

# HTML5 התגיות החדשות

# Structure and Meaning

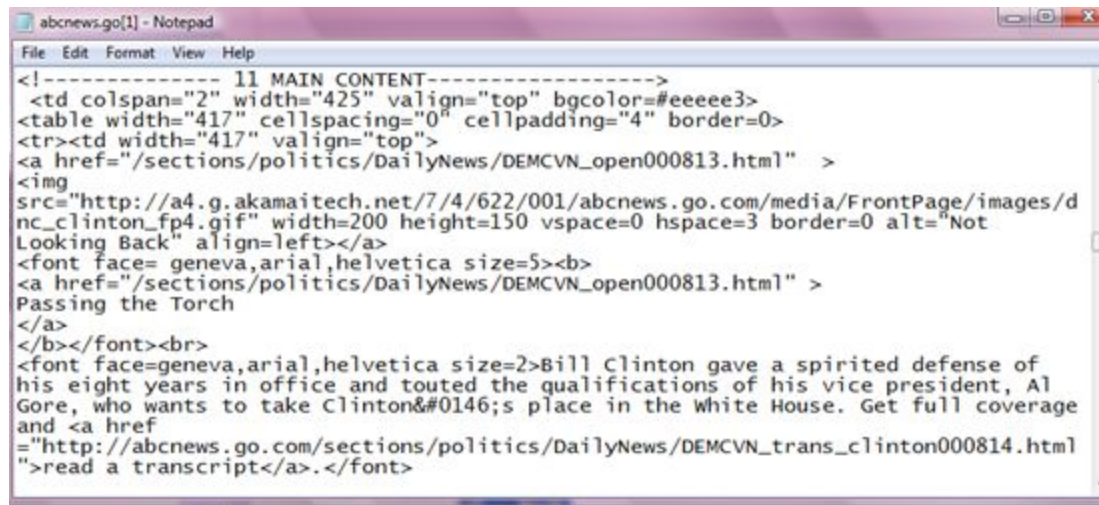
נכתב ונערך על ידי יונית רוש

# מסמך עם מבנה נכון והפרדת משמעות מעיצוב

- כמו שהיסודות של בניין חשובים, כך חשובים יסודות בניית הקוד.
- ככל שהאינטרנט התפתח, כך החלו להשתמש ב-HTML למטרות של תצוגה.
- במקום להשתמש בתגיות כותרת, החלו להשתמש בתגיות font ותגיות b, כדי ליצור את הוויזואליות הרצויה.
- טבלאות מידע הפכו להיות כלי ליצירת עימוד layout של דף, במקום דרך להצגת מידע.
- מהר מאוד איבדו עמודי ה-HTML משמעות והפכו להיות מערבולת של תגיות פונט וטבלאות.

# מסמך עם מבנה נכון והפרדת משמעות מעיצוב

- דוגמא לשימוש בתגיות פונט ובולד ליצירת כותרות ושימוש בטבלאות ליצירת layout באתר חדשות אמריקאי בשנת 2000.



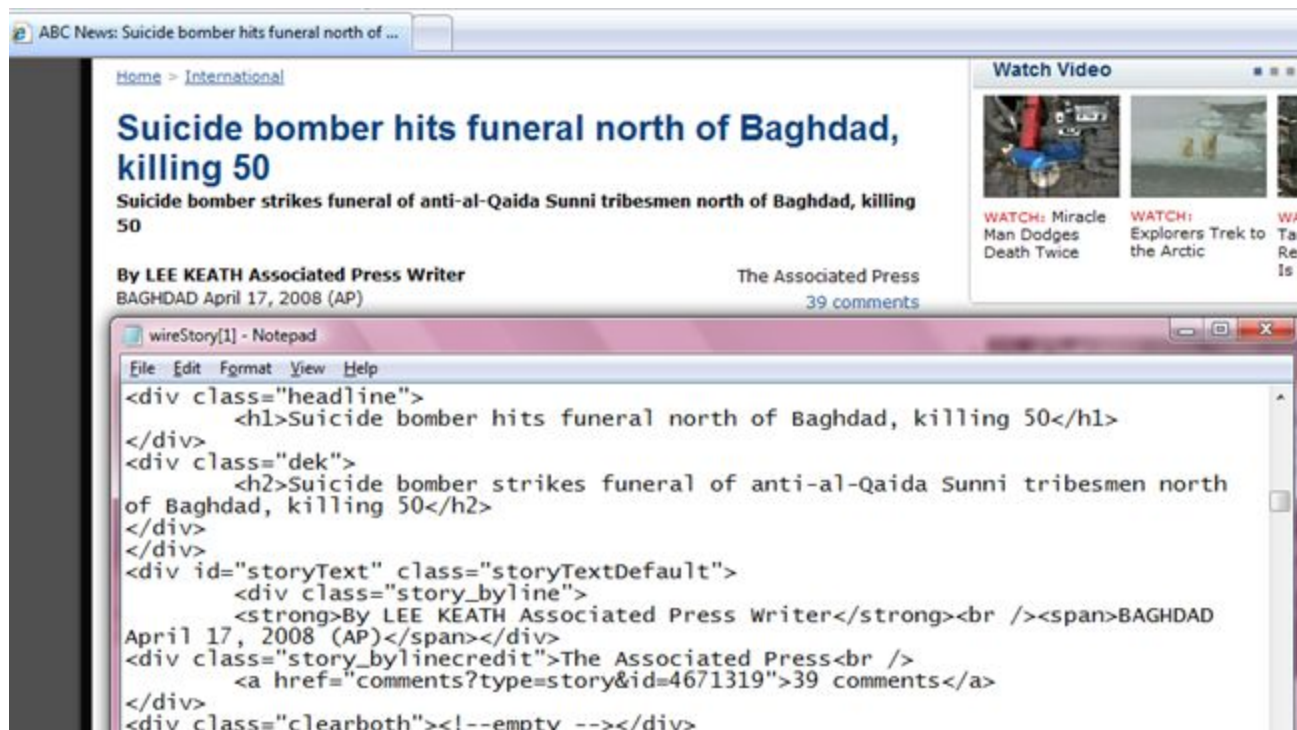
```
abcnews.go[1] - Notepad
File Edit Format View Help
<!------- 11 MAIN CONTENT----->
<td colspan="2" width="425" valign="top" bgcolor=#eeeeee3>
<table width="417" cellspacing="0" cellpadding="4" border=0>
<tr><td width="417" valign="top">
<a href="/sections/politics/DailyNews/DEMCVN_open000813.html" >
</a>
<font face= geneva,arial,helvetica size=5><b>
<a href="/sections/politics/DailyNews/DEMCVN_open000813.html" >
Passing the Torch
</a>
</b></font><br>
<font face=geneva,arial,helvetica size=2>Bill Clinton gave a spirited defense of
his eight years in office and touted the qualifications of his vice president, Al
Gore, who wants to take Clinton's place in the White House. Get full coverage
and <a href
="http://abcnews.go.com/sections/politics/DailyNews/DEMCVN_trans_clinton000814.html
">read a transcript</a>.</font>
```

# מסמך עם מבנה נכון והפרדת משמעות מעיצוב

- עם השנים הקוד הפך להיות פחות קריא ויותר מורכב ותצוגתי
- הכלים שהומצאו כדי להתמודד עם כמות התגיות חסרות המשמעות – במקום להקל, הוסיפו תגיות משלהם.
- ואז הגיע CSS – גיליונות סגנון מדורגים.
- עם CSS אפשר היה לשלוט על מראה הדף מחוצה לו ולהפריד בין הצד התצוגתי לבין התוכן.
- תגיות של תצוגה כמו תגית ה- font יצאו משימוש ועימוד הדף נשלט על ידי CSS במקום על ידי טבלאות.
- HTML חזר להיות פשוט ובעל משמעות.
- מפתחי אתרים החלו להשתמש בתגיות על פי המשמעות שלהן ולא על פי איך שהן מציגות את המידע.

# מאפייני תגיות tags attributes

- דוגמא לקוד באותו אתר חדשות אמריקאי בשנת 2006.
- הוא בנוי נכון וקל להבנה. שיפור משמעותי מהקוד הקודם.



# כתיבת קוד עם משמעות- היתרונות למתכנת

- קל לעבוד עם תגיות בעלות משמעות יותר מאשר עם תגיות תצוגה.
- קל לעבור על הקוד ולמצוא את המקום שבו אנחנו צריכים לעשות שינוי.
- מנועי החיפוש יודעים לזהות תגיות בעלות משמעות כגון כותרות וקישורים, ויודעים שכתרת ראשית חשובה יותר מכתרת משנית.
- ניהול ה- layout
- עבודה עם דאטה דינאמי וממשקי ניהול תוכן
- קוד נקי וללא זבל
- נגישות- לכלים שקוראים את התוכן (לעיוורים) או לבעלי מוגבלויות

# מבנה ותקניות

- מבניות **structure**:

- הדף חייב להתחלק לשני חלקים - body ו-head.
- שניהם חייבים להימצא בין תגי html
- ובהתחלה ה- DOCTYPE!

- **well-formed**:

- יש תגיות ריקות. יש תגיות שחייבות להיסגר. יש להקפיד על כך.
- **Properly nested**. התגיות הסוגרות חייבות להיות מקוננות בהתאמה לתגיות הפותחות.

- סוגי אותיות **case sensitive**. הקפדה על אותיות קטנות בלבד, ולא עירוב של השתיים.

- למעט יוצאי דופן, תכונה של מאפיין חייבת להתחם בגרשיים כאשר לפניה האופרטור שווה (=).

# הפרדת התוכן מהתצוגה

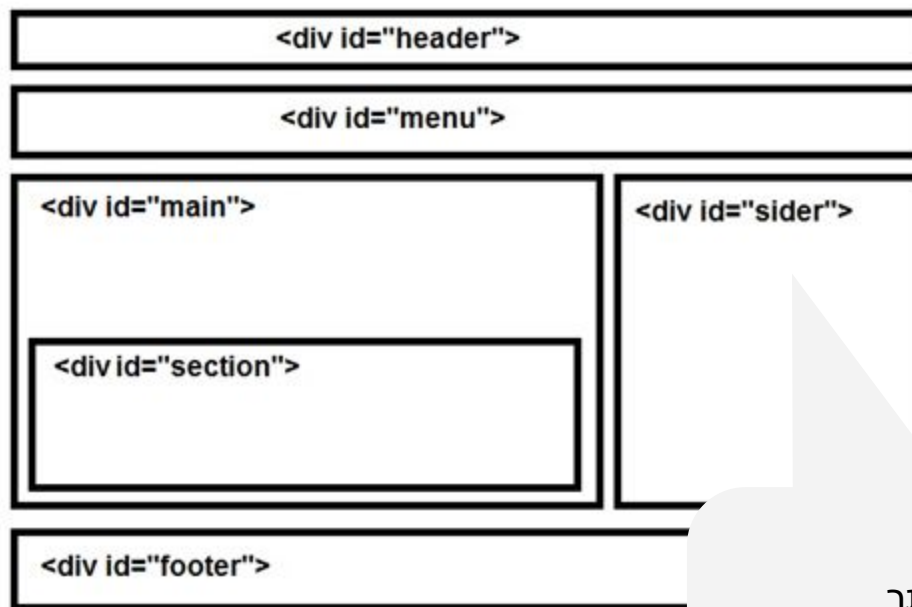
- Separating content and meaning (HTML) from presentation (CSS)

הערות	אלטרנטיבות	התג
משמעות: חשיבות מנועי החיפוש נותנים חשיבות למילים המופיעות בתוך התגית b	font-weight: bold	b
משמעות: ציטוט	font-style: italic	i
	h1, h2, font-size	big, small
	border-bottom, border-top	hr
יצא מהתקן, מקובל שרק לקישור יש קו מתחת		u
	text-align	center
<font color="red" size="4" face="Arial">do not use it this way</font>	CSS	font
	background-color	bgcolor
	background-image	background
	text-align	align



# DEFAULT PAGE STRUCTURE

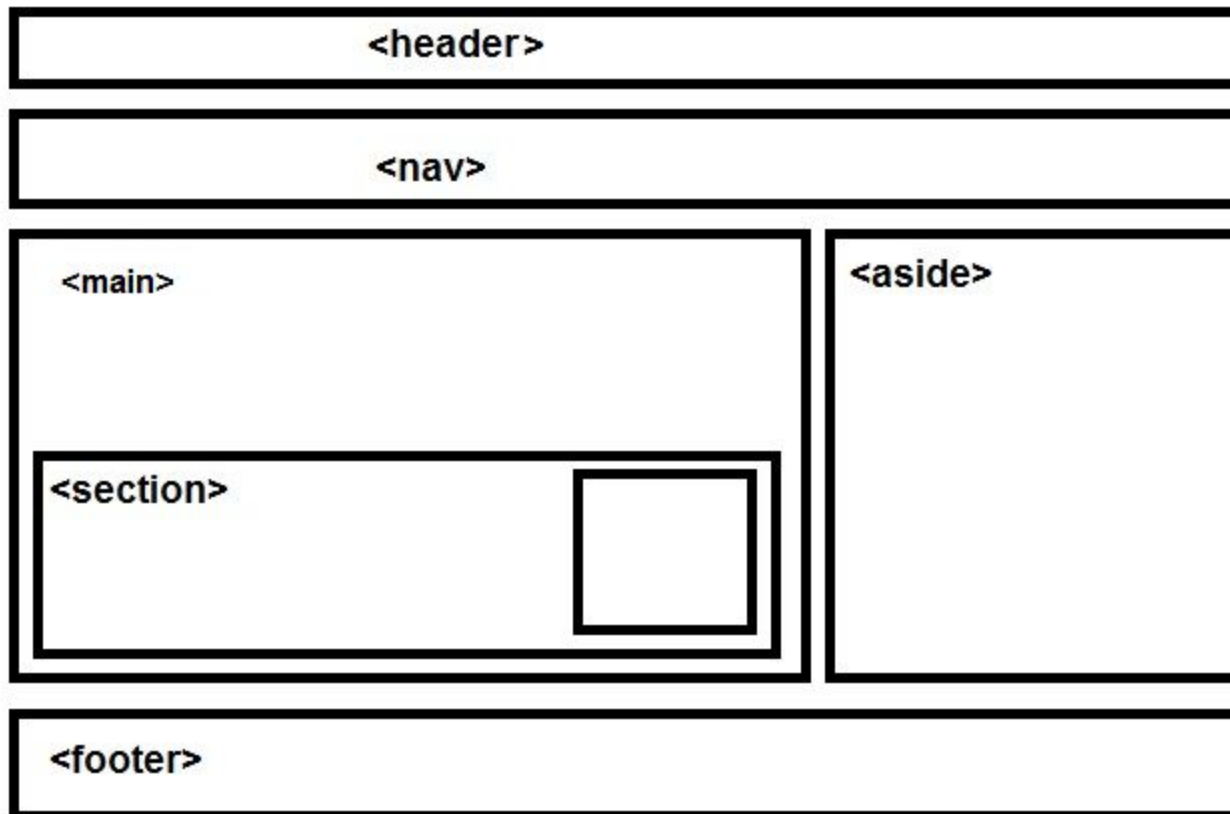
- מבנה סטנדרטי לעמוד ב- xhtml או ב- html4



`<div>`

אחת התגיות הנפוצות ביותר  
אין לה משמעות  
מגדירה אזור בדף, כמו קיר גבס,  
שאפשר למקם אותו, לשים עליו טפט,  
להעלים אותו ולצייר עליו גרפיטי

# השדרוג ב- HTML5



# HTML5 is all about MEANING

# תגיות חדשות - משמעות

---

## ● תגיות המבנה:

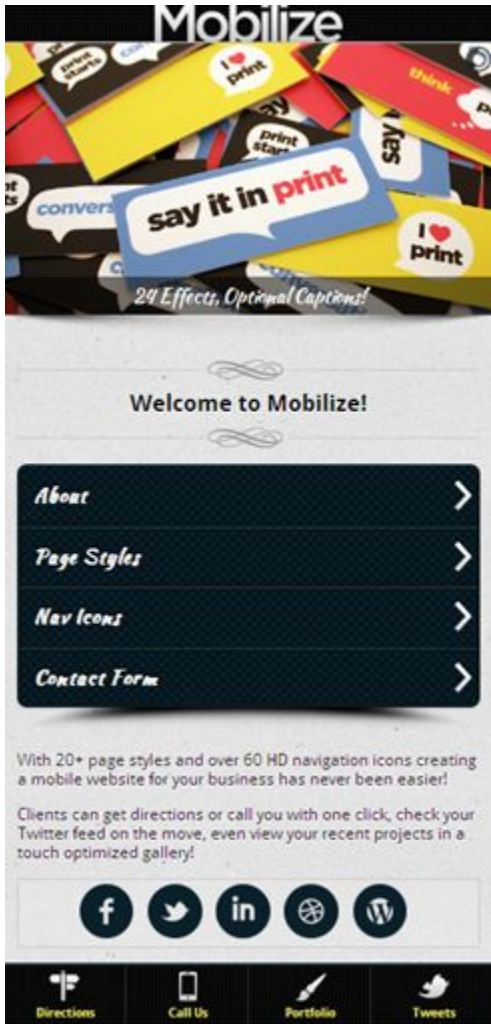
- header ○
- footer ○
- section ○
- article ○
- main ○
- nav ○
- aside ○

# תגית header

- האזור העליון של העמוד
- לעתים כולל בפנים תפריט עליון אופקי אם יש, באנר, לוגו.
- התגית יכולה להכיל כמעט כל תגית אחרת למעט header ו- footer
- מנועי החיפוש לומדים שתוכן שממוקם בתוך תגית זו חשוב.
- ניתן להגדיר יותר מתגית header אחת בעמוד
- דוגמא:

```
<body>  
  <header>  
    <h1>Our Company</h1>  
    <h2>21/9/2011</h2>  
  </header>
```

# תגית Footer



- החלק התחתון של העמוד.
- בדרך כלל משמש לקישורי הפוטר ולפרטי יצירת קשר וזכויות יוצרים.
- `<footer>footer links</footer>`
- תפריט הניווט המרכזי של האפליקציה בפוטר, בשילוב אייקונים.
- בדרך כלל, הפוטרים הללו נעלמים כשמתחילים לגלול, או נשארים קבועים במקומם, וכל התוכן נגלל מאחוריהם.

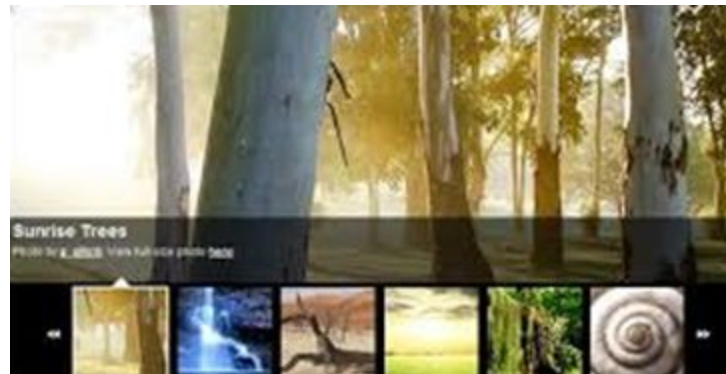
# תגית section

- אזור/חלק בדף.
- כל האלמנטים שבתוכו חייבים להיות קשורים זה לזה.
- אינו מוסיף לעיצוב, אלא למבניות ולמשמעות הקוד
- יכול לאגד בתוכו מספר תגיות תוכן, כולל article
- יש הרבה בלבול בינו לבין תגית article.
- התגית <section> יכולה להכיל את התגית <main> (התגית <article> לא).
- אם המטרה היא רק ליצור מלבן עם תמונת רקע, לדוגמא, וללא תוכן שיש לאגד, נשתמש בתגית ה-div

# תגית section

דוגמא ●

```
<section id="main_content">
  <h2>Our mission: expand your vision</h2>
  <p>In today's world, its very important to know what are your
best products..  
<br>and so on..  
</p>
</section>
```



דוגמה לאזור הכולל תמונה גדולה, תמונות קטנות, טקסט על כל תמונה, חיצים קטנים.  
כל אלו שייכים לאותו אובייקט ולכן יהיו כולם ב- section אחד.



# תגית article

- יחידת תוכן עצמאית
- כלל האצבע: היכולת של החלק הזה בדף להיות עצמאי בעמוד אחר.
- אם ניקח את תוכן התגית, ונמקם אותו בעמוד אחר, עדיין תהיה לתוכן משמעות.
- פוסט בפורום, טיזר למאמר, תגובה comment, חוות דעת, או כל תוכן עצמאי שאפשר לשלוח או לשתף רק אותו.
- אינה מוסיפה לעיצוב, אלא למבניות ולמשמעות הקוד
- יכולה לאגד בתוכה מספר תגיות תוכן, כולל section
- לא יכולה להכיל את התגית main

```
<article>  
  <p>posted by: Orit Levi</p>  
  <p>This was very interesting!</p>  
</article>
```

# תגית nav

- על פי ה- specification (מתורגם לעברית): "מייצג חלק של העמוד שיש בו קישורים לעמודים אחרים או לחלקים אחרים בעמוד זה."
- מיועד לאזורים שיש בהם קישורים חשובים מאוד, ולכן משתמשים בו עבור התפריטים הראשיים ועבור הפוטר.
- התגית יכולה להכיל כל תגית אחרת, גם כותרות.
- הטריק בתגית זו היא לדעת מתי להשתמש בה.
- ניתן להגדיר יותר מתגית nav אחת בעמוד

```
<nav>
  <ul>
    <li><a href="about.html">about</a></li>
    <li><a href="services.html">services</a></li>
    <li><a href="products.html">products</a></li>
    <li><a href="news.html">news</a></li>
    <li><a href="contact.html">contact us</a></li>
  </ul>
</nav>
```

# תגית nav

- פה לא נשתמש בה:

```
<aside>
  <h3 class="special">Special Offers</h3>
  <a href="read.pdf">read more..</a>
</aside>
```

# תגית main

- מיועדת לאזור התוכן המרכזי של הדף.
- התגית יכולה להופיע רק פעם אחת בדף.
- עד להגדרתה, היה מקובל ליצור את התוכן המרכזי של הדף בתוך תגית עם מאפיין id המרמז על משמעותה ומטרתה של התגית:  
<section id="main">
- מאחר וב- html5 יש חשיבות גדולה למשמעות התגיות ולמטרתן בעמוד, נולדה התגית <main>.
- אינה יכולה להיות מוכלת בתוך:
  - <article>
  - <aside>
  - <footer>
  - <header>
  - <nav>
- לא מומלץ לכלול בתוכה אלמנטים בהיררכיה גבוהה, שחוזרים על עצמם באתר, כמו תפריט ניווט ראשי, אזור ההאדר, לוגו, חיפוש, פוסטר וכן הלאה.

# תגית main

```
<body>
  <main>
    <h1>China</h1>
    <article>
      <h2>Beijing</h2>
      <p>The high air pollution caused by fossil</p>
    </article>
    <article>
      <h2>Shanghai</h2>
      <p>The AQI is an index for reporting</p>
    </article>
  </main>
</body>
```

# תגית aside

- אזור צדדי מבחינה לוגית
- תוכן משני או משלים לתוכן מרכזי אחר.
- מקובל להגדרת אזור login צדדי או אלמנט חדשות.
- כשהוא בתוך article, לדוגמה, הוא מהווה תוכן משני או משלים לתוכן שב-article.
- כשהוא ב-body, הוא מהווה תוכן משני או משלים לכל העמוד.
- לא לכלול בו מידע ראשי!

```
<aside>news</aside>
```

## 2 &lt;aside&gt;

## news &amp; updates

09/23/2011

**G By Equinox** Women up in NYW - G By Equinox is the newly launched, digital home for everything fitness, life and style.

02/01/2011

**I Want to Keep the Site Current.** Honestly - I've had so many different websites over the years from portfolios, blogs, band websites, etc. All of them start out with a bang, jam-packed with fresh content, but quickly become stale before I totally abandon it for real life.

01/01/2011

**1.1.11 - For our first New Year's in NYC** we decided to catch the "Drunkest Guns n' Roses Tribute band ever".

01/16/2010

**Get Ready NYC...** - Well, it finally happened. My wife and I finally got an apartment in New York. We will be residents of Brooklyn (Park Slope) next week. We are so excited!

06/02/2010

**New Work Added!** - Updates galore! Added some HTML, Email projects, as well as Flash repositories and outdoor banners to the portfolio.

05/07/2010

**Cheerio - You Rock!** - Thanks for adding my new site to your list!

[SEE ALL NEWS & UPDATES >](#)

&lt;section&gt;

&lt;section&gt; 4

## featured projects

[SEE ALL PROJECTS >](#)

## blog articles

&lt;section&gt;



## Things I Learn on my Morning Commute

TUESDAY, FEBRUARY 1ST, 2011 / 1 response, 740

I've learned so much from my morning commutes into Manhattan. For instance, I know which doctor has a plastic surgery plan for any budget, how many drinks ago I would have walked away from a fight, and something about the Knicks through the years...

&lt;article&gt;



## Developing a Sensible Folder Structure

MONDAY, APRIL 27TH, 2010 / 1 response

The folder structure of a designer or developer can quickly become unwieldy. Past concepts, invoicing and client communications, research and project updates really increase your file count rapidly. This article outlines my newest plan in the battle to end file-searching frustration, and maintain some type of convention for

&lt;article&gt;

## &lt;footer&gt;

## contact me

e: [team@twhite.com](mailto:team@twhite.com)  
p: 900.441.3029

## let's get social

follow me on Twitter  
friend me on Facebook  
connect with me on LinkedIn  
find me on Flickr

## subscriptions

subscribe to the portfolio feed  
subscribe to the blog  
subscribe to the news

## friends &amp; inspiration

Countrymore  
Ceklad  
Mister Wilper  
GIRL  
Sapient Creative  
Scott Hansen  
Scott Kaufman  
Shawn Inman  
Spring Holidid

# מודל התוכן של HTML5



# היסטוריה

- אלמנטים מסוג **Block-level** ומסוג **Inline**
- רוב האלמנטים ב-HTML נחלקו בעבר לאחת משתי הקבוצות: מבנה **Block** (בלוק) ומבנה **Inline** (כלול).
- תכונות של **block**:
  - מורידים שורה
  - רוחבם 100% מהרוחב הזמין להם, בהתאם לתגית שבתוכה הם מוכלים
- תכונות של **inline**:
  - לא מורידים שורה
  - לא ניתן להגדיר להם רוחב. הרוחב נקבע לפי התוכן

# div

---

- הייתה מקובלת בגרסאות קודמות של HTML.
- חסרת כל משמעות!
- מטרתה לתחום אזור מוגדר בעמוד.
- מקבילה לתגית section החדשה.
- הגדרות ברירת המחדל של <div> קובעות שהיא מסוג block, ועל כן ניעזר בה להמחשת הדוגמאות

# איך תראה התוצאה?

```
<div style="background-color:red;">I am a div tag. What is my default width? </div>  
<span style="background-color:green;">I am a span tag. What is my default width?</span>  
This text is outside the span tag.
```

# בלוק ואינליין

```
<div style="background-color:red;">I am a div tag. What is my default width? </div>  
<span style="background-color:green;">I am a span tag. What is my default width?</span>  
This text is outside the span tag.
```



התגית `<div>` מתפרשת על פני כל רוחב חלון הדפדפן.  
בין אם נפתח את הדפדפן במובייל או על כל רוחב מסך המחשב- בשני המקרים יהיה פס אדום אופקי לכל הרוחב.

# בלוק ואינליין

- להבדיל מאלמנטים מסוג block, האלמנטים מסוג inline אינם משפיעים על זרימת התוכן.
- הם יכולים להשפיע על עיצוב פרטני, אפילו של אות אחת.
- התגית span הייתה נהוגה כבר בגרסאות קודמות ומטרתה לתחום אזור תוכן.
- להבדיל מתגית div, התגית span הינה מסוג inline, ואינה גורמת למעבר לשורה חדשה לפניה ואחריה, ורוחבה נקבע בדיוק לפי רוחב הטקסט שכלול בה.

# תגיות בלוק מרכזיות

---

- `<div>`
- `<h1>-<h6>`
- `<hr>`
- `<p>`
- `<ul>,<ol>,<li>`

# חוקים

---

- רוב התגיות האחרות מוגדרות כאלמנטים מסוג inline, למעט יוצאי דופן, כמו `<br>`.
- הכלל היה קל ופשוט: אלמנטים מסוג blocked יכולים להכיל בתוכם אלמנטים inline ואלמנטים blocked.
- אלמנטים מסוג inline יכולים להכיל אלמנטים מסוג inline בלבד

# ה-DOM ומודל התוכן החדש



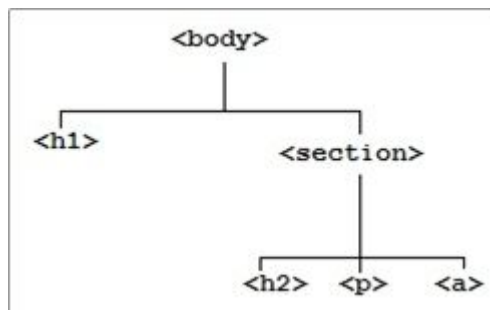
# DOM

- הדפדפן סורק את התגיות שבמסמך HTML5 והופך כל תגית לצומת בתוך מבנה המכונה עץ DOM=Document Object Model.
- מבנה זה מאפשר להגדיר את הקשר ההיררכי והתלותי בין תגית אחת לתגית אחרת.
- כמו בעץ משפחתי, כך גם בעץ DOM ניתן להגדיר תגית או צומת ראשי שתחתיו ניתן להגדיר צמתים-בנים.
- בעגה המקצועית הצומת נקרא node.
- הצומת, או ה-node, הראשי הוא השורש, או root.
- מצומת השורש מסתעפים שאר הצמתים.

# DOM - יחסים בין צמתים

## ● יחסים בין הצמתים:

- **אב ובן:** אם תגית מסוימת נמצאת בתוך תגית אחרת, הרי התגית הראשונה היא תגית "בן" והתגית השנייה היא תגית "אב".
- **אחים:** אם שתי תגיות נמצאות בתוך תגית שלישית, הרי הם "אחים" זה לזה ונמצאים באותה ההיררכיה.
- **אב קדמון:** כמו סבא מדור ראשון ומדור שני, כך "אב קדמון" הוא "אב של אב" בכל רמה שהיא. ראה למטה דוגמא מלאה, שתמחיש לך את המונחים.



# לקרוא סיפור

- ה- <section> מהווה את ה- root.
  - בתוכה תגית <ul> ותגית <h3>. בתוך התגית <ul> יש תגית פנימית <li>.
  - בדוגמה זו, <li> הוא node בן-ישיר של <ul> ו- node בן של <section>.
  - בהסתכלות הפוכה מבפנים כלפי חוץ, <section> הוא אב ישיר של <ul>
  - <section> הוא אב קדמון של <li>.
  - h3 node הינו אח sibling של ul node, מכיוון שלשניהם יש אותו אב: <section>.
- 
- כך סורק הדפדפן את כל התגיות במסמך, יוצר עץ היררכי ומשייך את התגיות לסוגי מידע.

```
<section>
  <ul>
    <li>Some Text</li>
  </ul>
  <h3>Some Title</h3>
</section>
```

# העשרה

החומר הבא אינו למבחן

# הקטגוריות הראשיות ב-HTML5

---

## Metadata

- סיווג זה כולל תגיות שתורמות להגדרות המסמך ומופיעות לרוב בתוך התגית `<head>`.
- `<base>`, `<title>`, `<command>`, `<link>`, `<meta>`, `<script>`, `<style>` לדוג'

# הקטגוריות הראשיות

---

## Sectioning

- תגיות שמציגות את המבנה של המסמך, בתוך תגיות מבנה אחרות.
- התגיות בסיווג זה: `<section>`, `<article>`, `<nav>`, `<aside>`.
- נושא זה רלוונטי לדיון הבא בנושא `outline algorithm`.
- התגית `<main>` אינה כלולה פה.

# הקטגוריות הראשיות

## Flow

- תגיות הזרימה של הדף
- כלומר התגיות שיוצרות את המבנה ואת התצוגה שלו.
- בסיווג זה נכללות רוב התגיות שניתן לשבץ בתוך התגית `<body>`, וביניהן:
- `<a>`, `<article>`, `<section>`, `<p>`, `<aside>`, `<canvas>`, `<div>`, `<cite>`, `<command>`, `<main>`, `<ul>`, `<li>`, `<ol>`, `<h1-6>`, `<figure>`, `<footer>`, `<header>` ותגיות נוספות.
- יש חפיפה בין תגיות מסוימות. התגית `<command>` שייכת הן לסיווג Metadata והן לסיווג Flow
- בהגדרות של HTML5 לא מוגדר במפורש שאלו הן תגיות `!block`

# הקטגוריות הראשיות

---

## Heading

- תגיות הכותרות
- `<h1-6>`
- התגית `<header>` אינה נכללת פה!



# הקטגוריות הראשיות

## Phrasing

- תגיות הקשורות לטקסט או מדגישות טקסט שבדף.
- `abbr`, `b`, `i`, `a`, `mark`, `span`, `iframe`, `select`, `button`, `input`, `>`, `<label` ועוד.
- רוב התגיות בסיווג זה יכולות להכיל בתוכן תגיות אחרות שגם נכללות בתוך סיווג זה בלבד.
- כך ניתן להכיל את `mark` בתוך `i` אך לא ניתן להכיל תגית `head` בתוך תגית `i`, וזה כמובן נשמע מאוד הגיוני.
- מאוד מזכיר את הסיווג Inline שהוצג בסעיף הקודם.
- אלמנטים אלה אינם גורמים למעבר לשורה חדשה, אלא ממשיכים את התוכן בשורה הנוכחית. ניתן לכתוב את התגית `mark` באותה שורה ליד `mark`.

# הקטגוריות הראשיות

## Embedded

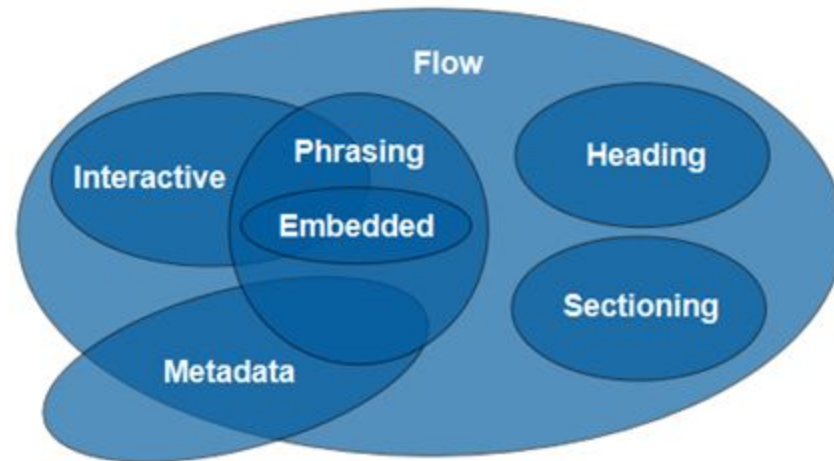
- תגיות מוטמעות עם מדיה או תמונות חיצוניות שמוצגות בדף, כגון `<audio>`, `<video>`, `<img>`, `<object>`, `<canvas>` ועוד.
- ברוב התגיות שבסיווג זה ניתן להגדיר מה יוצג אם הדפדפן אינו תומך בהצגת התמונה או המדיה המוגדרת בקוד.

# הקטגוריות הראשיות

## Interactive

- תגיות שמשמשות לאינטראקציה עם הגולש, כגון בחירת אלמנט מתוך רשימה נגללת או לחיצה על קישור.
- `<a>`, `<input>`, `<video>`, `<select>` ועוד.
- סיווג זה עדיין בפיתוח.
- התגית `<video>` הינה אינטראקטיבית רק אם לחצני הפעלת הסרטון גלויים לעיני הגולש, אחרת הוא לא יוכל להפעיל או להפסיק את הסרטון ולכן הפעולה לא תהיה אינטראקטיבית.

# יחסי הגומלין בין הקטגוריות הראשיות



# OUTLINE ALGORITHM

- אלגוריתם קווי-המתאר של התוכן
- מחולק בדרך כלל לפרקים, לנושאים ולתת-נושאים.
- כמו בתוכן העניינים בספר.
- מאפשר להבין את החשיבות של כל פריט בתוכן על פי המיקום של הנושא או של תת-הנושא בהיררכיה הכללית.
- בדרך דומה צריך גם הדפדפן להבין מהי ההיררכיה של המסמך שעליו להציג. איזו תגית חשובה יותר מאחרת, איזו תגית קשורה לאחרת, וכן הלאה.
- לדוגמה, בדף עם כותרת ראשית וכותרות משנה שתחת כל אחת מהן יש פסקת טקסט, הדפדפן צריך להבין את מה שאנחנו רואים בעיניים: איזה תוכן רלוונטי לאיזו תת-כותרת.
- כדי שהדפדפן יהיה מסוגל לדרג את הרכיבים השונים וליצור חלוקה היררכית של התוכן, גם מבלי להבין מה כתוב בו, הוגדרה עקביות מסוימת באמצעות אוסף של חוקים שהדפדפן יכול לפענח.

# החוקים

1. התגית `<body>` מוגדרת בתור sectioning root, כלומר זו תגית השורש. עבור הדפדפן זו תגית התוכן הראשית, המכילה את כל שאר תגיות התוכן. הדפדפן קורא אותה תחילה ומדרג אותה בתור הרמה הגבוהה ביותר.
  2. הדפדפן ממשיך וסוקר את התגיות שכלולות בתגית `<body>`. הוא מוצא עוד ועוד תגיות בסיווג sectioning. משמע, תגיות נוספות שיוצרות את המקטעים והאזורים הראשיים בדף האינטרנט. הרמה השנייה מוגדרת על ידי תגיות האזורים היעודיים מסוג `<section>`, `<main>`, `<aside>`, `<nav>`, `<article>`, שכלולים בתוך התגית `body`.
  3. התגיות שמוגדרות בתוך התגיות שברמה השנייה הינן בהיררכיה נמוכה יותר, בהתאם לסוג תגית הכותרת. תגית הכותרת הראשונה, נניח `h2`, תוגדר בתור ראש המקטע. תגיות `h2` נוספות במקטעים אחרים, תהינה בהיררכיה זהה.
- יש ברשת כלי תכנות שמסייעים לבנות ולהגדיר נכונה ובתבונה את אלגוריתם קווי המתאר.