin-bottom המתייחס לשוליים התחתונים בין התיבה לתיבה אחרת שמתחתיה.
4. margin-left המתייחס לשוליים בצד שמאל בין התיבה לתיבה אחרת שמשמאלה.

ובכך ניתן ליצור שוליים שונים לכל אחד מן ארבע הצדדים הסובבים את התיבה או שינויים אחידים בעלי אותו ערכים אך שנעשה באופן פרטני בכל צד.

כתיבת התכונה margin באופן מקוצר

אתם יכולים להשתמש בתכונה margin באופן מקוצר כאשר יכולים לתת ערך אחד שיוציג מרווח אחיד לכל הצדדים לדוגמא:

```
Td {  
  margin: 5px;  
}
```

בדוגמא אנחנו רואים שוליים אחידים של 5 פיקסלים בין התיבה המייצגת את תא הטבלה מכל ארבעת צדדיה לתיבות האחרות הסובבות אותה. אופן שני הינו כתיבה של כל ארבע צדדים באופן מקוצר בסדר הבא top (למעלה) right (ימין) bottom (למטה) left (שמאל) התחביר יראה כך:

```
selector {  
  margin: top right bottom left;  
}
```

דרך שלישית הוא שנותנים שני ערכים כאשר הערך הראשון מציין את margin ה- top ו- bottom והערך השני מציין את margin ה- right ו- left

חשיבות רווח אנכי בין התיבות

התכונות padding ו-margin הם מועילות מאוד בהוספת מרווחים בין מבחר הפריטים שבדף.

המעצבים מתייחסים למרווח בין הפריטים שבדף כחלק ריק בדף. מאחר נניח ששיש לכם גבול סביב התיבה, אתם לא תרצו שהטקסט יגע בגבול מאחר שאז זה יהפוך את קריאת הטקסט קשה לקריאה מחמת שזה נוגע בגבול ועל כן חשוב שיהיה מרווח בין הגבול הטקסט. או נניח שיש לכם שתי תיבות ששוכנות זה לצד זה כאשר כל הגבולות הם באותו צבע אז באפן אתם לא בהכרח רוצים שהתיבות יגעו בפאות של תיבה מאחר וזה ייצור קו שיראה פעמיים כקו עבה יחד עם התיבה שמולו על כן יש ליצור מרווח בין שתי התיבות.

אם השוליים התחתיים של איזשהו יגעו בשוליים בראש של תיבה אחרת שמתחתיה, הרי שהדפדפן יגיש לתצוגה את מרווח בין התיבות באופן שונה מאשר אתם מצפים שיראה. וזה יציג רק את השוליים הגדולים יותר מבין שתי השוליים הקיימים. ואם השוליים יחדיו הינם באותו גודל הוא יציג רק אחד מהשוליים.

מימדים של content box (תיבת התוכן)

עד עכשיו ראינו שהגבול (border) הוא סובב סביב כל תיבת התוכן והמרווח (padding) שמופיע בתוך הגבול של תיבת התוכן, ושוליים (margin) שנמצאים מחוץ לגבול של תיבת התוכן כעת נתעסק בתכונות שמאפשרות לכם לשלוט על המימדים של תיבת התוכן (content box)

הסבר התכונות height (גובה) ו-width (רוחב)

במצב ברירת מחדל התיבה נמדדת לפי גודל האם היא גדולה מספיק להחזיק את התוכן שלה. על כן התכונות height (גובה) ו-width (רוחב) מאפשרות לכם להגדיר את הגובה והרוחב של התיבות תוכן.

מרבית הדרכים הפופולאריות לציין את הגודל של התיבה הינו בפיקסלים, אחוזים או em. באופן מסורתי משתמשים בפיקסלים כיחידת מידה הפופולארית לציין גודל התיבה בגלל שפיקסלים מאפשרים למעצבים לשלוט באופן מדויק על הגודל התיבה. כאשר אתם משתמשים באחוזים כיחידת מידה המציינת את גודל התיבה הינו ביחס למידת הגודל של חלון הדפדפן, או כאשר התיבה סגורה בתוך תיבה אחרת שבה היא מוכלת אז אחוזים הם ביחס מיחידת המידה של גודל תיבת שבה התיבה מוכלת.

כאשר אתם משתמשים ביחידת מידה em כיחידת מידה המציינת את גודל התיבה באופן זה היחידת מידה מתבססת על גודל הטקסט בתוך התיבה מאחר ומדובר ביחידת מידה טיפוגרפית. לאחרונה המעצבים מתחילים להשתמש בשונה מהשימוש המסורתי ביחידת המידה לציין גודל התיבה בפיקסלים ומשתמשים ב-em או אחוזים ויחידות מידה נוספות במטרה לנסות ליצור עיצוב שיהיה גמיש בכך לגודלי מסכים שונים

לדוגמא:

```
div.box {
  height: 300px;
  width: 300px;
  background-color: #bbbbaa;
}
p {
  height: 75%;
  width: 75%;
  background-color: #0088dd;
}
```

בדוגמא אנחנו רואים באלמנט div במחלקה box אנחנו רואים שנעשה שימוש לציין את גובה ורוחב התיבה בפיקסלים כשר מצוין בדוגמא 300 פיקסלים וצבע רקע אדום אפרפר. ולגבי האלמנט p המציין פיסקא שמתייחס לתוכן של פיסקה של טקסט שבתוך התיבה משתמשים ביחידת המידה לציין את גודל שטח הפיסקה של הטקסט יתפוס בתוך התיבה הינו 75 אחוזים בגובה ורוחב התיבה כאשר בפיקסלים מדובר על 225 פיקסלים בגובה ורוחב התיבה ובצבע רקע כחול בשטח בו כתוב הפיסקה.

הסבר תכונות min-width ו- max-width

התכונות min-width ו- max-width מאפשרות לכם לציין רוחב המינימאלי ורוחב המקסימאלי של תיבת התוכן. עיצוב הדף מתרחב או מתכווץ על מנת להתאים למידת הגודל המסך של המשתמש. ועל מנת ליצור עיצובים של דפים באופן זה משתמשים בתכונה min-width שמציינת את הגודל הקטן ביותר של התיבה ברוחבה שתוכל להיות מוצגת בחלון של דפדפן שהוא צרף ובאמצעות התכונה max-width אנו מגדירים את רוחב המקסימאלי של התיבה שבה היא יכולה להשתרע בחלון הדפדפן כאשר הוא נרחב.

התכונות הללו מועילות מאד כך שהן מבטיחות שהתוכן של הדף יהיה קריא (בייחוד במסכים קטנים בטאבלטים ומובייל) דוגמא לתועלת שבשימוש בתכונות הללו כך שאתם משתמשים בתכונה max-width להבטיח שהשורות של הטקסט לא יופיעו באופן נרחב יותר כאשר מוצגים כשורות ארוכות כתוצאה מהשפעה מכך שחלון הדפדפן הינו רחב וכך שזה הופך את זה קשה לקריאה מחמת ששורות ארוכות מאוד קשות לריאה וכך בשימוש בתכונה min-width זה מבטיח גם ששורות הטקסט לא יופיעו שהם צרות באופן שיהיה לא קריא.

אתם יכולים לבדוק את השפעה המעשית ע"י שאתם מגדילים את החלון של הדפדפן ולהיווכח לתוצאה מה קורה כאשר אתם מקטינים את חלון הדפדפן או שאתם מקטינים את חלון הדפדפן.

דוגמא לשימוש בתכונות הללו:

```
<style type="text/css">
```

```
.container {max-width:1200px; width:100%; background-color: red;}
.container h2 {max-width:400px; width:100%; background-color: blue; }
.container p {max-width:900px; width:100%; background-color: green;}
</style>
```

בדוגמא אנחנו רואים שרוחב מותאם לגודל מלא של חלון הדפדפן אך אם החלון הדפדפן גדול מערך שבתכונה max-width אז יוצג בערך המקסימאלי שבתכונה max-width, אך באופן שהחלון הדפדפן בגודלו קטן מהערך שבתכונה max-width יוצג בגודלו המלא של החלון הדפדפן.

```
<style type="text/css">
.container {min-width:700px; width:100%; background-color: red;}
.container h2 {min-width:600px; width:100%; background-color: blue; }
.container p {min-width:500px; width:100%; background-color: green;}
</style>
```

בדוגמא אנחנו רואים שרוחב מותאם לגודל מלא של חלון הדפדפן כאשר הרוחב הצר ביותר שיוכל להיות מוצג הינו כערך המינימאלי המוצג בתכונה min-width.

יש לציין לגבי התאמה למובייל וטאבלטים כאשר המסכים קטנים יותר לא ניתן להסתפק רק בשימוש בתכונות הללו, מאחר וכל הגבלה יותר מותאם לחלון הדפדפן אלא צריך תכונות נוספות המתאימות לתצוגה במובייל.

לגבי התכונה min-width אין לשים ערך מידת אורך absolute כפיקסלים במידת width כאשר גם במידת min-width מצוין במידת אורך absolute- כפיקסלים כלדוגמא:
.container {min-width:1200px; width:1400px}

הסיבה שאין להשתמש באופן שמצוין בדוגמא מאחר כאשר בתכונה width מצוין כבר מידת אורך קבועה מראש כפי מובא בדוגמא למשל 1400px כך שאין השפעה לערך של התכונה min-width המצוין בדוגמה כ- 1200px. על כן יש להשתמש בתכונה width הערך של אחוזים ולגבי התכונה min-width בשימוש בערך של מידת אורך של פיקסלים שהינה מידת אורך absolute שאז יוצרת השפעה ממשית שזה האורך המינימאלי ולא לעשות שימוש בה בערך של אחוזים כאשר אף בתכונה width מאחר ואף פעם לא ייתן את התוצאה המצופה של מידת הרוחב כאשר תמיד יוצג בערך של התכונה min-width שהינו פחות מהתכונה של width.

בתכונה max-width כאשר הערך התכונה יותר מן הערך התכונה width ביחידת המידה absolute כפיקסלים הרי באופן זה הערך ברירת מחדל הינו כערך התכונה width, כך שהדפדפן יציג ברוחב כערך התכונה width.

שימוש בתכונה max-width בתמונות

בתכונה max-width כאשר משתמשים בו לגבי הצגת התמונה מומלץ להשתמש בערך של אחוזים כאשר הערך המצוין יהיה כ-100% כאשר זהו הערך המקסימאלי שבו התמונה תהיה

מוצגת בחלון הדפדפן לרוחב כל פני החלון כך שזהו האופן הטוב ביותר שהתמונה תהיה ניתנת לתצוגה באופן ברור כאשר התמונה תוצג לכל רוחבה וזאת מאחר שהדפדפן מציג כברירת מחדל את רוחב התמונה המצוין וכאשר התמונה לא ניתנת לתצוגה לכל רוחבה - viewport אז בכך הערך של 100% בתכונה max-width פותר את הבעה הזו וכך משפיע שהתמונה תוצג לכל viewport.

הסבר תכונות min-height ו- max-height

התכונות min-height ו- max-height מאפשרות לכם לציין את הגובה המינימאלי וגובה מקסימאלי. כך באותו דרך שאתם השתמשתם הגבלה של הרוחב בתכונות min-width ו- max-width כך נעשה שימוש בתכונות min-width ו- max-width על מנת להתאים את גובה הדף. לדוגמא:

```
.container {min-height 400px; height:400px; background-color: red;}  
.container h2 {max-height:300px; width:100px; background-color: blue; }  
.container p {min-height:100px; height:200px background-color: green;}
```

הסבר התכונות של overflow

התכונה overflow מורה לדפדפן מה לעשות אם התוכן המוכל בתוך התיבה גדול יותר מאשר התיבה עצמה. וכך נותן פתרון למצב שלא ניתן ליצור תיבה גדולה שתכיל את כל התוכן שהיא אמורה להכיל בגלל מגבלות מבנה מתווה הדף אינטרנט ע"י כך שהתכונה overflow ניתן כמה פתרונות עפ"י אותם ערכים הקיימים בתכונה זו.

רשימת הערכים של overflow

hidden הינו ערך התכונה כאשר מסתיר את התוכן הנוסף שחורג מן התיבה **visible** הינו ערך התכונה כאשר מאפשר להציג את התוכן שחורג מחוץ לתיבה באופן כמצב של ברירת מחדל שבו כל התוכן החורג מן התיבה מוצג כאשר החריגה מן התיבה נראית. אך לא מומלץ להשתמש בתכונה זו איפה ששינוי חריגה של התוכן מן גבולות התיבה מאחר וזה יכול לפגוע במתווה הדף.

scroll הינו ערך של התכונה היוצר הסתרה של התוכן החורג מן גבולות התיבה אך מאפשר להציג את אותו תוכן חבוי לקריאה ע"י שהתכונה מגדירה scrollbar (פס גלילה) כאשר ניתן באמצעותו לגלול ולראות את אותו תוכן הפנימי החבוי וזה ע"י פס גלילה לאורך התוכן חורג מן הגובה התיבה או פס גלילה לרוחב כאשר התוכן חורג מן רוחב התיבה. **auto** הינו ערך של התכונה המציג בדומה לערך scroll עם פס גלילה אך ההבדל ביניהם כאשר הערך auto מציג עם פס גלילה רק כאשר התוכן חורג מן התיבה משא"כ עם scroll מציג באופן זה גם תוכן שאינו חורג מן התיבה.

inherit הינו ערך המציג הורשה של ערך של התכונה workflow שמוגדר parent element שב האלמנט אותו שעליו מחילים את הערך ההורשה שוכן ונעשה שימוש ע"י ערך זה מאחר והתכונה לא יורשת במצב ברירת מחדל מ- parent element.

הסבר התכונה workflow-x

התכונה workflow-x מבצעת בדיוק אותו דבר כפי שהתכונה workflow מבצעת עם אותם ערכים אך ההבדל ביניהם כאשר התכונה workflow-x פועלת רק על תוכן שחורג באופן אופקי מן התיבה דהיינו לרוחב התיבה. זהו תכונה שבוצעה החל מ-CSS 3

הסבר התכונה workflow-y

התכונה workflow-y מבצעת בדיוק אותו דבר כפי שהתכונה workflow מבצעת עם אותם ערכים אך ההבדל ביניהם כאשר התכונה workflow-y פועלת רק על תוכן שחורג באופן אנכי מן התיבה דהיינו לאורך שהינו גובה התיבה. זהו תכונה שבוצעה החל מ-CSS3

פרק links (לינקים)

תוכן:

JAVA SCRIPT