

גירסה 1.00 - 10.3.2000

# ASP – Part 1

מסמך זה הורד מהאתר <http://underwar.livedns.co.il> אין להפיץ מסמך זה במדיה כלשהי, ללא אישור מפורש מאת המחבר. מחבר המסמך איננו אחראי לכל נזק, ישיר או עקיף, שיגרם עקב השימוש במידע המופיע במסמך, וכן לנכונות התוכן של הנושאים המופיעים במסמך. עם זאת, המחבר עשה את מירב המאמצים כדי לספק את המידע המדויק והמלא ביותר.

כל הזכויות שמורות לניר אדר

Nir Adar

Email: [underwar@hotmail.com](mailto:underwar@hotmail.com)

Home Page: <http://underwar.livedns.co.il>

במסמך זה נציג את שפת ה-ASP (Active Server Pages). ידע נדרש להבנת המסמך: שליטה בשפת ה-HTML על בוריה, ידע בסיסי בשפת תכנות כלשהי.

על מנת להשתמש ב-ASP יש צורך בשרת שתומך ב-ASP (בד"כ שרתים מבוססים על Windows NT או Personal Web Server עבור Windows 98).

כאשר HTML היה התקן היחיד, דפי web עבדו רק בכיוון אחד. לא הייתה שום אפשרות לשלוח מידע בחזרה אל השרת. מצב זה הגביל מאוד את כותבי האתרים, שרצו לקבל מידע מהמבקרים בהם, ולהתאים את האתר באופן אישי אל המבקרים השונים. כדי לטפל בבעיה, הוכנסו תחילה forms ו-CGI. CGI מאפשר לשרת לפנות לתוכנה אחרת שרצה במחשב, שתבצע פעולות עם המידע שהתקבל. עם זאת, בפיתרון זה הייתה בעיה. הבעיה של שרתי CGI הייתה שאם חמישה אנשים שונים היו ממלאים את אותו הטופס, חמישה עותקים שונים של התוכנה המטפלת בטופס היו נטענות לזיכרון השרת. זיכרון רב בשרת היה נצרך, וכך CGI האט את פעולת השרת, ואף לעיתים גרם לקריסתו. עקב הבעיות ב-CGI נוצר תקן חדש ISAPI. תקן זה פותר כמה מהבעיות שהוצגו ב-CGI. שרת כזה אינו מריץ עותק נפרד של תוכנה בכל פעם שמישהו שולח מידע. למרות זאת, ISAPI עדיין לא פתר את כל הבעיות. עדיין על המתכנת לכתוב תוכניות וממשקים מורכבים. יצירת דפים בעזרת ISAPI לוקחת זמן רב. לאחריהם הופיעו מספר פתרונות חדשים. אחד מהם הוא ASP, שנוצר במטרה לפתור את הבעיות שהוצגו על ידי הפתרונות הקודמים. בעזרת ASP, במקום ליצור תוכניות נוספות שירוצו במקביל לשרת ה-WEB, אליו יופנו הנתונים שמגיעים, הקוד נמצא בתור דפי ה-HTML עצמם. יתרון נוסף גדול של ASP הוא תמיכה בדפדפנים. הדף המגיע אל הדפדפן הוא HTML פשוט ללא תגיות של ASP. כתוצאה מכך, אין צורך שהדפדפן יתמוך ב-ASP באופן מיוחד. כל דפדפן תומך ב-ASP. התמיכה ב-ASP צריכה להיות רק בצד השרת. כמו כן, בעזרת ASP, ניתן לשלוח לכל דפדפן קוד HTML שיתאים אליו בצורה אופטימלית. ניתן לזהות את דפדפן הלקוח ולהתאים את התכנים אליו.

נדגים דף של ASP :

```
<HTML>
<HEAD>
<TITLE>Simple ASP Page</TITLE>
</HEAD>
<BODY>
<H1>Simple ASP Page</H1><BR>
<% If Time >= #12:00:00 AM# And _
    Time < #12:00:00 PM# Then %>
<H2>Good Morning</H2>
<% ElseIf Time >= #12:00:00 PM# AND _
    Time < #6:00:00 PM# Then %>
<H2>Good Afternoon</H2>
<% Else %>
<H2>Good Evening</H2>
<% End If %>
<H2>Welcome to my page</H2>
</BODY>
</HTML>
```

<%> ו >% הם הפותחים והסוגרים המציינים את תגי ה-ASP. סקריפט זה הינו סקריפט פשוט. הדף מציג Good Morning במקרה שהשעה היא בין 12:00 בלילה ל 12:00 בצהרים, מציגה Good Afternoon במקרה שהשעה היא בין 12 ל 6 אחרי צהרים ואחרת מציג Good Evening. השפה שבה כתוב הסקריפט היא VB Script. ההבדל בין ASP לבין סקריפטים רגילים הוא שבמקרה של סקריפטים שבצד הלקוח הסקריפט מועבר על הלקוח, שם הוא מתבצע, ואילו סקריפט של ASP מתבצע בשרת, והלקוח מקבל דף HTML נקי ללא סקריפטים בתוכו.

כאשר נשמור קובץ ASP, נזכור לתת לו את הסיומת ASP. את הקבצים נשמור בספרייה בה מותקן ומקונפג השרת. בדרך כלל ספרייה זו תהייה :

C:\inetpub\wwwroot

אבל ספרייה זו עלולה להשתנות בהתאם להגדרות בשרת. על מנת לגשת, למשל, לקובץ בשם 1.asp שנמצא בספרייה זו, נכתוב בדפדפן את הכתובת הבאה :

<http://localhost/1.asp>

או לחילופין :

<http://127.0.0.1/1.asp>

שתי צורות כתיבה אלו מתכוונות לפנייה לשרת שמותקן על המחשב המקומי. אם השרת נמצא על מחשב מרוחק, נכתוב פשוט את כתובתו כדי לראות את הדף המבוקש.

נעבור לדוגמא נוספת :

```
<HTML>
<HEAD>
<TITLE>Hello</TITLE>
</HEAD>
<BODY>
<CENTER>
<% Dim ftSize %>
<% For ftSize = 1 to 7 %>
<FONT SIZE= <%=ftsize %>>
Hello!!!<BR>
</FONT>
<% Next %>
</CENTER>
<H3>It is <% WeekdayName(Weekday(Date)) %>, <%=Date %>.
The time is now <%=Time %>.</H3>
</BODY>
</HTML>
```

בדף זה יראו Hello!!! בגדלים שונים, מקטן לגדול, ולאחר מכן יוצגו היום בשבוע, התאריך והשעה.

Dim זו פקודה בעזרתה אנו מגדירים בדוגמא זו משתנה בשם ftSize. For זוהי לולאה שבדוגמא פועלת 7 פעמים ובכל פעם מגדילה את הערך של ftSize ב.1

בהמשך למסמך זה נרחיב על פקודת For ועל הפקודות האחרות שהדגמנו בדוגמא זו.

<FONT SIZE= <%=ftsize %>> זהו שילוב של HTML רגיל ושל ASP. FONT SIZE היא פקודה של HTML. <%=ftsize %> זהו תג של ASP. במקום התג יוצב המשתנה ftsize.

הערות

ניתן להוסיף הערות בקוד. עושים זאת על ידי הסימן ' בתחילת ההערה. דוגמא:

```
<%
' this code loops seven time
' and print the same line on the page
' at seven different font sizes.
Dim ftSize
For ftSize = 1 to 7
%>
<FONT SIZE= <%=ftsize %>>
Hello!!!<BR>
</FONT>
<% Next %>
```

סגנון כתיבה

בתגי ASP לא משנה גודל האותיות. הביטויים הבאים זהים אחד לשני:

```
<% FOR FTSIZE= 1 TO 7 %>
<% For fsize = 1 to 7 %>
<% For FtSiZe = 1 tO 7 %>
```

ניתן לפצל שורת הוראה ל-2 חלקים באמצע (לצורך קריאה נוחה יותר) ע"י הסימן \_.  
דוגמא:

```
<% If Weekday(Date) > 3 and _
Weekday(Date) < 6 then
%>
```

משתנים

נתחיל בדוגמא לדף המשתמש במשתנים:

```
<HTML>
<%
Dim Cost, Tax, Total
Cost = 40
Tax = 2
Total = Cost + Tax
%>
</HTML>
```

בדוגמא זאת יצרתי שלושה משתנים בשם Cost, Tax, Total. Cost הצבתי 40,  
Tax הצבתי 2. Total מקבל את הערך של סכום המשתנים.

מה עושים עם משתנים?  
הנה דוגמא לשימוש במשתנה:

```
<HTML>
```

```
<%
```

```
Dim Cost, Tax, Total
```

```
Cost = 40
```

```
Tax = 2
```

```
Total = Cost + Tax
```

```
%>
```

```
<%=Total%>
```

```
</HTML>
```

אם תנסה לראות את הקוד של הדף דרך הדפדפן, תראה את הדבר הבא:

```
<HTML>
```

```
42
```

```
</HTML>
```

אל המשתמש מגיע דף HTML רגיל ללא הסקריפטים שכתבנו בתוכו.

דוגמא נוספת לשימוש במשתנים:

```
<HTML>
```

```
<%
```

```
Dim First, Last, Whole
```

```
First = "Bill"
```

```
Last = "Gates"
```

```
Whole = First & Last
```

```
%>
```

```
<%=Whole %>
```

```
</HTML>
```

במסמך יופיע BillGates.

כאשר מחברים רצפי אותיות, בניגוד למספרים, לא משתמשים בסימן +, לדוגמא

```
Total = Cost + Tax
```

אלא בסימן &, לדוגמא

```
Whole = First & Last
```

נניח שאנו רוצים לשים רווח בין המילים, נערוך את השינוי הבא בקוד:

```
<HTML>
```

```
<%
```

```
Dim First, Last, Whole
```

```
First = "Bill"
```

```
Last = "Gates"
```

```
Whole = First & " " & Last
```

```
%>
```

```
<%=Whole %>
```

```
</HTML>
```

נדגים עכשיו נושא חדש, שימוש בפונקציות של VBScript (שזו השפה בה אנו משתמים בASP).

```
<HTML>
<%
Dim CurrentDate
CurrentDate = Date
%>
The current date is <% = CurrentDate %>
</HTML>
```

Date זוהי פונקציה של VBScript המחזירה את התאריך הנוכחי. כידוע, פונקציה גם יכולה לקבל ערך מהפונקציה שקראה לה. נדגים את הפונקציה Weekday, המקבלת את התאריך ומחזירה ערך מספרי בין 1 ל 7 המציין את מספרו של היום בשבוע.

```
<HTML>
<%
Dim ThisDay, ThisDate
ThisDate = Date
ThisDay = Weekday(ThisDate)
%>
It is day number <% = ThisDay %>
</HTML>
```

It is day number 5

במסמך שיוצג על המסך יהיה כתוב

במקרה שהיום יום חמישי.

מספרים אקראיים

```
<HTML>
<%
Dim RandNum
Randomize
RandNum=Rnd
%>
<% =RandNum %>
</HTML>
```

על הדף יוצג מספר אקראי כלשהו בין 0 ל1. כל פעם שתעשה Refresh למסמך יופיע מספר אחר.

הפונקציה Randomize מבצעת חישוב בעזרת ערך משעון המערכת. מה שמעניין אותי זה שהיא מאפשרת לנו להשתמש בRnd כדי להשיג מספר אקראיים. הפונקציה rnd מחזירה מספר "אקראי" בין 0 ל1. אם לדוגמא אנחנו רוצים מספר אקראי בין 1 ל6 נוכל להשתמש בדרך הכתיבה הבאה:

```
RandNum = (rnd * 6) + 1
Rnd*6 נותן לנו מספר אקראי בין 0 ל5, ואז אנו מוסיפים לו 1 ומקבלים מספר בין 1 ל6.
```

אנו מקבלים מספרים עשרוניים עם הרבה ספרות אחרי הנקודה. במקרה שאנחנו רוצים מספרים שלמים אנו משתמשים בפונקציה int, המקבלת מספר עשרוני ומחזירה רק את החלק השלם שלו, דוגמא:

```
int(43.3453445)
יחזיר 43.
```

```
RandNum = int(rnd * 6) + 1
יחזיר מספר שלם בין 1 ל6.
```

פונקציות מתמטיות

הפונקציות המתמטיות atn, cos, sin, tan ניתמכות על ידי VBScript וASP. הפונקציות מקבלות ערך, מבצעות את החישוב הדרוש עליו ומחזירות את תוצאת החישוב. למשל:

```
<%=sin(2)%>
```

פונקציות לטיפול במחרוזות

הפונקציה הראשונה שנציג לטיפול במחרוזות היא InStr.

המבנה הכללי של InStr:

(מחרוזת\_לחיפוש, מחרוזת\_עיקרית) = InStr(משתנה)

הפונקציה מקבלת מחרוזת עיקרית ומחרוזת שניה. הפונקציה מחפשת את ההופעה הראשונה של המחרוזת לחיפוש במחרוזת הראשונה, ומחזירה את המקום שלה במחרוזת הראשונה.

דוגמא:

<%

Dim BigString, FindString, Position

BigString = "I do not like green people and dogs."

FindString = "people"

Position = InStr(BigString, FindString)

%>

הערך של Position יהיה 21, מכיוון שהמילה People הופיעה 21 אותיות מהתחלת המחרוזת הגדולה.

הפונקציה הבאה שנראה היא Len.

המבנה של Len:

(מחרוזת) = Len(משתנה)

Len מקבלת מחרוזת ומחזירה את מספר האותיות במחרוזת.

נביט בסדרה של 3 פונקציות: LTrim, RTrim, Trim.

פונקציות אלו מקבלות מחרוזת ומחזירות מחרוזת.

LTrim מוחקת את כל הרווחים מצידה השמאלי של מחרוזת.

RTrim מוחקת את כל הרווחים מצידה הימני של מחרוזת.

Trim מוחקת את כל הרווחים משני צדי המחרוזת.

הסדרה הבאה היא של שלוש פונקציות מאוד שימושיות בVBScript:

Left, Mid, Right.

הפונקציות האלו מאפשרות לך לקחת מחרוזת ולחלק אותה לחלקים קטנים יותר. הצורה של שלושת הפונקציות היא:

(מספר, מחרוזת) = Left(משתנה)

(מספר, מחרוזת) = Right(משתנה)

(מספר, התחלה, מחרוזת) = Mid(משתנה)

Left מקבלת שני פרמטרים, מחרוזת ומספר. היא מחזירה מחרוזת שהיא מורכבת ממספר האותיות השמאליות של המחרוזת המקורית.

דוגמא:

Left("Dog Bones", 3) תחזיר לנו "Dog".

Right פועלת באותה צורה, רק מהצד הימני של המחרוזת.  
Mid מקבלת 2 ארגומנטים. הארגומנט הראשון אומר מהו מספר התו הראשון ממנו יש לגזור את המחרוזת המוחזרת, ואילו num אומר כמה תווים מהתו הראשון של לקחת למחרוזת המוחזרת.

פונקציות הקשורות לזמן:

Date מחזירה את התאריך הנוכחי.  
Time מחזירה את השעה הנוכחית.  
Now מחזירה את התאריך ואת השעה הנוכחית ביחד.  
Weekday מקבלת תאריך כארגומנט ומחזירה לנו את מספר היום בשבוע.  
WeekdayName מקבלת מספר בין 1 ל 7 ומחזירה את שם היום.

הסתעפויותif הפקודה

בעזרת הפקודה if ניתן לגרום למחשב לקבל החלטות בהתאם לנתונים שונים.  
המבנה של פקודת if:

if תנאי Then תנאי if

דוגמא:

```
<HTML>
<%
Dim Temperature, Boiling
Temperature = 100
%>
The water is heated for 5 Minutes.<P>
<%
If Temperature = 100 Then
  Boiling = True
%>
The water is boiling!<P>
<% End If %>
</HTML>
```

בדוגמא זו אנו מגדירים משתנה Temperature ונותנים לו את הערך 100.  
לאחר מכן אנו אומרים לשרת: "אם הערך של Temperature שווה ל-100, אז תגדיר את Boiling בתור True ותציג את השורה The water is boiling! בדף.  
תוכל לשנות את הערך של Temperature ואז לראות שהשורה לא תופיע בדף.

ישנם אופרטורים נוספים שיכולים להיות במשפטי if, חוץ מ=.  
דוגמא:

```
<% If Grade > 90 Then %>
You get an A<P>
<% End If %>
```

בדוגמא זו אמרנו לשרת שאם משתנה בשם Grade גדול מ90, ייכתב המשפט  
.You get an A

נביט ברשימה של האופרטורים השונים שניתן להשתמש בהם בפקודת if:

משמעות	סימן
שווה	=
גדול מ	>
קטן מ	<
גדול או שווה ל	>=
קטן או שווה ל	<=
שונה מ	<>

ניתן לחבר שני ביטויים בעזרת And או Or. דוגמא:

```
<% If Grade >= 80 And Grade < 90 Then %>
```

התנאי יתקיים בתנאי ש grade גדול מ80 וקטן מ90.

### לולאות For.. Next

בעזרת For ניתן להפעיל לולאה מספר פעמים. הדגמנו את For מעט קודם לכן,  
וכעת נראה פקודה זו ביתר פירוט.

דוגמא:

```
<HTML>
<%
Dim Counter, RandNum
Randomize
For Counter = 1 To 5
  RandNum = Int(Rnd*6) + 1
  %>
Number #<% =Counter %> is <% RandNum %><P>
<%
Next
%>
</HTML>
```

יודפס על המסך משהו בסגנון :

Number #1 is 4  
 Number #2 is 5  
 Number #3 is 2  
 Number #4 is 2  
 Number #5 is 3

המספרים השניים יהיו שונים בכל ריצה של הסקריפט.

בגזרת For יצרתי לולאה שפעלה 5 פעמים ובכל פעם בוצע הקטע שבין For next.

לולאה לא חייבת להתחיל ב1. ניתן לדוגמא לעשות לולאה כזו :

For Counter = 123 To 500

Do.. Loop

מבנה נוסף של לולאה זהו לולאות Do.. Loop

דוגמא :

```
<HTML>
<%
  Dim Counter1, Counter2, randnum
  Randomize
  Counter1 = 0
  Counter2 = 0
  Do while Counter1 < 4
    randnum = int(rnd * 2) + 1
    if Randnum = 1 Then
%>
I rolled 1!!!<BR>
<%
  Counter1 = Counter1 + 1
  Else
%>
I rolled 2
<%
  Counter2 = Counter2 + 1
  End If
  Loop
%>
Counter 1 = <% =Counter1 %> <BR>
Counter 2 = <% =Counter2 %> <BR>
```

השורה הזאת אומרת :

```
Do while Counter1 < 4
```

כל עוד Counter1 קטן מ-4, תבצע את הקטע בין ההוראה הזו להוראה Loop.

בניגוד לfor, לולאת Do.. Loop לא מגדילה אוטומטית את המשתנה המביא לסיומה ועל המתכנת לדאוג שהלולאה תגיעה לסיומה בשלב כלשהו. (על ידי הגדרת המשתנה בתוך גוף הלולאה).

אם נרצה במהלך ריצת הלולאה, לצאת מהלולאה ולסיים אותה לגמרי, נוכל לבצע זאת בעזרת הפקודה Exit :

```
For Count = 1 To 100
```

```
...
```

```
If Temp > Temp2 Then Exit For
```

```
...
```

```
Next
```

בעזרת Exit For סיימנו לולאת For באמצע. באותה שיטה בדיוק בעזרת הפקודה Exit Do ניתן לסיים לולאת Do.. Loop.

מערכים

מערך זוהי דרך להגדיר קבוצה של משתנים בהגדרה אחת.  
הגדרת מערך נראית למשל כך:

```
Dim Names(20)
```

הגדרנו למעשה 20 משתנים. לכל המשתנים שם זהה Names ומשתמשים במספר כדי לזהות אותם, לדוגמא:

```
Names(7)= "Bill Gates"
```

שמנו במשתנה השביעי את הערך "Bill Gates".

דוגמא:

```
<%  
Dim CurName, Found  
Found = FALSE  
For CurName = 1 To 20  
  If Names(CurName) = "Bill Gates" Then  
    Found = True  
    Exit For  
  End if  
Next  
If Found = True Then  
  %>  
  I found Bill Gates!!<P>  
  <% Else %>  
  I did not find Bill Gates.<P>  
  <% end if %>
```

בדוגמא זו הדגמנו סריקה של המערך, במטרה לחפש שם מסוים.  
אם נמצא השם, תופיע ההודעה I found Bill Gates!! אם לא נמצא השם תופיע  
ההודעה I did not find Bill Gates.

ניתן ליצור VBScript מערכים דינמיים, דוגמא, מגדירים מערך על ידי  
Dim Codes()

בהמשך כאשר נחליט כמה אלמנטים אנו צריכים, נשתמש בפקודה ReDim.

```
ReDim Codes(15)
```

במקרה שאנו רוצים להגדיל אחר כך שוב את המערך, נכתוב

```
ReDim Preserve Codes(20)
```

וכך נגדיר את המערך מחדש בתור מערך של 20 משתנים.

הפקודה Preserve אומרת לשרת שעליו לשמור את הנתונים של המערך הקודם. אם לא נרשום את הפקודה הזו כל הערכים שהיו במערך לפני שינוי גודלו ימחקו.

ניתן להגדיר גם מערכים דו ממדיים, תלת ממדיים וכך הלאה, בצורה הבאה:  
Dim GameBoard(8, 8)

ואז ניתן לגשת למשתנה מסוים בצורה

GameBoard(4,4) = 20

ניתן ליצור פרוצדורות משלך בVBScript בעזרת מילת המפתח Sub:

```
Sub MyProcedure ()
```

```
...
```

```
End Sub
```

פרוצדורות אינן מחזירות ערך.

ניתן ליצור גם פונקציות חדשות בצורה הבאה:

```
Function FunctionName (Value1, Value2)
```

```
Dim ReturnValue
```

```
...
```

```
FunctionName = ReturnValue
```

```
End Function
```

קריאה לפונקציה היא כמו קריאה לפונקציה שבאה מובנית בVBScript.

זהו סוף החלק הראשון בסדרת המסמכים בנושא ASP. את ההמשך ניתן למצוא באתר <http://www.underwar.livedns.co.il>.

EOF