

## חידון נשיאים

### איור 8-1

חידון הנשיאים הוא משחק טיפוחי על מנהיגים לשעבר של ארצות הברית. למרות שהחידון הזה עוסק בנשיאים, אתה יכול להשתמש בו כתבנית לבניית חידונים או מדרכי לימוד בכל נושא.



בפרקים הקודמים הוצגו לך לכמה מושגי תכנות בסיסיים עכשיו, אתה מוכן למשהו יותר מאתגר. תגלו שהפרק הזה דורש קפוצה מושגית מבחינת מיומנויות תכנות וחשיבה מופשטת. בפרט, תשתמש בשני משתני רשימה כדי לאחסן את הנתונים -במקרה זה, השאלות והתשובות של החידון -ותשתמש במשתנה אינדקס כדי לעקוב אחר היכן נמצא המשתמש בחידון. כשתסיים, תהיה חמוש בידע ליצור אפליקציות חידונים ואפליקציות רבות אחרות הדורשות עיבוד רשימה.

פרק זה מניח שאתה מכיר את היסודות של App Inventor שימוש ב- מעצב רכיבים לבניית ממשק המשתמש, ושימוש בעורך בלוקים כדי לציין מטפלים באירועים ולתכנת את התנהגות הרכיב. אם אינך מכיר את היסודות הללו, הקפד לעיין בפרקים הקודמים לפני שתמשיך.

אתה תעצב את החידון כך שהמשתמש ימשיך משאלה לשאלה לפי לחיצה על כפתור הבא ומקבלת משוב על התשובות.

## מה תלמד

אפליקציה זו, המוצגת באיור 8-1, מכסה:

• הגדרת משתני רשימה לאחסון השאלות והתשובות ברשימות.

• ירצף דרך רשימה באמצעות אינדקס;  
בכל פעם שהמשתמש לוחץ על הבא, תציג את השאלה הבאה.

• שימוש בהתנהגויות מותנות (אם): ביצוע פעולות מסוימות רק בתנאים ספציפיים.  
אתה תשתמש בבלוק אם כדי לטפל



התנהגויות האפליקציה כאשר המשתמש מגיע לסוף החידון.

איור 8-2. חידון הנשיאים בפעולה

• החלפת תמונה כדי להציג תמונה שונה עבור כל שאלת חידון.

## מתחילים

נווט לאתר App Inventor והתחל פרויקט חדש. תן לפרויקט את השם "PresidentsQuiz" והגדר את כותרת המסך. "Presidents Quiz" לחבר את המכשיר או האמולטור שלך לבדיקה חיה.

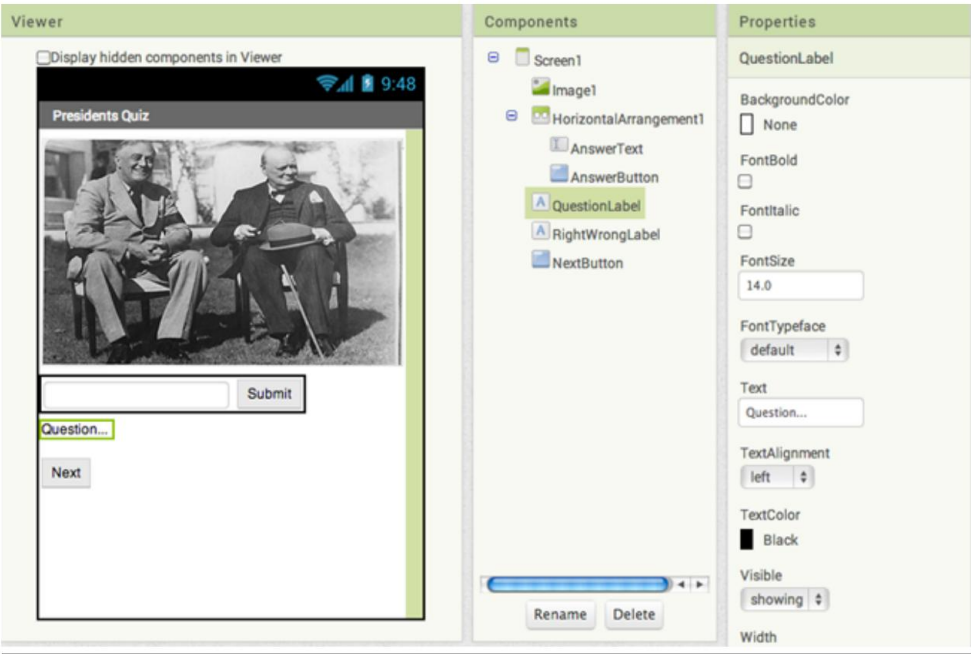
יהיה עליך להוריד את התמונות הבאות עבור החידון מה-  
אתר [appinventor.org](http://appinventor.org) למחשב שלך:

- <http://appinventor.org/bookFiles/PresidentsQuiz/roosChurch.gif>
- <http://appinventor.org/bookFiles/PresidentsQuiz/nixon.gif>
- <http://appinventor.org/bookFiles/PresidentsQuiz/carterChina.gif>
- <http://appinventor.org/bookFiles/PresidentsQuiz/atomic.gif>

אתה תטען את התמונות האלה לפרויקט שלך בסעיף הבא.

## עיצוב הרכיבים

לאפליקציית Presidents Quiz ממשק פשוט להצגת השאלות ולאפשר למשתמש לענות. אתה יכול לבנות את הרכיבים מתמונות המצב של מעצב הרכיבים המוצגת באיור 8-2.



איור 8-3. חידון הנשיאים במעבד

כדי ליצור ממשק זה, טען תחילה את התמונות שהורדת לפרויקט. ב באזור המדיה, לחץ על הוסף ובחר אחד מהקבצים שהורדת (למשל, roosChurch.gif). עשה את אותו הדבר עבור שלוש התמונות האחרות. לאחר מכן, הוסף את הרכיבים המפורטים בטבלה 8-1.

טבלה 8-1. רכיבים לאפליקציית חידון הנשיאים

מטרה	קבוצת צבעים איך תקרא לזה	סוג רכיב
התמונה המוצגת עם תמונת ממשק משתמש שאלה.		תמונה
ממשק משתמש QuestionLabel		תווית
הכנס את טקסט התשובה ואת הכפתור		
המשתמש יזין את תשובתו כאן.	ממשק משתמש AnswerText	תיבת טקסט
המשתמש לוחץ על זה כדי לשלוח תשובה.	ממשק משתמש AnswerButton	לחצן
הצג "נכון!" או "לא נכון!"	ממשק משתמש RightWrongLabel	תווית
המשתמש לוחץ על זה כדי להמשיך השאלה הבאה.	ממשק משתמש NextButton	לחצן

1. הגדר את Image1 לקובץ התמונה roosChurch.gif, התמונה הראשונה שאמורה להופיע. הגדר את הרוחב שלו ל"מלא הורה" ואת הגובה שלו ל-002.
2. הגדר את QuestionLabel.Text ל-"שאלה..." (תזין את השאלה הראשונה ב- עורך בלוקים).
3. הגדר את AnswerText.Hint ל"הזן תשובה". הגדר את מאפיין הטקסט שלו לריק. מהלך ולזוז לעבור זה לתוך Horizontal Arrangement1.
4. שנה את "Submit" ל-AnswerButton.Text והעבר אותו אל סידור אופקי. 1.
5. שנה את "Next" ל-NextButton.Text.
6. שנה את RightWrongLabel.Text לריק.

# הוספת התנהגויות לרכיבים

יהיה עליך לתכנת את ההתנהגויות הבאות:

- כאשר האפליקציה מתחילה, מופיעה השאלה הראשונה, כולל התואמת לה תמונה.
  - כאשר המשתמש לוחץ על NextButton, מופיעה השאלה השנייה. כשזו לחצו שוב, השאלה השלישית מופיעה, וכן הלאה.
  - כאשר המשתמש מגיע לשאלה האחרונה ולוחץ על כפתור Next, הראשון השאלה צריכה להופיע שוב.
  - כאשר המשתמש עונה על שאלה, האפליקציה תדווח האם היא נכונה. תקודד את ההתנהגויות האלה אחת אחת, תוך כדי בדיקה.
- הגדרת רשימות השאלות והתשובות
- כדי להתחיל, הגדר שני משתני רשימה על סמך הפריטים בטבלה 2-8: כדי להחזיק את רשימת השאלות, ו- AnswerList כדי להחזיק את רשימת התשובות המתאימות. איור 3-8 מציג את שתי הרשימות שתיצור בעורך הבלוקים.

טבלה 2-8. משתנים להחזקת רשימות שאלות ותשובות

משתנים ומקורה		
אתחול גלובלי ל- ("QuestionList") משתנים אחסן את רשימת השאלות (שנה את שמה. QuestionList)		
משתנים לאחסן ("AnswerList") תשובות (שנה את שמה. AnswerList-7)		
משתנים לאחסן הפריטים של השאלות.		

הוספת התנהגויות לרכיבים 133

מטרת מגירה	סוג בלוק
השאלות.	טקסט (שלושה מהם)
הכנס את הפרטים של רשימת התשובות.	עשה רשימה
טקסט.	טקסט (שלושה מהם)

initialize global QuestionList to

make a list

Which president implemented the 'New Deal' during the Great Depression?

Which president granted communist China formal recognition in 1979?

Which president resigned due to the Watergate scandal?

initialize global AnswerList to

make a list

Roosevelt

Carter

Nixon

איור 4-8. הרשימות לחידון

הגדרת משתנה האינדקס

האפליקציה צריכה לעקוב אחר השאלה הנוכחית כשהמשתמש לוחץ על הבא כדי להמשיך בחידון. אתה תגדיר עבורו משתנה בשם `currentQuestionIndex` זה, והמשתנה ישמש כאינדקס הן לשאלות השאלות והן רשימת תשובות. טבלה 3-8 מפרטת את הבלוקים שתצטרך לעשות זאת, ואיור 4-8 מראה מה המשתנה הזה ייראה.

טבלה 3-8. יצירת האינדקס

מטרת מגירה	סוג בלוק
משתנים החזק את האינדקס (המיקום) של השאלה/תשובה הנוכחית.	אתחול גלובלי ל ("CurrentQuestionIndex")
הגדר את הערך ההתחלתי של <code>currentQuestionIndex</code> ל-1 (הראשון שאלה).	מספר (1)

initialize global currentQuestionIndex to

1

איור 5-8. התחלת משתנה האינדקס עם ערך של 1

הצגת השאלה הראשונה

כעת, לאחר שהגדרת את המשתנים הדרושים לך, תוכל לציין את האינטראקטיביות של האפליקציה התנהגות. כמו בכל אפליקציה, חשוב לעבוד בהדרגה ולהגדיר אחת התנהגות בכל פעם. כדי להתחיל, חשוב רק על השאלות, ובמיוחד, הצגת השאלה הראשונה ברשימה בעת הפעלת האפליקציה. נחזור להתמודד עם התמונות ותשובות קצת מאוחר יותר.

אתה רוצה שקוביות הקוד שלך יעבדו ללא קשר לשאלות הספציפיות המופעלות הרשימה. בדרך זו, אם תחליט לשנות את השאלות או ליצור חידון חדש על ידי העתקה ושינוי של אפליקציה זו, תצטרך לשנות רק את השאלות בפועל בהגדרות הרשימה, ולא תצטרך לשנות אף מטפל באירועים.

אז, להתנהגות הראשונה הזו, אינך רוצה להתייחס ישירות לשאלה הראשונה, "איזה נשיא יישם את 'העסקה החדשה' במהלך השפל הגדול?" במקום זאת, אתה רוצה להתייחס, באופן מופשט, לשקע הראשון ב- `QuestionList` (ללא קשר לשאלה הספציפית שם). כך, הבלוקים עדיין יפעלו גם אם תשנה את השאלה בשקע הראשון הזה.

אתה בוחר פריטים מסוימים ברשימה עם בלוק פריטי הבחירה. הבלוק מבקש ממך לציין את הרשימה ואינדקס (מיקום ברשימה). אם רשימה כוללת שלושה פריטים, אתה יכול להזין 1, 2 או 3 בתור האינדקס.

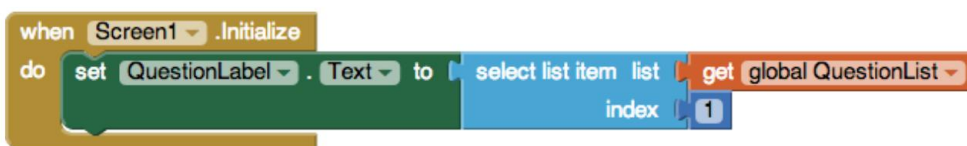
עבור התנהגות ראשונה זו, כאשר האפליקציה מופעלת, אתה רוצה לבחור את הפריט הראשון ב- `QuestionList` ולמקם אותו בתווית השאלות. נזכר מאנדרואיד, איפה המכונת שלי? אפליקציה, אם אתה רוצה שמשוהו יקרה כשהאפליקציה שלך מופעלת, אתה מתכנת את ההתנהגות הזו במטפל האירוע `Screen1.Initialize`. אתה יכול להשתמש בלוקים המפורטים בטבלה 4-8.

טבלה. 4-8 חסימות לטעינת השאלה הראשונית כשהאפליקציה מתחילה

מטרה	מגרה	סוג בלוק
מטרה 1: באירועים מופעל כשהאפליקציה מתחילה.		מסך. 1. אתחול
ל-QuestionLabel Text set QuestionLabel.Text שים את השאלה הראשונה. QuestionLabel-ב		
בחר את השאלה הראשונה מרשימת השאלות.	רשימות	
הרשמה לבחור שאלה ולבליט	משתנים	
במסך הראשון הראשונה באמצעות אינדקס של 1.	מתמטיקה	

איך הבלוקים עובדים

אירוע Screen1.Initialize מופעל כאשר האפליקציה מתחילה. כפי שנראה ב איור 5-8 הפריט הראשון של המשתנה QuestionList נבחר וממוקם לתוך QuestionLabel.Text, כאשר האפליקציה מופעלת, המשתמש יראה את השאלה הראשונה.



איור 6-8 בחירת השאלה הראשונה



בדוק את האפליקציה שלך לחץ על התחבר והתחבר למכשיר שלך או ל- אמולטור לבדיקה חיה. כאשר האפליקציה שלך נטענת, האם אתה רואה את הפריט הראשון של QuestionList "איזה נשיא יישם את 'ניו דיל' בתקופת השפל הגדול?"

איטריציה דרך השאלות

כעת, תכנת את ההתנהגות של ה- NextButton. כבר הגנת על currentQuestionIndex כזיכרון את השאלה הנוכחית. כאשר המשתמש לוחץ על NextButton, האפליקציה צריכה להגדיל את ה- currentQuestionIndex (כלומר, לשנות אותו מ-1 ל-2 או מ-2 ל-3, וכן הלאה). לאחר מכן תשתמש בערך המתקבל של currentQuestionIndex כדי לבחור את השאלה החדשה להצגה. כאתגר, ראה אם אתה יכול לבנות את הבלוקים האלה בעצמך. כשתסיים, השווה את התוצאות שלך נגד איור 6-8



## איור 7-8. עוברים לשאלה הבאה

איך הבלוקים עובדים

שורת הבלוקים הראשונה מגדילה את המשתנה `currentQuestionIndex` ל-1, והוא משתנה ל-2. אם יש לו 2, הוא משתנה ל-3, וכן הלאה. לאחר שינוי המשתנה, `currentQuestionIndex` האפליקציה משתמשת בו כדי לבחור את השאלה החדשה להצגה. כאשר המשתמש לוחץ על `NextButton` בפעם הראשונה, בלוקי ההגדלה ישנו את `currentQuestionIndex` מ-1 ל-2, כך שהאפליקציה תבחר את הפריט השני מתוך `QuestionList`. "איזה נשיא העניק לסין הקומוניסטית הכרה רשמית ב-1979?" בפעם השנייה שלוחצים על `currentQuestionIndex`, `NextButton` יוגדר מ-2 ל-3, והאפליקציה תבחר בשאלה השלישית ברשימה, "איזה נשיא התפטר עקב שערוריית ווטרגייט?"



### NextButton.Click בתוך הבלוקים של

אלה ב- `Screen.Initialize` מטפל באירועים. בתוך ה

מסך. אתחול בלוקים, האפליקציה השתמשה בפריט רשימה עם מספר קונקרטי (1) לבחור את פריט הרשימה. בלוקים אלה, אתה בוחר את פריט הרשימה באמצעות משתנה בתור האינדקס. האפליקציה לא בוחרת את הפריט הראשון ברשימה, או את השני או השלישי; הוא בוחר בפריט `currentQuestionIndexth`, יבחר פריט אחר בכל פעם שהמשתמש לוחץ על `NextButton`. זהו שימוש נפוץ מאוד לאינדקס -הגדלת הערך שלו כדי למצוא ולהציג או לעבד פריטים ברשימה.



**בדוק את האפליקציה שלך בדוק את התנהגות ה- NextButton כדי לראות אם**  
האפליקציה פועלת כהלכה. לחץ על הלחצן הבא בטלפון.  
האם הטלפון מציג את השאלה השנייה, "איזה נשיא  
העניקה לסין הקומוניסטית הכרה רשמית ב-1979?" זה  
צריך, והשאלה השלישית אמורה להופיע כשאתה לוחץ  
שוב כפתור Next. אבל אם תלחץ שוב, אתה אמור לראות  
שגיאה: "ניסיון להשיג פריט 4 מתוך רשימה באורך 3." האפליקציה  
יש באג! אתה יודע מה הבעיה? נסה לעצב את זה  
לצאת לפני שממשיכים הלאה.

הבעיה עם הקוד עד כה היא שהוא פשוט עולה לקוד הבא

שאלה בכל פעם ללא כל חשש לסוף החידון. מתי

`currentQuestionIndex` הוא כבר 3 והמשתמש לוחץ על `NextButton`, ההאפליקציה משתנה  
`currentQuestionIndex` מ-3 עד 4. לאחר מכן הוא קורא לפריט בחר רשימה כדי לקבל את  
`currentQuestionIndex` פריט - במקרה זה, הפריט הרביעי. כי יש רק  
שלושה פריטים במשתנה `QuestionList`, מכשיר האנדרואיד לא יודע מה לעשות  
לעשות. כתוצאה מכך, הוא מציג שגיאה ומאלץ את האפליקציה להפסיק. איך אנחנו יכולים לתת לאפליקציה  
יודע שזה הגיע לסוף החידון?

האפליקציה צריכה לשאול שאלה כאשר לוחצים על `NextButton` ולהפעיל

בלוקים שונים בהתאם לתשובה. כי אתה יודע שהאפליקציה שלך מכילה שלושה

שאלות, דרך אחת לשאול את השאלה תהיה, "האם המשתנה

`currentQuestionIndex` גדול מ-3?" אם התשובה היא כן, עליך להגדיר

`currentQuestionIndex` חזרה ל-1 ולקחת את המשתמש חזרה לשאלה הראשונה. ה

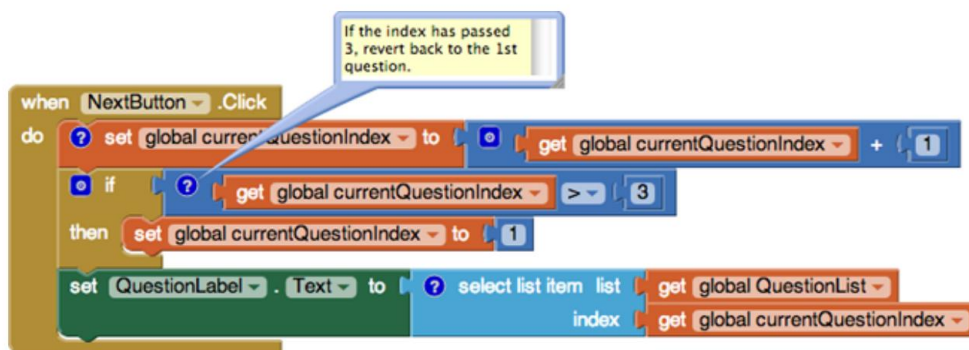
בלוקים שתזדקק לכך מופיעים בטבלה 5-8.

טבלה 5-8 בלוקים לבדיקת ערך האינדקס לסוף הרשימה

מטרה	מגרה	סוג בלוק
גלה אם המשתמש נמצא בשאלה האחרונה.	לשלוט	אם
בדוק אם <code>currentQuestionIndex</code> גדול מ-3.	מתמטיקה	<code>&gt;=</code>
שים את זה בצד שמאל של <code>=</code> .	גרור מ אתחול בלוק	קבל <code>currentQuestionIndex</code> העולמי
שים את זה בצד ימין של <code>=</code> כי 3 הוא המספר של פריטים ברשימה.	מתמטיקה	מספר 3
הגדר ל-1 כדי לחזור לשאלה הראשונה.	מגדרנות אתחול בלוק	<code>currentQuestionIndex</code> העולמי ל
הגדר את האינדקס ל-1.	מתמטיקה	מספר 1

הבלוקים צריכים להופיע כמתואר באיור 7-8.

## פרק 138: חידון נשיאים



איור 8-8. בודקים אם האינדקס עבר את השאלה האחרונה

איך הבלוקים עובדים

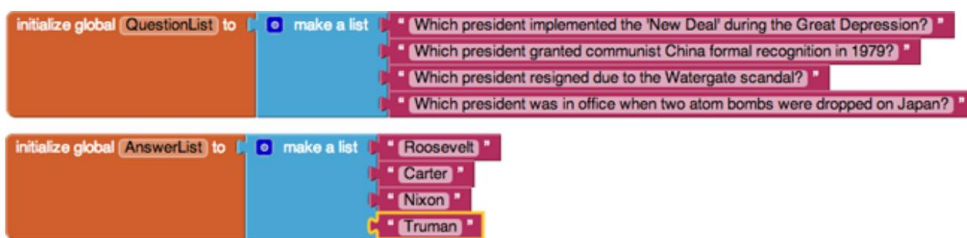
כאשר המשתמש לוחץ על NextButton, האפליקציה מגדילה את האינדקס כפי שעשתה קודם לכן. אבל אז, כפי שמוצג באיור 8-8, הוא בודק אם currentQuestionIndex גדול מ-3, שהוא מספר השאלות. אם הוא גדול מ-3, currentQuestionIndex מוחזר ל-1, והשאלה הראשונה מוצגת. אם הוא 3 או פחות, החסימות בתוך בלוק if לא מבוצעות, והשאלה הנוכחית מוצגת כרגיל.



בודק את האפליקציה שלך הרם את הטלפון ולחץ על הלחצן הבא. השאלה השנייה, "איזה נשיא העניק לסין הקומוניסטית הכרה רשמית ב-1979?" אמור להופיע QuestionLabel-בבטלפון, כמו קודם. כשתלחץ שוב, השאלה השלישית אמורה להופיע בטלפון. עכשיו, לגבי ההתנהגות שאתה באמת בודק: אם תלחץ שוב, אתה אמור לראות את השאלה הראשונה ("איזה נשיא יישם את 'העסקה החדשה' במהלך השפל הגדול?") מופיעה בטלפון.

הופך את החידון לקל לשינוי

אם הבלוקים שלך עבור NextButton עובדים, טפחי לעצמך על השכם; אתה בדרך להיות מתכנת! אבל מה אם הוספת שאלה (ותשובה) חדשה לחידון? האם הבלוקים שלך עדיין יעבדו? כדי לחקור זאת, תחילה הוסף שאלה רביעית לרשימת השאלות ותשובה רביעית ל-AnswerList, כפי שמוצג באיור 8-8.



איור 8-9. הוספת פריט לשתי הרשימות



בדוק את האפליקציה שלך לחץ על הלחצן הבא מספר פעמים. תשים לב שהשאלה הרביעית לעולם לא מופיעה, לא משנה כמה פעמים תלחץ על הבא. אתה יודע מה הבעיה? לפני שתמשיך לקרוא, בדוק אם אתה יכול לתקן את הבלוקים כך שהשאלה הרביעית תופיע.

הבעיה כאן היא שהבדיקה כדי לקבוע אם המשתמש נמצא בשאלה האחרונה היא ספציפית מדי; הוא שואל אם המשתנה `currentQuestionIndex` גדול מ-3. אתה יכול פשוט לשנות את המספר 3 ל-4, והאפליקציה תפעל שוב כהלכה. עם זאת, הבעיה עם הפתרון הזה היא שבכל פעם שאתה משנה את השאלות ורשימות התשובות, עליך לזכור גם לבצע את השינוי הזה במבחן אם.

תלות כזו בתוכנת מחשב מובילה לרוב לבאגים, במיוחד כאפליקציה גדל במורכבות. אסטרטגיה הרבה יותר טובה היא לעצב את הבלוקים כך שהם יעבדו לא משנה כמה שאלות יש. כלליות כזו מקלה על אם אתה, כמתכנת, רוצה להתאים אישית את החידון שלך לנושא אחר. זה גם חיוני אם הרשימה שאיתה אתה עובד משתנה באופן דינמי - לדוגמה, חשבו על אפליקציית חידון המאפשרת למשתמש להוסיף שאלות חדשות (אתה תבנה זאת בפרק 10). כדי שתוכנית תהיה כללית יותר, היא לא יכולה להתייחס למספרים קונקרטיים כמו 3, כי זה עובד רק עבור חידונים של שלוש שאלות.

לכן, במקום לשאול אם הערך של `currentQuestionIndex` גדול מהמספר הספציפי 3, שאל אם הוא גדול ממספר הפריטים ב- `QuestionList`. האפליקציה שואלת שאלה כללית יותר זו, היא תעבוד גם כאשר תוסיף או תסיר פריטים מרשימת השאלות. הבה נשנה את המטפל באירועים `NextButton.Click` כדי להחליף את הבדיקה הקודמת שהתייחסה ישירות ל-3. תזדקק לבלוקים המפורטים בטבלה 8-6.

טבלה 8-6. חסימות כדי לבדוק את אורך הרשימה

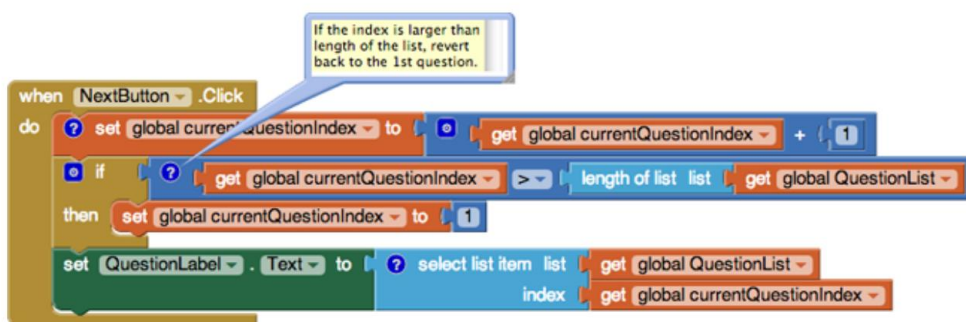
מטרה	מגרה	סוג בלוק
שאל כמה פריטים יש ברשימת השאלות.	רשימות	אורך הרשימה

## 140 פרק 8: חידון נשיאים

מטרה	מגרה	סוג בלוק
הכניסו את זה לשקע "רשימה" של אורך הרשימה.	גרור פנימה מבליק האתחול	קבל רשימת שאלות גלובלית

איך הבלוקים עובדים

מבחן ה- if משווה כעת את ה- `currentQuestionIndex` לאורך השאלות, כפי שמוצג באיור 11-178. אם `currentQuestionIndex` הוא 5, והאורך של `QuestionList` הוא 4, ה- `currentQuestionIndex` יוחזר ל-1. שימו לב, מכיוון שהבלוקים כבר לא מתייחסים ל-3 או למספר ספציפי כלשהו, ההתנהגות תעבוד לא משנה כמה פריטים נמצאים ברשימה.



איור 10-8 בדיוק אם האינדקס גדול מאורך הרשימה (במקום 3)



**בדוק את האפליקציה שלך כאשר אתה לוחץ על הלחצן Next, האם האפליקציה עוברת כעת בין ארבע השאלות, ועוברת לראשונה אחרי הרביעית?**

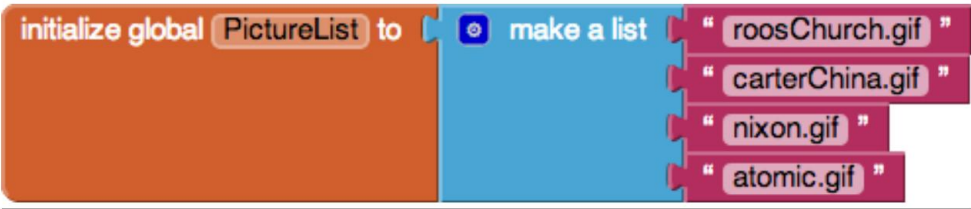
### החלפת תמונה עבור כל שאלה

כעת, לאחר שתכנתת את כל ההתנהגויות למעבר בין השאלות (והפכת את הקוד שלך לחכם וגמיש יותר על ידי הפיכתו יותר מופשט), המשימה הבאה שלך היא לגרום לתמונות לעבוד כמו שצריך. כרגע, האפליקציה מציגה את אותה תמונה לא משנה איזו שאלה נשאלת. אתה יכול לשנות זאת כך שתופיע תמונה הקשורה לכל שאלה כאשר המשתמש לוחץ על הבא.

קודם לכן, הוספת ארבע תמונות כמדיה עבור הפרויקט. כעת, תיצור רשימה שלישית, `PictureList`, עם שמות התמונה כפריטים שלה. תוכל גם לשנות את מטפל האירועים `NextButton.Click` כדי להחליף את התמונה בכל פעם, בדיוק כפי שאתה מחליף את טקסט השאלה. (אם אתה כבר חושב להשתמש ב- `currentQuestionIndex` כאתר בדרך הנכונה!) ראשית, צור `PictureList` ואתחול אותה עם שמות התמונות. ודא שהשמות זהים לחלוטין לשמות

הוספת התנהגויות לרכיבים 141

flenamesשטענט למקטע המדיה של הפרויקט. איור 8-10מראה כיצד בלוקים עבור PictureListצריכים להיראות.



איור 8-11. PictureList-ה עם שמות תמונות כפריטים

לאחר מכן, שנה את המטפל באירועים `NextButton.Click` כך שישנה את התמונה שמופיע לכל שאלה. המאפיין `Image.Picture` משמש לשינוי תמונה המוצגת. כדי לשנות את `NextButton.Click`, תצטרך את הבלוקים הרשומים ב-7.8 טבלה

טבלה 7.8-בלוקים להוספת התמונה המלווה את השאלה

מטרה	מגרה	סוג בלוק
הגדר את זה כדי לשנות את התמונה.	תמונה 1	הגדר תמונה 1.תמונה ל
בחר את התמונה המתאימה לתמונה הנוכחית שאלה.	רשימות	בחר פריט רשימה
בחר שם שם מרשימה זו.	גרור החוצה בלוק אתחול	קבל PictureListגלובלי
בחר בפריט. currentQuestionIndex-ה	גרור החוצה בלוק אתחול	קבל currentQuestionIndex העולמי


איך הבלוקים עובדים

ה- `currentQuestionIndex` משמש כאינדקס הן עבור השאלות והן עבור `PictureList`. כל עוד הגדרת את הרשימות שלך כראוי כך שהשאלה הראשונה מתאים לתמונה הראשונה, השנייה לשניה, וכן הלאה, האינדקס היחיד יכול לשרת את שתי הרשימות, כפי שמוצג באיור 8-11. לדוגמה, התמונה הראשונה, `roosChurch.gif`, היא תמונה של הנשיא פרנקלין דלאנו רוזוולט (יושב עם בריטים ראש הממשלה וינסטון צ'רצ'יל), ו"רוזוולט" היא התשובה לשאלה הראשונה.

```
when NextButton.Click
do
  set global currentQuestionIndex to (get global currentQuestionIndex + 1)
  if (get global currentQuestionIndex > length of list list get global QuestionList)
  then set global currentQuestionIndex to 1
  set QuestionLabel.Text to (select list item list get global QuestionList
                                index get global currentQuestionIndex)
  set Image1.Picture to (select list item list get global PictureList
                           index get global currentQuestionIndex)
```

The same index is used to select both a pic and a question.

איור 12-8 בחירת התמונה הנוכחית Qtnoitseuhtxעד בכל פעם



**בדוק את האפליקציה שלך לחץ על הבא כמה פעמים. עושה תמונה אחרת להופיע בכל פעם שאתה לוחץ על הלחצן הבא?**

בדיקת תשובות המשתמש

עד כה, יצרת אפליקציה שפשוט עוברת בין שאלות ותשובות (בשילוב עם תמונה של התשובה). זו דוגמה מצוינת לאפליקציות שמשתמשות ברשימות, אבל כדי להיות אפליקציית חידון אמיתית, היא צריכה לתת למשתמשים משוב אם התשובות שלהם כן נכון. בואו נוסיף את הבלוקים כדי לעשות זאת. הממשק שלנו מוגדר כך שהמשתמש הקלד תשובה ב- AnswerText ולאחר מכן לחץ על AnswerButton. האפליקציה חייבת השווה את הערך של המשתמש עם התשובה לשאלה הנוכחית, באמצעות אם אחרת לחסום כדי לבדוק. לאחר מכן יש לשנות את ה- RightWrongLabel כדי לדווח אם התשובה נכונה. יש לא מעט בלוקים הדרושים כדי לתכנת את ההתנהגות הזו, כולם המפורטים בטבלה 8-8.

טבלה 8-8 בלוקים לציון האם תשובה נכונה

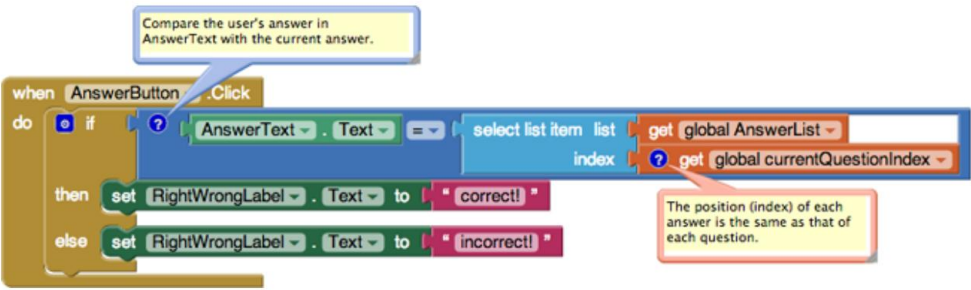
מטרה	מגרה	סוג בלוק
מופעל כאשר המשתמש לוחץ על כפתור תשובה.	כפתור תשובה	AnswerButton.Click
אם התשובה נכונה, עשה דבר אחד; אחרת, לעשות אחרת.	לשלוט	אחרת
שאל אם התשובה נכונה.	מתמטיקה	=
מכיל את תשובת המשתמש.	תשובה טקסט	AnswerText.Text
בחר את התשובה הנוכחית מרשימת התשובות.	רשימות	בחר פריט רשימה
הרשימה לבחירה.	גרור החוצה בלוק אתחול	קבל רשימת תשובות גלובלית

הוספת התנהגויות לרכיבים 143

מטרה	מגרה	סוג בלוק
מספר השאלה (והתשובה) הנוכחית.	גרור החוצה בלוק אתחול	קבל currentQuestionIndex העולמי
דווח על התשובה כאן.	הגדר RightWrongLabel ל-RightWrongLabel.Text	
הצג זאת אם התשובה נכונה.	טקסט ("נכון!")	
דווח על התשובה כאן.	הגדר RightWrongLabel ל-RightWrongLabel.Text	
הצג זאת אם התשובה שגויה.	טקסט ("לא נכון!")	

איך הבלוקים עובדים

איור 12-8 מדגים כיצד מבחן אם אחרת שואל האם התשובה למשתמש מספק (AnswerText.Text) שווה לפרט currentQuestionIndexth-1. רשימת תשובות. אם currentQuestionIndex הוא 1, האפליקציה תשווה את תשובת המשתמש עם הפרט הראשון ברשימת התשובות, "רוזולט". אם currentQuestionIndex הוא 2, האפליקציה תהיה השווה את תשובת המשתמש לתשובה השנייה ברשימה, "קרטר" וכן הלאה. אם תוצאת הבדיקה חיובית, הסניף לאחר מכן מבוצע וה- RightWrongLabel מוגדר לתקן! אם הבדיקה היא שקר, הסניף else מבוצע וה- RightWrongLabel מוגדר ל"לא נכון!"



איור 13-8 בדוק את התשובה



בדוק את האפליקציה שלך נסה לענות על אחת השאלות. זה אמור דווח אם ענית על השאלה בדיוק כפי שצוין ברשימת התשובות. בדוק עם נכון ולא נכון תשובה. סביר להניח שתבחין בכך כדי שתשובה תסומן כ נכון, זה חייב להיות התאמה מדויקת וספציפית מקרה למה שהזנת ברשימת התשובות. הקפד לבדוק גם את הדברים האלה לעבוד על שאלות עוקבות.

האפליקציה אמורה לעבוד, אך ייתכן שתבחין שכאשר תלחץ על הכפתור הבא, הנכון! או "לא נכון!" הטקסט והתשובה הקודמת עדיין שם, כפי שמוצג ב

## 144 פרק: 8 חידון נשיאים

---

איור 13-18 למרות שאתה מסתכל על השאלה הבאה. זה די תמים, אבל משתמשי האפליקציה שלך בהחלט ישימו לב לבעיות כאלה בממשק המשתמש. כדי לבטל את התוויות RightWrongLabel ואת ה- AnswerText תכניס את הבלוקים הרשומים בטבלה 9-8 בתוך המטפל באירוע . NextButton.Click



---

איור 14-8 חידון בפעולה כאשר התשובה הקודמת מופיעה כאשר לא אמורה

טבלה. 8-9 חסימות כדי לנקות את התווית RightWrongLabel

מגרה	סקטור
הגדר RightWrongLabel.Text זהו התווית לריק.	הגדר 7-RightWrongLabel
	כאשר המשתמש לוחץ על NextButton, נקה את הקודם המשוב של התשובה.
הגדר את AnswerText.L	תשובת המשתמש תישאר קודמת.
טקסט ("")	כאשר המשתמש לוחץ על הלחצן Next, נקה את התשובה קודמת.

איך הבלוקים עובדים

כפי שמוצג באיור, 8-14 על ידי לחיצה על כפתור Next, המשתמש עובר לבא שאלה, כך ששתי השורות העליונות של המטפל באירועים מנקות את RightWrongLabel וטקסט התשובה.

```
when NextButton.Click
do
  set AnswerText.Text to ""
  set RightWrongLabel.Text to ""
  set global currentQuestionIndex to (get global currentQuestionIndex + 1)
  if (get global currentQuestionIndex > length of list global QuestionList)
  then
    set global currentQuestionIndex to 1
  set QuestionLabel.Text to (select list item list global QuestionList
                             index get global currentQuestionIndex)
  set Image1.Picture to (select list item list global PictureList
                         index get global currentQuestionIndex)
```

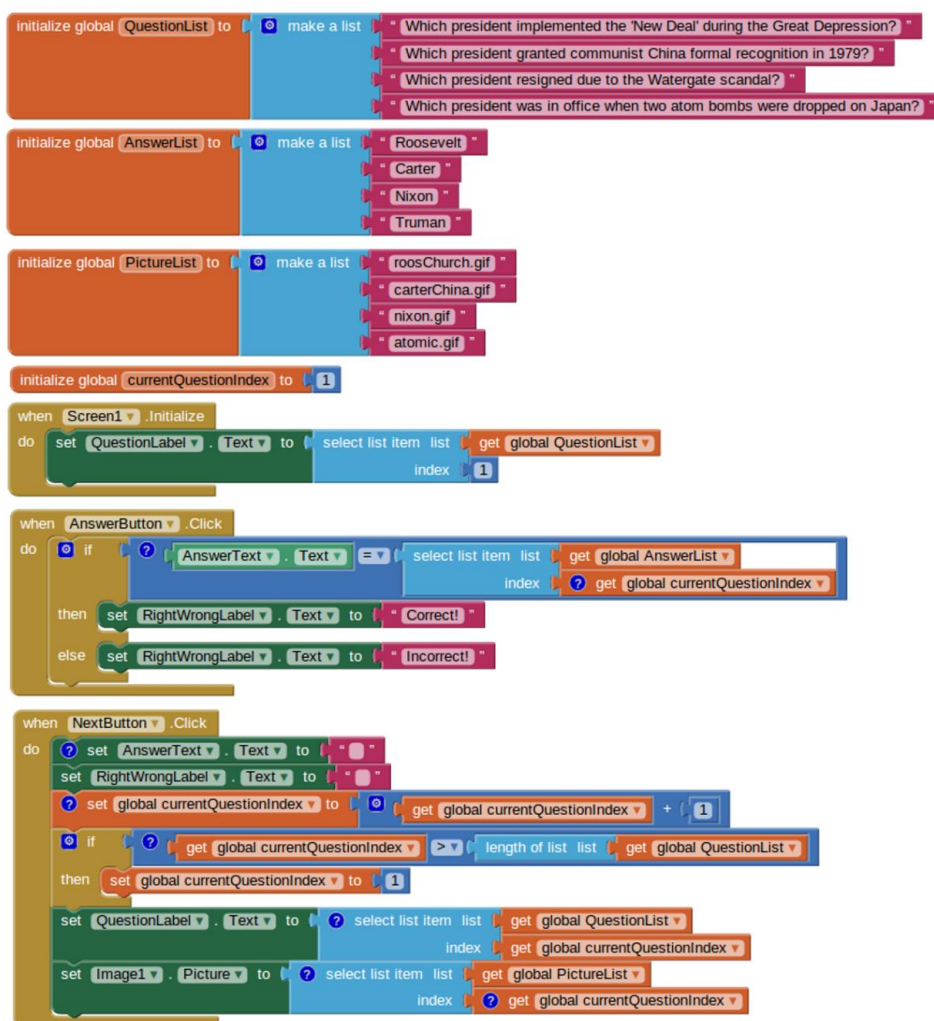
איור 8-15. עם שמות תמונות כפריטים



בדוק את האפליקציה שלך ענה על שאלה ולחץ על שלח ולאחר מכן לחץ כפתור הבא. האם תשובתך הקודמת והמשוב שלה להעלים?

# האפליקציה השלמה: חידון הנשיאים

איור 8-15 מציג את תצורת הבלוק הסופי עבור חידון הנשיאים.



איור 16-8. הבלוקים לחידון הנשיאים

## וריאציות

כאשר אתה מקבל את החידון הזה לעבוד, אולי תרצה לחקור כמה וריאציות, כגון הבאות:

- במקום להציג רק תמונות עבור כל שאלה, נסה להפעיל קליפ קול או סרטון קצר. עם סאונד, אתה יכול להפוך את החידון שלך לאפליקציית Name That Tune.
- החידון מאוד נוקשה מבחינת מה שהוא מקבל כתשובה תקפה. ישנן מספר דרכים לשפר זאת, תוך ניצול היתרון של בלוקים לעיבוד טקסט במגירת הטקסט. אחת הדרכים היא להשתמש בלוק `uppercase` במגירת הטקסט כדי

המר את תשובת המשתמש ואת התשובה האמיתית לכל האותיות הגדולות לפני ההשוואה. אחר הוא להשתמש בבלוק `text.contains` כדי לראות אם התשובה של המשתמש היא הכלול בתשובה בפועל. אפשרות נוספת היא לספק מספר תשובות לכל שאלה ולבדוק על ידי איטרציה (לפנים) דרך כדי לראות אם יש התאמה.

• דרך נוספת לשפר את בדיקת התשובות היא להפוך את החידון כך שיהיה רב ברירה. תזדקק לרשימה נוספת כדי להחזיק את אפשרויות התשובות לכל שאלה. התשובות האפשריות יהיו רשימה של רשימות, כאשר כל תת-רשימה תכיל את אפשרויות התשובות לשאלה מסוימת. השתמש ברכיב `ListPicker` כדי לתת למשתמש את היכולת לבחור תשובה. תוכל לקרוא עוד על רשימות בפרק 19.

## סיכום

הנה כמה מהרעיונות שכסינו במדריך זה:

• לרוב האפליקציות המתוככמות למדי יש נתונים (לעיתים מאוחסנים ברשימות) והתנהגות - שלהם מטפלי אירועים.

• השתמש בבלוק `ifelse` כדי לבדוק תנאים. למידע נוסף על

תנאים, ראה פרק 18.

• החסימות במטפלי אירועים צריכים להתייחס רק באופן מופשט לפריטי רשימה ולגודל רשימה, כך שהאפליקציה תעבוד גם אם הנתונים ברשימה ישתנו.

• משתני אינדקס עוקבים אחר המיקום הנוכחי של פריט בתוך רשימה. כשאתה מגדיל אותם, השתמש ב- `block` כדי לטפל בהתנהגות האפליקציה כאשר המשתמש מגיע לסוף הרשימה.

