

10

שיעור

משתנים

עד אתה התרכזנו יותר ביצירת ממשק המשתמש, כלומר בחלק של התוכנית שאותו המשתמש רואה ובעזרתו הוא משתמש בתוכנית. כעת נעבור יותר לכתיבת הקוד.

מהו משתנה?

משתנה, באנגלית variable, מאחסן נתונים. נתונים אלו יכולים להיות מספרים, מחרוזות, תאריכים ועוד. כדי ליעל את השימוש במשתנים, ויזואל בייסיק ורוב שפות התכנות נותנות לכם אפשרות להצהיר על המשתנים, כך שהם יוכלו לקבל רק סוג מסוים של נתונים. למשל אתם יכולים להצהיר על משתנה שיכיל רק אותיות (תווים).

אומנם ויזואל בייסיק לא מכריחה את המשתמש להצהיר על המשתנים בהם הוא משתמש, אבל מומלץ מאוד לעשות כן, ממספר סיבות:

1. אם התוכנית שלכם, מסובכת ומורכבת, היא תפעל יותר מהר כאשר כל משתנה תופס בזיכרון את השטח הדרוש לו ולא שטח גדול יותר, שמבזבז משאבים, בזמן שלא מצהירים על המשנה.
2. כמו כן למען הסדר, הצהרה על כל משתנה לפני שמשתמשים בו מקילה על הבנת התוכנית, במיוחד בתוכניות ארוכות במיוחד. כעת נראה איך מצהירים על משתנים.

הצהרה על משתנים

כמו שנאמר לעיל, בויזואל בייסיק אתם לא חייבים להצהיר על משתנים, אך רצוי מאוד שתעשו זאת. מצהירים על משתנים בעזרת הפקודה Dim. למשל:

`Dim StudentName As String`

בואו ננתח את חלקי ההצהרה:

Dim – אומרת שכאן אנו הולכים להצהיר על משתנה.

StudentName – זהו שם המשתנה שנתנו למשתנה, ניתן לבחור כמעט כל שם למשתנה. הכללים לבחירת שמות למשתנים, דומים לכללים לבחירת שמות לבקרים, המוזכרים בשיעור השלישי.

As String – זהו סוג המשתנה, כאן אנו הצהרנו כי המשתנה בשם StudentName, הוא מסוג מחרוזת (String), כלומר משתנה זה יכול להכיל רק תווים.

בואו נראה עוד דוגמא:

`Dim ID As Integer`
`ID = 8092`

בדוגמא זו הצהרנו על משתנה בשם ID, שהוא מסוג Integer (מספרים שלמים בלבד), בשורה השניה של הקוד הצבנו למשתנה ID, את הערך 8092.

אם היינו מציבים למשתנה את הערך "דוקטור ויבי", למשל, היינו מקבלים שגיאה, כי המשתנה ID הוצהר כמשתנה היכול לקלוט רק מספרים שלמים ולא מחרוזות (תווים).

גם תכונות (Properties), אותן הכרתם בפרקים קודמים, הן משתנים בויזואל בייסיק, אך עליהן אין צורך להצהיר.

לדוגמא:

תפתחו פרויקט חדש בויזואל בייסיק על ידי בחירת New מהתפריט File (למשל), תוסיפו כפתור פקודה בשם Command1 לטופס (Form1). לחצו לחיצה כפולה על כפתור הפקודה Command1, ובאופן אוטומטי נוצר הנוהל Command1_Click.

הוסיפו כעת את הקוד הבא לנוהל:

```
1 Private Sub Command1_Click()
2 Dim ButtonName As String
3 ButtonName = "Doctor VB"
4 Command1.Caption = ButtonName
5 End Sub
```

בקוד זה הצהרנו בשורה 2, על משתנה חדש בשם ButtonName שהוא מסוג מחרוזת. בשורה 3 הצבנו למשתנה זה את המחרוזת "Doctor VB".

בשורה 4 הצבנו לתכונה Command1.Caption, שהיא בעצם משתנה המכיל את הכיתוב שיופיע על גבי כפתור הפקודה, את הערך המאוכסן במשתנה ButtonName. וכעת התכונה Command1.Caption תקבל את הערך שמאוכסן בו, כלומר ערך התכונה יהיה כעת: "Doctor VB". כעת תפעילו את התוכנית, על ידי לחיצה על מקש F5 לדוגמא, לחצו על כפתור הפקודה, ותראו כי הכותרת שלו משתנה מ-Command1 ל-Doctor VB.

כלל חשוב לגבי הצבת ערכים במשתנים:

שימו לב, שכאשר מציבים ערך למשתנה מסוג מחרוזת (String) יש לכתוב את המחרוזת בין גרשיים, למשל: ButtonName = "Doctor VB".

לעומת זאת למשתנים המאכסנים מספרים יש להציב את המספרים ללא גרשיים למשל:

```
Dim LessonNumber As Integer
LessonNumber = 10
```

הכלל הזה נראה ברור מאוד, אך יכולה להיווצר בעיה במקרה בו אתם רוצים להציב למשתנה למשל מספר טלפון, אז איך יש להציב אותו? אם תציבו אותו כך:

```
Dim PhoneNumber As Integer
PhoneNumber = 06-7654321
```

תיווצר בעיה, כי ויזואל בייסיק תציב למשתנה PhoneNumber, את הערך הנוצר מ-06 מינוס 7654321 ולא את מספר הטלפון.

לכן אם אתם מאכסנים מספר טלפון עם קידומת רצוי לאכסן אותו בתור מחרוזת כך:

```
Dim PhoneNumber As String
PhoneNumber = "06-7654321"
```

אפשרות נוספת היא להציב את קידומת אזור החיוג במשתנה אחד ואת מספר הטלפון במשתנה שני, לדוגמא:

```
Dim AreaCode As Integer
Dim PhoneNumber As Integer
AreaCode = 6
PhoneNumber = 7654321
```

החיסרון בשיטה זו, הוא שאתם מאבדים את האפס שלפני ה-6 במשתנה AreaCode שבו הצבתם את אזור החיוג, לכן מומלץ בכזה מקרה להשתמש במחרוזת כמו שעשינו קודם.

סוגי המשתנים

עד אתה ראינו לצורך התרגול שני סוגי משתנים: אחד Integer שהוא משתנה היכול להכיל מספרים שלמים, והשני הוא String שהוא משתנה שיכול להכיל מחרוזות (תווים).

אך, בויזואל בייסיק ישנם סוגים נוספים של משתנים, שנעבור עליהם כעת:

דוגמא	הערכים שהוא יכול לקבל	סוג המשתנה
Dim ID As Integer ID = 28000	מספרים שלמים בלבד, בטווח שבין -32768 עד 32767	Integer
Dim Population As Long Population = 9900000	מספרים שלמים בלבד, בטווח שבין -2147483648 עד 2147483647	Long
Dim PI As Single PI = 3.14159	מספרים, כולל שברים עשרוניים בטווח שבין 1.401298E-45 עד 3.402823E38 לגבי מספרים חיוביים. ולגבי מספרים שליליים בטווח שבין: -3.402823E38 עד -1.401298E-45	Single
Dim Payment As Double Payment = 9.99E+47	מספרים, כולל שברים עשרוניים בטווח שבין 4.94065645841247D-324 עד 1.79769313486232D308 לגבי מספרים חיוביים. ובטווח שבין -1.79769313486232D308 עד -4.9406564584124D-324 לגבי מספרים שליליים.	Double
Dim Level As Currency Level = 9000000000#	מספרים בטווח שבין 922,337,203,685,477.5808 עד 922,337,203,685,477.5807	Currency
Dim Lesson As Byte Lesson=10	מספרים שלמים בלבד בטווח שבין 0 ל-255	Byte
Dim HaveMoney As Boolean HaveMoney = True	הערכים True (אמת) או False (שקרי) בלבד.	Boolean
Dim MyName As String MyName = "Yaniv"	מחרוזות באורך של עד כ-65,400 תווים.	String
Dim Today As Date Today = #12/31/2000#	ערכים של תאריכים בלבד בין 1/1/100 עד 31/12/9999	Date
Dim aValue As Variant aValue = "Name"	כל ערך שמוצב אליו. כל משתנה שלא מצהירים עליו הוא אוטומטית מסוג Variant.	Variant
Dim aValue aValue = 4192	או	
Dim obj As Object Set obj = Command1 obj.Caption = "Yaniv"	מראה מקום לאובייקט.	Object

הערות לגבי סוגי משתנים

מדוע לא להשתמש תמיד ב-Long במקום ב-Integer?

אתם בטח שואלים את עצמכם מדוע שרוצים לאחסן מספרים שלמים, לא נשתמש תמיד ב-Long במקום ב-Integer, הרי Long מכיל טווח גדול יותר? התשובה לכך היא שמשנתה מסוג Integer תופס פחות מקום בזכרון ממשנתה מסוג Long ולכן אם אנו משתמשים במספרים שלא חורגים מהערכים שיכולים

להיות מוצבים במשתנה מסוג Integer נצהיר על המשתנה כ-Integer. אם אנו נדרשים למספרים גדולים יותר, אז אין ברירה, אלא להשתמש ב-Long.

משתנים מסוג Date (תאריך)

משתנים מסוג Date משמשים לאחסון תאריכים. בוודאי שמתם לב, בדוגמא למעלה לשימוש במשתנה מסוג Date, לצורך להציב את התאריך הרצוי בין סולמיות (#), לדוגמא:

```
Dim Today As Date
Today = #1/8/2001#
```

ניתן לחילופין להשתמש במקום בסולמית בגרשיים, לדוגמא:

```
Dim Today As Date
Today = "1/8/2001"
```

ניתן להציב תאריכים גם בצורה הבאה:

```
Today = #August 21, 2001#
```

ותראו איך ויזואל בייסיק משנה את התאריך באופן אוטומטי ל:

```
Today = #21/8/2001#
```

משתנים מסוג Variant

Variant הוא סוג משתנה היכול להכיל ערכים שונים. אם לא מצהירים על סוג המשתנה, סוג המשתנה אוטומטית נקבע כ-Variant.

למשל, ההצהרה הבאה: Dim aValue זהה מבחינת ויזואל בייסיק להצהרה הבאה:

```
Dim aValue As Variant
```

משתנה מסוג Object – משתנה מיוחס

משתנה מסוג Object הוא משמש מעין מראה מקום לאובייקט אחר, כמו כפתור פקודה למשל. כדי להבהיר את העניין נשתמש בדוגמא:

פתחו פרוייקט חדש בויזואל בייסיק, והציבו כפתור פקודה בשם Command1, על הטופס. הוסיפו את הקוד הבא לעורך הקוד:

```
1 Private Sub Command1_Click()
2 Dim obj As Object
3 Set obj = Command1
4 obj.Caption = "Yaniv"
5 End Sub
```

הריצו את התוכנית, על ידי לחיצה על מקש F5 (למשל).

לחצו על כפתור הפקודה ותראו כי כותרתו משתנה מ-Command1 ל-Yaniv. כעת, נעבור על התוכנית:

בשורה 2 הצהרנו על משתנה בשם obj מסוג Object.

בשורה 3 הגדרנו כי המשתנה obj יהיה משתנה מיוחס לאובייקט Command1. משמעות הדבר היא כי כל פעולה שתעשו על המשתנה obj, תתבצע כאילו עשיתם אותה על האובייקט Command1 עצמו.

בשורה 4 הצבנו לתכונה Caption של המשתנה המיוחס obj, את המחרוזת "Yaniv". הצבה זו גרמה לשינוי ערך ה-Caption של כפתור הפקודה Command1.

שיטה זו נוחה במקרים מסוימים, שלא רוצים לשנות ערך של תכונה ישירות באמצעות קריאה לאובייקט. דוגמא למצב בו יעיל להשתמש במשתנים מיוחסים נראה בשיעור על תנאים.

משתנה מסוג Boolean – משתנה בוליאני

משתנה בוליאני הוא משתנה שיכול לקבל רק את הערכים הבאים: True (אמת) או False (שקר).

הדבר שימושי ביותר לשמירה של מצב מסוים. למשל לכל אובייקט כמעט, יש לו את התכונה Visible שהיא בעצם משתנה בוליאני, והיא יכולה לקבל את הערך True, כדי שהאובייקט יהיה מוצג, או False על מנת להסתיר את האובייקט.

טווח ההשפעה של המשתנים

משתנים מקומיים

משתנה עליו אתם מצהירים בתוך נוהל אירוע, משפיע רק על נוהל האירוע, לדוגמא:

```

1 Private Sub Command1_Click()
2 Dim SiteName As String
3 SiteName = "Doctor VB"
4 Command1.Caption = SiteName
5 End Sub

```

המשתנה במקרה שלנו SiteName קיים רק בתוך הנוהל בין ההצהרה על הנוהל בשורה 1 לסוף הנוהל בשורה 5.

אם נוסיף עוד נוהל אירוע לתוכנית למשל לכפתור פקודה נוסף בשם Command2, אז המשתנה SiteName לא יהיה קיים, עבורו, ולכן הקוד הבא:

```

1 Private Sub Command2_Click()
2 Command2.Caption = SiteName
3 End Sub

```

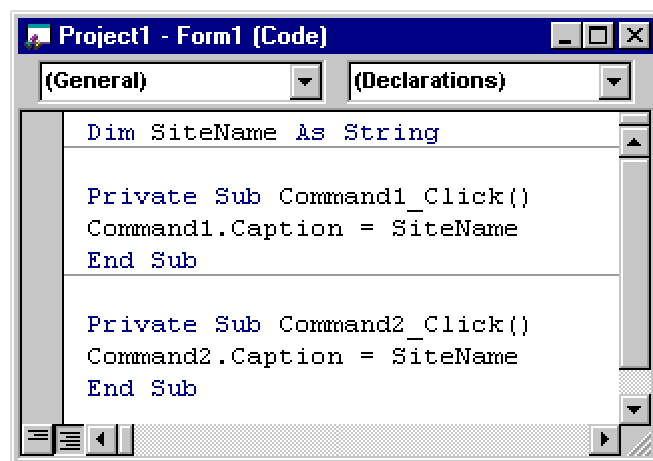
יגרום, שכאשר המשתמש ילחץ על הכפתור Command2 יוצב הערך של המשתנה SiteName ככותרת של כפתור הפקודה, אך מכיוון שהמשתנה קיים רק בנוהל Command1_Click, הדבר יגרום להצבת מחזורות ריקה בתור כותרת כפתור הפקודה Command2. נסו את זה בעצמכם.

משתני ביניים:

כדי להצהיר על משתנים שיהיו נגישים מכל נוהל בטופס, יש להצהיר עליהם באזור ההגדרות הקיים בראש עורך הקוד.

איך מצהירים על משתמש באזור ההגדרות?

עושים זאת כך: בחלון הקוד בוחרים בתיבת האובייקטים (משמאל – ראו באיור הבא) ב-General ומתיבת הפרוצדורות (מימין) בוחרים ב-Declarations. כעת הסמן יעבור לראש עורך הקוד, שם אתם כותבים את המשתנים שיהיו נגישים לכל הפרוצדורות בטופס:



נצהיר כעת על המשתנה SiteName שם, באזור ההגדרות.

משתנה זה עליו מצהירים באזור ההגדרות של טופס, נקרא משתנה ביניים. כי ניתן להשתמש בו מכל הנהלים בטופס, אך לא מנוהלים מחוץ לטופס.

לגבי משתני ביניים, רצוי להצהיר עליהם באמצעות Private במקום באמצעות Dim, כדי להדגיש כי אילו משתני ביניים. הדבר לא הכרחי אך רצוי.

לדוגמא:

```
Private SiteName As String
```

משתנים פומביים

אם ברצונכם להצהיר על משתנה שיהיה נגיש מכל מקום בתוכנית, יש להצהיר עליו כמשתנה פומבי. ההצהרה על משתנה פומבי מתבצעת באזור ההגדרות של הטופס, באמצעות ההצהרה Public, למשל:

```
Public SiteName As String
```

עכשיו המשתנה יהיה זמין מכל טופס או מודול. למשל אם ברצונכם לקרוא ערך מהמשתנה SiteName הנמצא ב-Form1 מנוהל ב-Form2, עליכם לקרוא אותו כך:

```
MyValue = Form1.SiteName
```

כלומר, יש להקדים לשם המשתמש את שם הטופס לו הוא שייך.

משתנים גלובליים

כדי ליצור משתנה שיהיה נגיש מכל מקום בתוכנית, בלי שנצטרך ליחס אותו לטופס מסוים, יש להצהיר עליו כפומבי באזור ההגדרות של מודול.

פתחו פרוייקט חדש בוויזואל בייסיק, והוסיפו לו מודול על ידי בחירה ב-Add Module מהתפריט Project.

באזור ההגדרות של המודול תצהירו על המשתנה כך:

```
Public SiteName As String
```

וכעת המשתנה SiteName יהיה נגיש מכל מקום בפרוייקט, מבלי צורך להוסיף לאיזה מודול הוא שייך. קריאה למשתנה תתבצע לדוגמא כך:

```
MyValue = SiteName
```

הערה:

רצוי להשתמש במשתנים גלובליים בזהירות רבה, כי בתוכניות גדולות, אתם עשויים להצהיר על משתנה ביניים או משתנה מקומי בעלי אותו שם כמו משתנה גלובלי, וזה עלול לגרום לבעיות רבות.

אי הצהרה על משתנים

ויזואל בייסיק, לא מחייבת אתכם להצהיר כלל על משתנים. במקרה שאתם לא מצהירים על משתנה הוא נוצר כמשתנה מסוג Variant בפעם הראשונה בו אתם קוראים לו. למשל:

```
1 Private Sub Command1_Click()
2 SiteAddress = "www.dr-vb.co.il"
3 Command1.Caption = SiteAddress
4 End Sub
```

בשורה 2 מופיע לראשונה המשתנה SiteAddress, עליו לא הצהרנו, אז ויזואל בייסיק אוטומטית יוצרת את המשתנה כאילו שהצהרנו עליו כך: `Dim SiteAddress As Variant`. מדוע לא רצוי לנהוג כך?

1. בגלל שיותר יעיל להצהיר על משתנה לפי הסוג הנדרש לו (String, Integer וכו') כדי לחסוך במשאבי מערכת.
2. קשה לעקוב אחרי תחביר התוכנית, במיוחד אם היא ארוכה ועבר זמן רב מאז שכתבתם אותה.

יש ברירה אין ברירה

נגיד שהצהרתם על משתנה בשם SiteName מסוג String (מחרוזת), ואחר כך בטעות כתבתם את שם המשתנה כ-SiteNana, לדוגמא:

```
1 Private Sub Command1_Click()
2 Dim SiteName As String
3 SiteName = "Doctor VB"
4 Command1.Caption = SiteNana
5 End Sub
```

מה שקורה בקוד הבא הוא שבמקום שכותרת כפתור הפקודה Command1 תשתנה לערך השמור במשתנה SiteName, יוצר משתנה ריק חדש מסוג Variant בשם SiteNana (בשורה 4), והכותרת תהיה ריקה. לפעמים תתקשו מאוד למצוא מה מקור הבעיה בתוכנית, במיוחד שהאיות של שם המשתנה המוטעה מאוד דומה לשם המשתנה שעליו הצהרתם.

מה הפתרון?

הפתרון הוא לכתוב בראש עורך הקוד את ההצהרה: **Option Explicit**. משתנים מסוג Variant מיוצרים על כל משתנה בטופס/מודול לפני שאתם משתמשים בו. במקרה של הדוגמא הקודמת, כשהייתם מריצים את הקוד, ויזואל בייסיק היתה נעצרת על SiteName שבשורה 4 וכותבת את השגיאה הבאה: Variable not defined, כלומר המשתנה לא מוגדר, ואז הייתם שמים לב כי יש טעות באיות של שם המשתנה ומתקנים אותה. כמו שניתן לראות ההצהרה Option Explicit, חוסכת חיפוש מתיש אחרי טעויות שמקורן בשגיאת כתיב, ולכן מומלץ מאוד להשתמש בה בראש כל טופס או מודול בראש הקוד.

סיכום

בשיעור זה למדתם על משתנים, על סוגי המשתנים, אופן ההצבה וטווח ההגדרה. בכל תוכנית כמעט קיימים משתנים ולכן חשוב מאוד להבין את עקרונות העבודה עמם, שבויזואל בייסיק הם פשוטים מאוד להבנה.

שאלה לחזרה

מה קורה אם לא מצהירים על משתנה?

1. המחשב יתריע על כך שנמאס לו מתוכניות הכתובות בויזואל בייסיק.
2. יתפוס אתכם שוטר וייתן לכם קנס על אי הצהרת משתנה.
3. תקבלו הודעת כי הגיע הזמן לעבור לגרסה הבאה של ויזואל בייסיק.
4. ויזואל בייסיק תיצור את המשתנה עבורכם, בפעם הראשונה שתשתמשו בו, ואם השתמשתם ב-Option Explicit בראש הקוד, אז ויזואל בייסיק תתריע על כך שהמשתנה לא הוגדר.