

מה שראינו עד כה היו דפי אינטרנט שניתנים להצגה על ידי הדפדפן (עם סיומת של html או html)

כעת נלמד על דפים שבהם יש קוד שמיועד להרצה בשרת. אחד המאפיינים עבור דף כזה הוא:

```
runat="server"
```

זה פשוט מציין שהדף צריך להיות מורץ על ידי השרת, כי הדפדפן לא מכיר את כל מה שמופיע בדף.

הדף שמורץ על השרת, יראה דומה לדף html עם תוספות. הנה דוגמה: (מספרי השורות אינם חלק מהקוד)

```
1 <%@ Page Language="C#" %>
2 <script runat="server">
3   public string mytime;
4   protected void Page_Load()
5   {
6       mytime=(DateTime.Now.ToString());
7   }
8 </script>
9 <html xmlns="http://www.w3.org/1999/xhtml">
10 <head>
11 <title> My first page </title>
12 </head>
13 <body dir="rtl">
14 <h1> תאריך ושעה </h1>
15 <% =mytime %>
16 </body>
17 </html>
```

הגיע הזמן לשלב קצת **תכנות** באינטרנט - שורת הכותרת מצינת מהי שפת התכנות. התגיות של <% %> מגדירות איזור של קוד שאינו מובן לדפדפן.

בדף מסוג זה, אפשר 'לשתול' קוד בין תגיות של html. כמו למשל בשורה 15.

שורות 2 ו-8 תוחמות איזור של קוד בשפת C#

לאחר שנריץ את הדף הנ"ל על השרת, הוא יתרגם את התוצאה לדף html רגיל, וזה הדף שיוצג ללקוח. באו ננסה להריץ את הדף ונראה מה קורה.

כעת ננסה לראות את התוצאה של ההרצה על ידי בחינה של הקוד (view source)

השרת 'מחולל' את הקוד שנמצא בין התגיות של ה- % ויוצר את הדף שמתאים לתצוגה בדפדפן.

כדי להבין מה בעצם קרה כאן, עלינו לעבור על מספר מושגים שקשורים לתקשורת שמתרחשת בין השרת והלקוח.

נסתכל תחילה על הדף שנקרא:

http://www.lamed-oti.com/school/tmy/ya_cs/internet/internet_overview.pdf

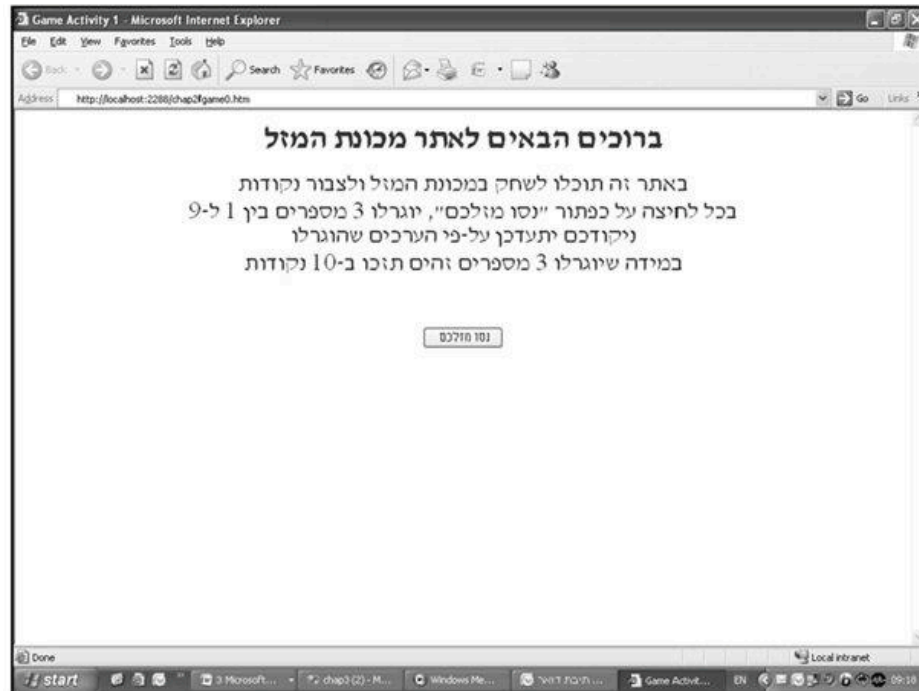
עצמים	מופע של המחלקה של (Namespace)	תיאור	פעולות
Request	HttpRequest	עצם זה מטפל בבקשת HTTP של לקוח.	QueryString Form הם מאפיינים של המחלקה HttpRequest
Response	HttpResponse	עצם זה מטפל בהכנת תגובות HTTP ללקוח.	Write Redirect
Application	HttpApplicationState	עצם מנהל מצב של יישום	אתחול, השמה, אחזור
Session	HttpSessionState	עצם מנהל מצב של לקוח	אתחול, השמה, אחזור הפעולה Abandon המאפיין Timeout

Request אלה הם הערכים שנשלחים מהלקוח לשרת (או כ- form או כמחרוזת שאילתה) - זהו אוסף של ערכים.

Response - בנית התשובה ללקוח (יצירת ה- html) בעל פעולות write ו- redirect

שני העצמים הבאים מיועדים לשמירת מצב. אמרנו שפרוטוקול האינטרנט אינו מכיל שמירת מצב מובנה. העצם application למשל יכול לשמור אינפורמציה על כמה משתמשים כרגע גולשים באתר(על היישום כולו). session ישמור נתונים פר לקוח - ישנו לדוגמה מאפיין של session בשם Timeout שבו אפשר למלא כמות דקות שאם עברו, יפוג תוקפו. (את שניהם נראה יותר בפרוט בהמשך).

נעקוב בעזרת הפרוייקט של **מכונת המזל**, אחרי כל הדברים שנכתבו לעיל. תחילה ניצור דף HTML שנראה כך (טופס) - Game1.html - בטופס הזה במקום של ה- action יהיה כתוב דף אחר שהוא מסוג aspX בשם: Game1.aspx- דף זה יחולל את התוצאה של ההגרלה שתופיע חזרה ונרחיב עליו בהמשך (שימו לב לקוד מתחת למסגרת)



```
<!-- Game1.htm ----->
<html>
<head >
  <title>Game Activity 1</title>
  <link rel = "Stylesheet" type="text/css" href = "Chap3.css" />
</head>

<body dir="rtl">
  <form id="form1" action="Game1.aspx" >
  <div>
    <h1>ברוכים הבאים לאתר מכונת המזל </h1>
  <p>
```

כל לחיצה על הכפתור תגרום להגרלה חדשה.

איזה פעולות האתר יאפשר?



nta
מנהל המכס והמע"מ

דרישות מהפרויקט המלווה

- משתמשים יכולים
 - לאחזר דפי מידע
 - להירשם
 - לשחק ולקבל נתונים על המשחק שלהם
 - לעדכן פרטים
 - להיכנס בצורה מאובטחת
- מנהלים
 - לאחזר פרטי משתמשים (דוחות שונים)
 - לעדכן ולבטל משתמשים

לשמור למשל משחקים ששחקו על ידי אותו שחקן. אפשר לקבל משתמשים שאינם רשומים וכאלה שרשומים. אלה שאינם רשומים, יכולים לראות נתונים כלליים ולא לשחק.

המנהלים יוכלו לראות פרטים של המשתמשים הרשומים וגם לעדכן.

נראה כיצד אפשר למשל לנהל את המשחק בעזרת הדרישות ותרגומן לקוד ממשי:

1. למשתמש יש בהתחלה 0 נקודות.
2. כל משתמש בכל משחק מקבל באופן אוטומטי 2 נקודות על זה ששיחק.

3. בנוסף, יש למשתמש אפשרות לשלוח את שמו בתיבת טקסט או לא. אם ישלח, יקבל בונוס של 2 נקודות.
4. אם המספרים שהוגרלו, שלושתם זהים, יתוּספו למשתמש 10 נקודות.

כעת חזרה ל- Game1.aspx. (דף דינמי)

הקוד שלנו בדף זה צריך להיות מוגדר בשני חלקים: (1) חלק ביצועי של הגרלת המספרים וחישוב הניקוד המתאים, ו- (2) החלק שיחזיר את התצוגה למסך המשתמש.

(1) נתחיל עם הקוד הביצועי - קוד זה יראה מוכר לכם, כן תוכנית סי שרפ. לפני הקוד תהיה שורה שמגדירה את שפת התכנות של הקוד:

```
<%@ Page Language="C#" %>
```

אחת הדרכים להגדיר תסריט היא להשתמש בתגים <script> ו- </script> של שפת הסימון HTML. לתג זה ניתן להגדיר מאפיין runat מצייין שהקוד הזה מורץ בשרת. בתסריטים מסוג זה נשתמש כדי להגדיר משתנים ופונקציות שתחום ההכרה שלהם הוא בכל הדף. תסריטים אלו יכתבו לפני התג הפותח <html>.

לדוגמה, את קטע הקוד בשפת C# לחישוב הניקוד נרשום כך:

```
<script runat="server">
int n1, n2, n3, points; // הגדרת משתנים מטיפוס שלם לאחסון המספרים שיוגרלו והניקוד
public void Page_Load()
{
    Random rnd=new Random(); // הגדרת עצם ליצירת מספר אקראי
    n1 = rnd.Next(1,10); // הגרלת מספרים אקראיים
    n2 = rnd.Next(1,10);
    n3 = rnd.Next(1,10);
    // חישוב הניקוד
    if ((n1 == n2) && (n2==n3))
        points = 10; // אם שלושת המספרים זהים - הניקוד הוא 10
    else
        points=2; // אחרת - הניקוד הוא 2
}
</script>
```

האירוע Page_Load

שימו לב, המשפטים להגרלת שלושה מספרים ולחישוב הניקוד נרשמו בתוך פעולה בשם Page_Load. למעשה, זהו **אירוע** (event) המתרחש בכל פעם שמתקבלת בקשה לטעינת הדף – בעקבות לחיצה על הכפתור 'נסו מזלכם', או בעת בקשה ראשונית לטעינת הדף. כאשר מתרחש האירוע הזה, המערכת מבצעת את הקוד הרשום בתוך האירוע Page_Load.

מהו אירוע? תכנית מורצת צריכה להגיב לסדרה של אירועים – את קצתם יוזמת התכנית (למשל, היא מבקשת מן המשתמש להכניס נתונים) ואת האחרים יוזם המשתמש (למשל,

המשתמש לוחץ עם העכבר על מקום או אייקון במסך, וכתוצאה מזה נפתחת אפליקציה - כללית זה נקרא 'תכנות מונחה אירועים'. משייכים לכל דבר על המסך שרוצים, קטע תוכנה שיופעל כאשר לוחצים על הדבר. שיטה זו מאפשרת תיכנות דינמי.

במקרה הנ"ל השרת מקבל בקשת HTTP ואז הוא מריץ את דף ה-ASP ואז מתבצעת הפעולה של Page_load.

איך נציג את התוצאה של הרצת הקוד ללקוח?