

CSS

נכתב ונערך על ידי יונית רוש

מבוא

1

CSS

- גיליונות סגנון מדורגים
- סגנונות הקובעים כיצד יוצגו האלמנטים של מסמך ה-HTML
- מאפשרים למפתח לעצב דפים בנוחות וביעילות
- ב-HTML בלבד אנו מוגבלים ברמת העיצוב
- CSS מאפשרת הפרדה מוחלטת בין העיצוב לבין התוכן

דוגמה

- http://w3schools.com/css/demo_default.htm
- במסמך HTML מגדירים את תבנית הדף עם משמעות ואת התוכן **content** (הטקסטים, התמונות, הקישורים) שמגיעים מה- DB
- ב-CSS מגדירים את העיצוב שלו- את אופן התצוגה **presentation**
- ניתן להחליף עיצוב על ידי שינוי הגדרות ב- CSS מבלי לשנות דבר בקוד HTML עצמו
- HTML היא שפה מבוססת תגיות
- CSS היא שפה שמוכוונת על ידי רשימה של הגדרות עיצוב.
- בתגיות HTML לכל תגית יש תמיד סימני <...> מסביב לשם התגית
- ב-CSS אין את אלו.

ההגדרה והערך

- כל הגדרת עיצוב מורכבת משני חלקים: ההגדרה והערך שלה.
- צבע רקע: כחול, היישום: `background-color:blue`
- צבע גופן: אדום, היישום: `color:red`
- גודל אות: 12 פיקסלים, היישום: `font-size:12px`;
- רווח מלמעלה: 56 פיקסלים היישום: `padding-top:56px`;
- סוג גופן: אריאל (Arial), היישום: `font-family:Arial`

הדרכים להגדרת העיצוב

2

INLINE

- Inline - הגדרת העיצוב כתכונה נוספת עבור התגית. לכל תגית יש מאפיין בשם style האחראי על העיצוב של תוכן התגית.
- `<b style="background-color: red;">Text in b`
- נפריד בין תת מאפיין לתת מאפיין באמצעות ; (נקודה פסיק) :
- `<b style="background-color:red;color:blue;">Text in b`

INLINE הבעיה

- `<h1 style="color: green;">My Header</h1>`
- אם נעתיק את העיצוב עבור כל האלמנטים מסוג `h1` הקיימים בדף ה-HTML, ובעתיד נרצה לשנות את הצבע נצטרך לעבור שוב על כל האלמנטים ולשנות את עיצובם.

INTERNAL

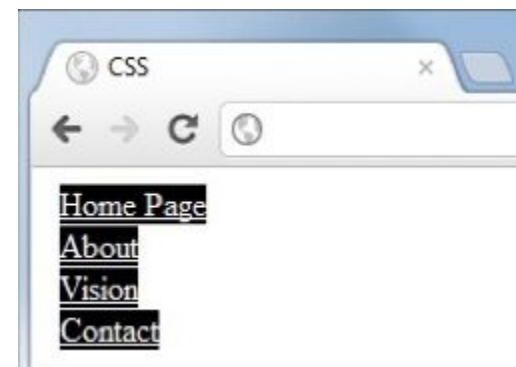
- תגית style (Internal) – לשמירת אחידות עיצוב האלמנטים.
- ניתן לפתור בעיה זו באמצעות תגית style.
- את התגית נגדיר בתוך תגית ה-head הנמצאת בראש הדף.
- בתגית זו נכתוב את העיצוב בצורה הבאה:

```
<style>
  elementName {
    color: blue;
    background-color: red;
    margin-left: 5px;
  }
</style>
```

```

<!DOCTYPE html>
<html>
  <head>
    <title>CSS</title>
    <style>
      a {
        color: white;
        background-color: black;
        margin-left: 5px;
      }
    </style>
  </head>
  <body>
    <nav>
      <a href="index.html">Home Page</a><br>
      <a href="about.html">About</a><br>
      <a href="vision.html">Vision</a><br>
      <a href="contact.html">Contact</a>
    </nav>
  </body>
</html>

```



בעיה 1: מה נעשה עם הקישורים שבפוסטר? הרי יש להם עיצוב שונה.

```
<style>
  b {
    color: green;
    background-color: #4F4F4F;
  }
  h1 {
    color: #0000FF;
    background-color: blue;
  }
</style>
```

בעיה 2: מערכת עם 20 סוגי עמודים ורוצים לשנות בכל אחד מהם את תמונת הרקע

EXTERNAL

- קובץ חיצוני (External)
- היתרון בעיצוב באמצעות קובץ חיצוני הוא שניתן להשפיע על עיצוב המערכת כולה במסמך אחד.
- ניתן לשייך קובץ CSS אחד למספר רב של דפי HTML.
- סיומת קובץ העיצוב היא CSS.
- בתוך קובץ זה נכתוב את הגדרות העיצוב בדומה מאוד להגדרה בתגית `.style`.
- על מנת לקשר בין קובץ ה-CSS לדף HTML יש להוסיף את התגית `link` בראש המסמך בצורה הבאה:

```
<head>  
  <link rel="stylesheet" href="includes/style.css">  
  <title></title>  
</head>
```

- בתכונה `href` יש לרשום את מיקום קובץ ה-CSS
- אנו נשאף להשתמש באופציה זו – קישור לקבצי CSS חיצוניים
כל הזכויות שמורות ליונית רוש

דוגמאות

3

-Inline

```
<a href="lastpage.html" style="text-decoration:none;">  
THIS IS A LINK!</a>
```

-Internal

```
<head>
  <style>
    h1 {color: red;}
    h3 {color: blue;}
  </style>
  <title>Daily News</title>
</head>

<body>
  <h1>This is header 1</h1>
  <h3>This is header 3</h3>
</body>

<p style="color:red;margin-left:20px;">
This is a paragraph
</p>
```

דוגמאות External

אם יש משולשים מסביב אלו תגיות ב-html, ואם אין- אלו הגדרות עיצוב בקובץ CSS

בתוך קובץ ה- CSS החיצוני:

```
a {  
    text-decoration: none;  
}
```

בתוך קובץ ה-html:

```
<a href="lastpage.html"> THIS IS A LINK!</a>
```


בתוך קובץ ה- CSS החיצוני:

```
h1 {  
    color: #666666;  
}  
h3 {  
    color: #909045;  
}
```

בתוך קובץ ה- :html

```
<body>  
    <h1>This is header 1</h1>  
    <h3>This is header 3</h3>  
</body>
```

בתוך קובץ ה- CSS החיצוני:

```
section {  
    padding-left: 15px;  
    margin-top: 10px;  
}  
nav {  
    padding-top: 8px;  
    margin-top: 10px;  
}
```

בתוך קובץ ה- CSS החיצוני:

```
body {  
    background-image: url(ptrn.png);  
}
```

```
p {font-family: Arial;}  
p {text-align:center; color:red;}
```

OR-

```
p {  
    text-align: center;  
    color: red;  
    font-family: Arial;  
}
```

- **color** תמיד מתייחס לצבע הפונט
- צבע ניתן להגדרה גם בצורת **RGB** וגם בתוספת הגדרת שקיפות:

```
background-color:red;
```

```
background-color:#000000;
```

```
background-color:rgb(0,0,255); //blue
```

```
background-color:rgba(255,0,0,0.8); //80% opacity red
```

סינטקס והיררכיה

4

ריבוי הגדרות עיצוב

- יש לשים נקודה פסיק בכל סוף שורה / הגדרת מאפיין-ערך

בתוך קובץ ה- CSS החיצוני:

```
h4 {  
  color: #ffffff;  
  background-color: #663366;  
  margin-left: 5px;  
}
```

עדיפויות

- סדרי עדיפויות נקבע מבפנים החוצה (מה שיותר קרוב יותר קובע)
- ניתן להגדיר עבור אותו אלמנט הגדרות עיצוב במספר מקומות.
- כאשר אין סתירה בין העיצובים, יחולו כל הגדרות העיצוב על האלמנט.
- כאשר ישנה סתירה, העדיפות נקבעת לפי הסדר הבא:

Inline style	1
Internal style	2
External style sheet	3
Browser default	4



עדיפויות

- אבל מה קורה כשבשני מקומות מוגדר, נניח, צבע פונט?
- זהו נושא מורכב: מה החוקים שקובעים אילו הגדרות עיצוב יתפסו?
- בעיקרון, id תמיד יתפוס לפני שם התג:

id	1
class	2
tag name	3



- מי יתפוס? האדום או הכחול?

```
#news {color:red;}  
.my_news {color:blue;}  
  
<aside id="news" class="my_news">
```


עדיפויות

- **תשובה:** האדום, מאחר והוא מוגדר ב- id

- אבל בשימוש בכמה סוגי הגדרות בהיררכיה זה מתחיל להסתבך, כמו בדוגמה הזו:

- `#news h1.ttle {color:green};`

- ולכן נקפיד שלא ליצור בלאגן ב- CSS.

- ניתן להוסיף את המילה `!important` להגדרת עיצוב כדי לסמן לדפדפן את העדיפות.

משמע, אם יש התנגשות עם מאפיין עיצוב אחר – המאפיין שלידו יש "חשוב" ישפיע.

`a {color:yellow !important}`

- זה מקרה קיצון ועדיף שלא להגיע לזה

Grouping

- ניתן להגדיר מספר תגיות ולהפרידם באמצעות פסיק, לדוגמא:

```
h1,h2,h3,b {  
  color:#ffffff;  
  background-color:#000000;  
}
```

CSS comments

```
p {  
    text-align: center;  
    /* This is a comment */  
    color: black;  
}  
  
/*-----  
    Purpose: Main CSS File- DesignersAreUs.com  
    Version: 0.2  
    Author: Ofek G  
    Date: 18-1-13  
    -----  
*/
```

Classes

5

שימוש ב- classes

- כאשר מגדירים עיצוב עבור אלמנט מסוים - נקבע העיצוב של כל האלמנטים מאותו סוג באופן גורף.
 - ישנם מקרים בהם אנו רוצים לנצל את יכולות ה- CSS אך איננו רוצים שהעיצוב שהגדרנו יחול באופן גורף.
- ```
.mainTitle {
 background-color: red;
 margin-top: 40px;
}
```
- שיוך האלמנט ל- class:
- ```
<h1 class="mainTitle">My Header</h1>
```
- הגדרת העיצוב לא משפיעה באופן גורף על כל האלמנטים מסוג h1
 - הנקודה היא כל ההבדל

מי נדבק למי?

Inside the CSS file:

```
.titles {  
    text-align: center;  
    font-size: 18px;  
    color: black;  
    font-family: arial;  
    font-weight: bold;  
}
```

Inside the HTML file:

```
<body>  
    <h1 class="titles">First Lesson</h1>  
    <h1>November 1999</h1>  
    <p>What is the...</p>  
</body>
```

- מה יקרה אם נחליף את תג ה- h1 בתג p?
- מה יקרה אם נשאיר את תג ה- h1, אך נוריד את הגדרת הבולד?

הגדרת שני classes



```
<p class="centerIt boldIt">  
    This is a paragraph.  
</p>
```

הגדרת שני תנאים

- ניתן לחייב ששני תנאים יתקיימו, לדוגמה :

```
h1.childish {  
  border-bottom-style:groove;  
}
```

- הגדרות עיצוב אלו יעבדו רק לאלמנט ששני התנאים מתקיימים:

(1) תגית h1, ו- (2) מוגדר לה class של childish

- לדוגמה-

```
<h1 class="childish">First Lesson</h1>
```

- והגדרות העיצוב לא יחולו על-

```
<h2 class="childish">First Lesson</h2>
```

- שימו לב לכך, שאין רווח בין שם התג לבין שם ה- class

- אם יש רווח (נלמד בהמשך) - המבנה המתבקש יהיה שונה: אלמנט כלשהו עם class childish בתוך

אלמנט h1

דוגמאות

דוגמאות נוספת, בקובץ החיצוני-

```
p.right {text-align: right;}  
  
.nice.center {text-align: center;}  
  
- - - - -
```

וב-HTML:

```
<p class="right">  
    This paragraph will be right-aligned.  
</p>  
<aside class="nice center">  
    This paragraph will be nicely center-aligned.  
</aside>
```

לסיכום

לסיכום- class שמוגדר כך:

```
.centerIt {text-align: center;}
```

ישפיע על כל התגיות בדף, שבתגית הפתיחה שלהם יהיה כתוב:

```
<?? class="centerIt">
```

ו- class שמוגדר כך:

```
p.centerIt {text-align: center;}
```

ישפיע על רק על תגיות ה- p שבדף, שבתגית הפתיחה שלהם כתוב:

```
<p class="centerIt">
```

הגדרות עיצוב לפי id

6

ה-id

- שימוש ב-id - שם המאפיין, ה-id:

```
#leftMenu {color: green;}
```

- וב-body:

```
<nav id="mainMenu">
```

- id לא יכול רווחים ולא יחזור על עצמו יותר מפעם אחת בדף
- שימושי עבור האזורים היעודיים הראשיים, שבוודאות אינם חוזרים על עצמם, כגון:
logo, mainMenu, wrapper

- לא לתת id זהה לשם של התגית

```
<header id="header">
```

הגדרות עיצוב לקישורים

7

הגדרות סטנדרטיות לקישור

- סט של הגדרות CSS עבור קישורים עם מעבר עכבר:
- ללא מעבר עכבר + הגדרת צבע פונט

```
a {  
  text-decoration:none;  
  color: #659965;  
}
```

מעבר עכבר מעל קישור

```
a {  
  font-size:14px; /*stays the same*/  
  text-decoration:none;  
  background-color: #000000;  
  color: #659965;  
  display: block;  
}  
  
a:hover {  
  color: #00FF00;  
  text-decoration:underline;  
  background-image: url(../images/bgr.jpg);  
}
```

- איך יראה הקישור כשהעכבר עליו?
- מה יקרה על מובייל?

visited

- הפרדת העיצוב של קישור שביקרו בו

```
a:link {  
    color: #659965;  
}  
  
a:visited {  
    color: #666666;  
}
```


הגדרת תמונת רקע לקישור

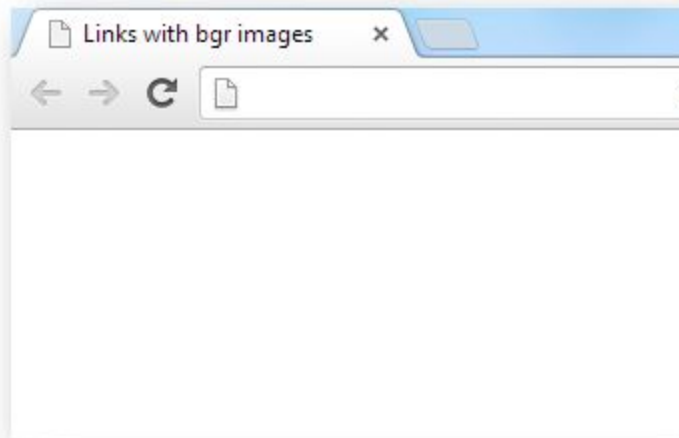
- בעיה: לתגיות שאינן מורידות שורה (inline) – לא ניתן להגדיר גודל!
- הפתרון: להפוך אותן לתגיות שכן מורידות שורה (**block**).
- להלן דוגמה של קוד (שימו לב, תגית הקישור ריקה):

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Links with bgr images</title>
5     <link rel="stylesheet" href="includes/style.css">
6   </head>
7   <body>
8     <section>
9       <a href="contact.html"></a>
10    </section>
11  </body>
12 </html>
```

```
1 a {
2   background-image:url(../images/logo.jpg);
3   width:100px;
4   height: 200px;
5 }
6
```

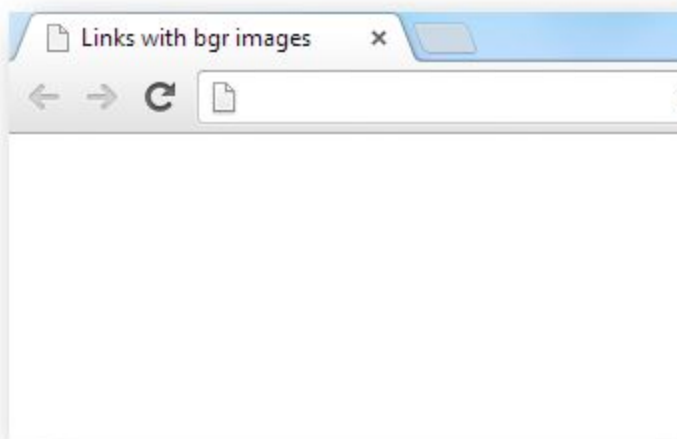
התוצאה בדפדפן

- איפה תמונת הרקע?????



התוצאה בדפדפן

- איפה תמונת הרקע?????

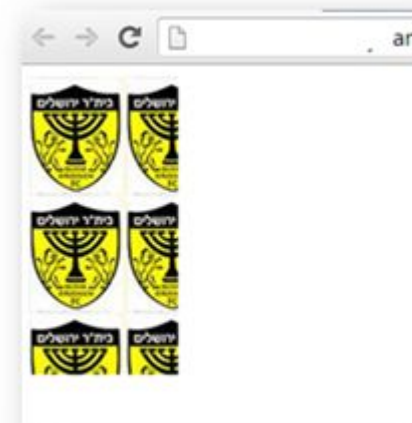


- מאחר והתגית ריקה מתוכן, ומאחר ולא ניתן להגדיר לה גודל – לא רואים את תמונת הרקע!!!

הפתרון

- השינוי הדרוש עבור הפתרון (שורה 5):

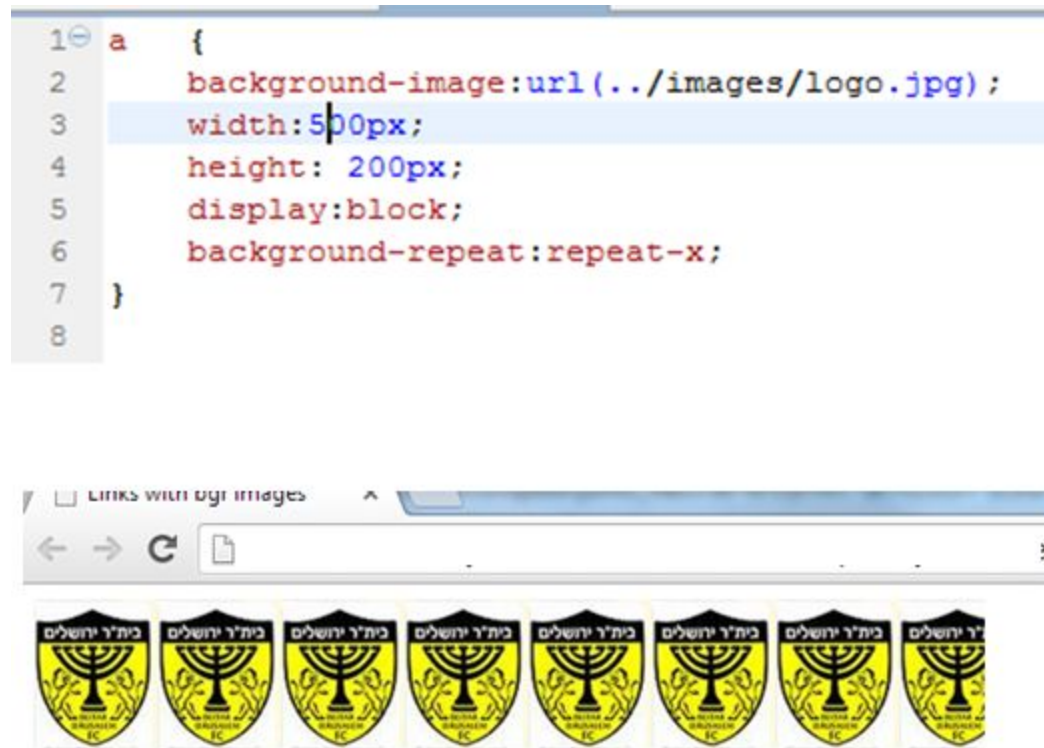
```
1 a {  
2   background-image:url(../images/logo.jpg);  
3   width:100px;  
4   height: 200px;  
5   display:block;  
6 }
```



- תמונת הרקע שכפלה את עצמה, כי לא הוגדר אחרת, על פני כל הגובה המוגדר.

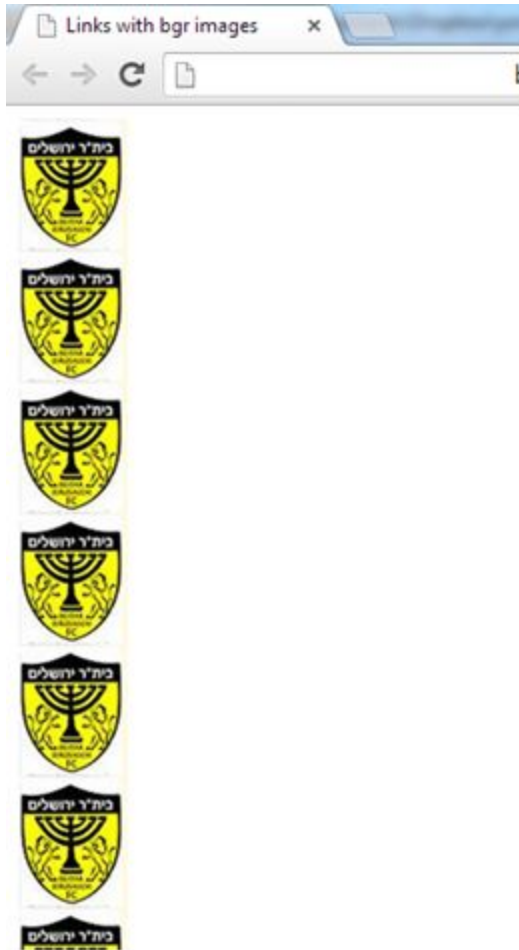
שכפול תמונות רקע

- שכפול לרוחב:



שכפול תמונות רקע

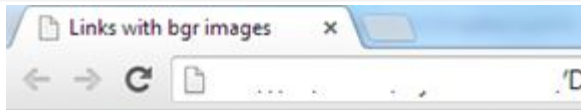
- שכפול לגובה:



```
1 a {  
2   background-image:url(../images/logo.jpg);  
3   width:100px;  
4   height: 500px;  
5   display:block;  
6   background-repeat:repeat-y;  
7 }  
8
```

ללא שכפול

- למה נוצר כל הרקע האפור?



```
1 a {  
2   background-image:url(../images/logo.jpg);  
3   width:100px;  
4   height: 500px;  
5   display:block;  
6   background-repeat:no-repeat;  
7   background-color:#666666;  
8 }  
9
```

לוגו לעולם אינו תגית תמונה

- תזכורת: מימוש שונה לתמונות שהן תוכן לעומת תמונות שהן גרפיקה



תמונות- איפה להגדיר?

- תמונה שהיא חלק מהגרפיקה: נגדיר בתמונת רקע ב- CSS:

```
background-image: url(../images/logo.jpg);
```

- תמונה שהיא חלק מהתוכן: נגדיר בתגית `img` כדי שקוד דינאמי יוכל לשנות אותה:

```

```

מסקנות

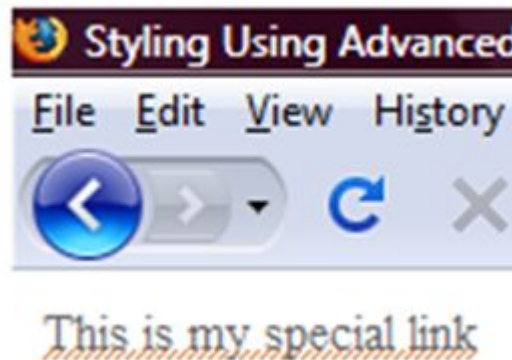
- **קישור** שאין בו תוכן והוא אך ורק **תמונת רקע** – חייב להיות **block** כדי שנראה את תמונת הרקע.
- ההתנהגות זהה לכל תגיות ה- `inline`

- נפריד בין תמונה שהיא חלק **מהתוכן** ותוגדר על ידי התגית ``
- לבין תמונה שהיא חלק **מהעיצוב הגראפי** ותוגדר על ידי תמונת רקע ב- `CSS` כך:
`background-image:url();`

- קישור יכול לשנות צבע במעבר עכבר. יופי.
- אבל **קישור שבמעבר עכבר משתנה התמונה** שלו – התמונה חייבת להיות מוגדרת **כתמונת רקע** ולא בתגית ``.

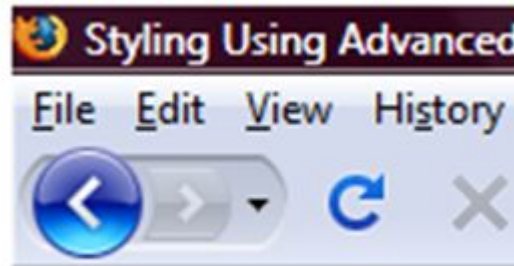
עיצוב קישורים מתקדם

- הדרך הפשוטה ביותר לעצב קישור היא: `a {color: blue}`



- קו תחתון בעיצוב אישי

עיצוב קישורים מתקדם



קו תחתון בעיצוב אישי



```
a:link, a:visited {  
    color: #666666;  
    text-decoration: none;  
    background: url(images/underline.jpg) repeat-x left  
bottom;  
}
```

תמונות רקע- חזרה ודגשים

8

הבסיס

- הוספת רקע לאתר פשוטה וידועה לנו

```
body {  
    background:url(some_pattern.jpg);  
}
```

- איך נחתוך?



כל הזכויות שמורות ליונית רוש

לפני CSS3

- ליצירת גרדיאנט gradients אופקי:



- נחתוך תמונה ברוחב 1 פיקסל ונכתוב:

```
background: #999999 url(some_pattern.jpg) repeat-x;
```

בעמוד ארוך מרובה תוכן

- יש להניח שהעמוד דינאמי ושהטקסט נכתב בממשק הניהול CMS.
- אנחנו לא יודעים מראש מה יהיה גובה העמוד
- אם יש בעמוד הרבה טקסט – תמונת הרקע לא תכסה את כל העמוד
- ומתחת לאפור הכהה יופיע פתאום לבן



long text
long text
long text
long text

long text
long text

long text
long text

long text
long text

long text
long text

long text
long text

long text
long text

long text
long text

long text
long text

long text
long text

long text
long text

בעמוד ארוך מרובה תוכן

- פתרון: נגדיר צבע רקע שיתמזג עם תחתית תמונת הרקע



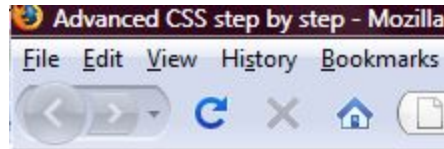
גרדיאנט על ציר Y

- כיצד ניצור את הגרדיאנט הבא?



מיקום תמונת רקע

- לדוגמא, נניח ורוצים להוסיף icon מעוצב משמאל לכותרת:



■ My Head Title

```
h1 {  
  padding-left: 20px;  
  background: url(../images/bullet_bgr.jpg) no-repeat left center;  
}
```

```
<h1>My Head Title</h1>
```

- שתי המילים השמורות בסוף מציינות מיקום לתמונת הרקע.
- Left - המילה הראשונה מציינת מיקום אופקי.
- Center - המילה השנייה מציינת מיקום אנכי.

משמעות

9

class names ו-ID

- אזורים ייעודיים שיש להם תגית ייחודית שמופיעה פעם אחת בעמוד: header, footer, main לא צריכים id
- אזורים חשובים שאין להם תגית כמו לוגו, צריכים id
- תפריטי ניווט צריכים id כי יכולים להיות כמה כאלו בעמוד (mainNav, secondNav)
- יש להשתמש במעט id בעמוד, מפני שמבחינה **היררכית** האלמנטים האלו מקבלים **עדיפות** בהגדרות העיצוב שלהם ב-CSS וזה יכול לגרום לקונפליקטים אם יש הרבה.

רשימה פשוטה של קישורים

- ניתן לתגית האב nav שם mainNav ובכך נהפכה לאלמנט ייעודי שמטרתו ברורה לכל:

```
<nav id="mainNav">
  <ul>
    <li><a href="#">Home</a></li>
    <li><a href="#">Company Profile</a></li>
    <li><a href="#">Contact Us</a></li>
  </ul>
</nav>
```


Good and bad names

- לא להכניס עיצוב לשמות.
- מיקום של אובייקט וצבע הם חלק מהעיצוב

- Bad names:
 - red
 - topNav
 - leftColumn
- Good names:
 - errorMsg
 - mainNav
 - secondaryContent
 - intro

מינימליזם =
צמצום כמות ה-classים

10

היררכיה

- הגדרת מאפיינים לכל התגיות מסוג מסוים, שבתוך תגית עם class מסוים (שימו לב לרווח שמציין "בתוך")

```
.mainMenu a {  
    text-align: left;  
    font-size: 12px;  
    color: #0000FF;  
    text-decoration: none;  
    background-image: url(../images/phone.jpg);  
}
```

And in the body-

```
<nav class="mainMenu">  
    <a href="index.html">Home</a>  
    <a href="contact.html">Contact Us</a>  
</nav>
```

הכוכבית

- שימושית מאוד.
- מציינת התייחסות לכל הצאצאים/ילדים של אלמנט מסוים.
- קיימות אפשרויות נוספות לכוון הגדרות עיצוב למטרה.

* `{margin:0; padding:0;}`

צמצום class'ים

- יש להשתמש ב-class במידה. לא בכמות מופרזת:

```
<article>
  <h1 class="articleHead">Article 1</h1>
  <p class="articleText">One more milestone for the
company, as it turned 5 yesterday.</p>
  <p class="articleText">
    <a href="article1.html"
      class="articleLink">Read More</a>
  </p>
</article>
```

להלן אותו המבנה בפשטות

- * יחידות מידה em הן יחסיות. אם מוגדר גודל פונט דיפולטי לעמוד או לכל אלמנט אב (נניח ל- article), אזי יחידת 1.5em היא פי 1 וחצי מהגודל הדיפולטי.

```
<article class="article1">
  <h1>Article 1</h1>
  <p>One more milestone for the company, as it turns 5
  today</p>
  <a href="articles.html">Read More</a>
</article>
```

```
.article1 h1{font-size:1.8em;}
.article1 p{font-size:1.1em;}
.article1 a{font-size:1.2em;}
```

selectors

- http://www.w3schools.com/css3/css3_ref_selectors.asp
- http://www.w3schools.com/css/css_attribute_selectors.asp
- <http://www.w3.org/TR/CSS2/selector.html> (מסמך רשמי)

- אבא - התג המכיל את התג הנוכחי
- בן/ילד דור ראשון - כל התגים שמוכלים בתוך תג נוכחי
- אח - כל התגים שמוכלים באותו אבא שהתג הנוכחי מוכל בו
- נכד - תג בתוך בן של התג הנוכחי
- אב קדמון - כל התגים בהיררכיה למעלה
- בן/ילד לדורותיהם - כל הילדים של התג הנוכחי, בכל ההיררכיה הפנימית שיש

ייעול הקוד

● המטרה:

ייעול הקוד, מינימליזם, מינימום שימוש ב-
class, "קליעה למטרה"

descendants

```
li {color:green;}
```

```
.nav1 > li {  
  color:blue;  
}
```

```
<ul class="nav1">  
  <li>Home</li>  
  <li>Products  
    <ul>  
      <li>Candies</li>  
      <li>Sweets</li>  
      <li>Gumms</li>  
    </ul>  
  </li>  
  <li>Contact Us</li>  
</ul>
```

- נבדיל בין ילדים לדורותיהם (descendants) לבין ילדים מדור ראשון.
- אסטרטגיה – איך נטפל?
- בדוגמא הבאה כל הילדים מדור ראשון שנמצאים ברשימה החיצונית יהיו בכחול והילדים שברשימה הפנימית לא יושפעו ויישארו בירוק.

- Home
- Products
 - Candies
 - Sweets
 - Gumms
- Contact Us

immediate following sibling

- אח עוקב מידי
- בדוגמא הבאה נגדיר את האלמנט שעוקב אחרי אלמנט אחר (הפסקה הראשונה והמיידית שבאה אחרי הכותרת):

```
h1 + p {font-weight:bold;}
```

```
<h1>Main Title</h1>  
<p>first paragraph</p>  
<p>second paragraph</p>
```

Main Title

first paragraph

second paragraph

following sibling

- נבדיל בין הדוגמא הקודמת, לבין:

```
h1 ~ p {font-weight:bold;}
```

- במצב זה **כל הפסקאות** שמגיעות איפשהו **אחרי** **h** יהיו **בולד**:

- `<h1>Main Title</h1>`
- `<p>will be bold</p>`
- `<p>will be bold</p>`
- `<h2>Main Title</h2>`
- `<p>will be bold</p>`

Attribute

- רק לתגיות שיש להן attribute מסוים:

```
[title] {  
  color:blue;  
}
```

ערך ספציפי של מאפיין

```
input[type="text"] {  
    width:100px;  
    display:block;  
    margin-top:15px;  
    background-color:pink;  
}  
input[type="submit"] {  
    width:150px;  
    margin-right:35px;  
    display:block;  
}
```

תוכן בתוך מאפיין

- דוגמא לתפיסת אלמנט בעל מילה בתוך מאפיין, מופרדת על ידי רווחים

```
a[title~="friend"]{color:red;}
```

```
<a href="http://www.goforit.com" title="friend at  
work">Mike</a>
```

Mike

- כדי למצוא ביטוי שאינו מופרד על ידי רווחים משתמשים ב- `a[title*="X"]`

מתחיל ב..., נגמר ב....

- לדוגמה, לצורך הדגשת קישורים מסוגים שונים
- הצורך: סימון קישורים חיצוניים external שמקשרים לעמודים במערכת אחרת לעומת קישורים internal שמקשרים לעמודים במערכת הנוכחית

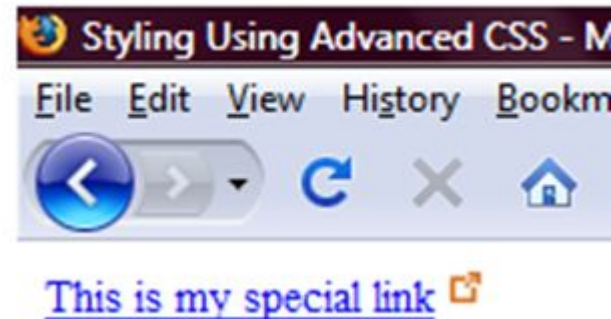
הדרך הפשוטה

- ניצור תמונה קטנה –



- הדרך הפשוטה היא ליצור class לכל קישור חיצוני באופן ידני:

```
.external {  
    background: url(images/external-website.jpg)  
no-repeat right top;  
    padding-right: 15px;  
}
```



הדרך הדינאמית

- ננסה ליצור הפרדה בין קישורים חיצוניים לקישורים פנימיים ללא התערבות ידנית, בעזרת ה- CSS בלבד-

- הערה: אנו נשתמש ב- selectors של CSS3 ולא כל הדפדנים תומכים בכך.

- להלן קוד שמתייחס לכל הקישורים שמתחילים ב- http, מתוך הנחה שהם חיצוניים:

```
a[href^="http://"] {  
    background: url(images/external-website.jpg) no-repeat  
    right top;  
    padding-right: 15px;  
}
```

הדרך הדינאמית- SEO

- אך כיום ברוב האתרים המקודמים בגוגל, משתמשים בקישורים אבסולוטיים ולא בקישורים רלטיביים, ולכן כל הקישורים מתחילים ב- http.
- לכן, נוסיף בהתחלה את שם הדומיין של האתר שלנו-
- ונגדיר שלכל הקישורים הפנימיים לא תהיה תמונת רקע:

```
a[href^="http://www.mydomain.co.il"], a[href^="http://mydomain.co.il"] {  
    background-image: none;  
    padding-right: 0;  
}
```

- התוצאה: לכל הקישורים שמתחילים ב- http יש את האייקון, אך רק לאלו שמקשרים לעמודים באתר עצמו לא יהיה את האייקון. כך האייקון יישאר רק לקישורים חיצוניים.
- באותו האופן ניתן לבדל קישורים למייל:

```
a[href^="mailto:"]
```

- ולשם בידול מסמכים להורדה, שניתן לזהותם על פי הסופית שלהם

נשתמש בהגדרת הדולר ב-CSS3 שמתייחסת לסוף הערך:

```
a[href$=".pdf"] {  
    background: url(images/pdf_icon_link.jpg) no-repeat right top;  
    padding-right: 15px;  
}  
a[href$=".doc"] {  
    background: url(images/doc_icon_link.jpg) no-repeat right top;  
    padding-right: 15px;  
}
```

pseudo classes

?11