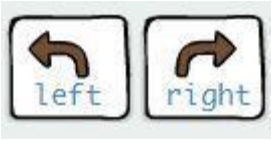
מקרא

תיאור	מילות מפתח / לחצן

כדי לגרום לקוף לצעוד למרחק מסוים, ז"א לבצע "step", אנחנו צריכים לכתוב "step X" ולהשתמש במספר הצעדים שאנחנו רוצים שהוא יעשה, לדוגמה, "step 10". לחיצה על לחצן **step** תכתוב את המילה "step" בקוד שלכם.

"turn" צריך להיות מלווה בכיוון (left / right) או במעלות (45, 90, 180).
דוגמאות: "turn 90", "turn right"
לחיצה על לחצן **turn** תכתוב את המילה "turn" בקוד שלכם.



"Left" ו-"Right" משמשים אחרי הפקודה "turn" כדי לגרום לקוף לפנות לכיוון הרצוי.

לחיצה על הלחצנים **left** או **right** תכתוב בהתאמה את המילים "left" או "right" בקוד שלכם.

"TurnTo" היא דרך נוספת לפנות. במקום להשתמש בכיוון או מעלות, אנחנו מבקשים מהקוף לפנות אל אובייקט מסוים, לדוגמה, " turnTo banana". לחיצה על הלחצן **turnTo** תכתוב את המילה "turnTo" בקוד שלכם.

לולאה פשוטה היא רצף של הוראות החוזר על עצמו למשך מספר מוגדר של פעמים.

דוגמה:

```
3.times ->
    step 5
    turn left
```


© 2015 קודמאנקי סטודיוז בע"מ

בדוגמה זו, קוף יחזור שלוש פעמים על "step 5, turn left". ההוראות שאנחנו כותבים בלולאה צריכות להיות כתובות מתחתיה עם אינדנטציה (....). תוכלו לעשות את זה על ידי לחיצה על מקש **Tab** במקלדת.

לחיצה על לחצן **times** תכתוב תחילת לולאה פשוטה בקוד שלכם: "3.times - >".

השמות למשתנים. משתנה הוא כמו יחידת אחסון. אנחנו מאחסנים בו נתונים, ואנחנו משתמשים בו רק כאשר אנחנו צריכים אותו. השמה למשתנה מורכבת ממזהה וערך. הפרדה זו של שם וערך מאפשרת להשתמש בשם באופן עצמאי מהמידע שהוא מייצג. אנחנו יכולים להשתמש ב-X בעת כתיבת התכנית, בלי לדעת מה הערך שלו יהיה כאשר ההוראות תבוצענה.

```
x = 10
step x
```

"say" יגרום לכך שבסמוך לקוף תופיע בועת דיבור עם הטקסט שהקלדנו, למשל:

```
say "Boo!"
```

יגרום לקוף לומר "Boo!"

אנחנו משתמשים בגרשיים ("") סביב הביטוי שאנחנו רוצים שהקוף יאמר, כדי שהמחשב יבין שהטקסט שהקלדנו הוא לא משתנה. נסו להשתמש ב"say" כאשר יש עכבר בסביבה. לחיצה על לחצן **say** תכתוב את המילה "say" בקוד שלכם.

"DistanceTo" משמש עם פקודה אחרת כמו "step" או "say" ואובייקט. להשתמש ב-"distanceTo" זה כמו לשאול שאלה, למשל, "מה המרחק לבננה?" התשובה היא מספר, המחושב על ידי המחשב, שמייצג את המרחק.

דוגמה:

```
step distanceTo banana
```

המחשב ימדוד את המרחק בין הקוף והאובייקט (הבננה). אז הוא ישתמש במספר שהוא הגיע אליו כדי לבצע מה שהורינו לו, תוך שימוש בערך הנמדד כארגומנט ל"step".לחיצה על לחצן **distanceTo** תכתוב את המילה "distanceTo" בקוד שלכם.

© 2015 קודמאנקי סטודיוז בע"מ

זוהי לולאת "for". נשתמש בלולאת "for" כאשר יהיה לנו אוסף של אובייקטים ונרצה לחזור על פעולה המתייחסת לכל אחד מהם באופן ספציפי. לולאת ה-"for" תמשיך לפעול עד שכל הפעולות יעשו על כל האובייקטים שבאוסף שלנו (מערך).כאשר המחשב מבצע את הלולאה הזו, הוא מחליף את שם המשתנה עם הפריט הראשון באוסף. אחרי שהוא מסיים עם הפריט הראשון, הוא עובר לשני, וכך הלאה.

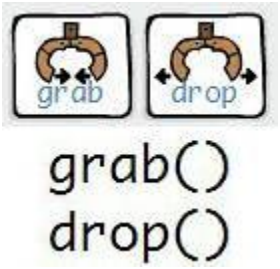
לדוגמה:

כמו כן, אנחנו יכולים להשתמש בלולאת "for" בתוך לולאת "for"; הדוגמה משמאל נלקחה משלב מספר 70.

```
for b in bananas
    for c in crocodiles
        c.turnTo b
    turnTo b
    step distanceTo b
```

לחיצה על לחצן for תכתוב את הטקסט הבא בקוד שלכם.
שימו לב לשורת ההערה:

```
for name in array
    # Your code here
```



"grab() " ו-"drop()" הן פונקציות ללא ארגומנט המשמשות אותנו בחלק השלישי כדי לעזור לעכבר לאסוף גפרורים. פונקציות ללא ארגומנט מבצעות פעולה כלשהי, אך לא מחייבות אותנו ***להעביר*** קלט כלשהו. לחיצה על הלחצנים grab או **drop** תכתוב את המילים "grab()" או "drop()" בקוד שלכם, בהתאמה.

```
# By the way, this is a comment
# It starts with a '#', like th
```

זוהי שורת הערה. היא מסומנת בקוד באמצעות סמל הסולמית (#) בתחילתה. המחשב לא מתייחס אל שורה זו כהוראה. במקום, נעשה בה שימוש על ידי המתכנתים שכותבים וקוראים את הקוד כדי להבין אחד את השני.

© 2015 קודמאנקי סטודיוז בע"מ

פונקציה היא סט של הוראות שמבצע משימה מסוימת. המחשב יבצע את הפונקציה רק כאשר אנחנו נקרא לה, כלומר, נשתמש בה בקוד שלנו.

זוהי דוגמה לאיך אנחנו מגדירים פונקציה:

```
goto = (t) ->
    turnTo t
    step distanceTo t
```

ברגע שהגדרנו פונקציה, יופיע לחצן ריק חדש עם השם של הפונקציה.

כאשר אנחנו רוצים לקרוא לפונקציה בקוד, אנחנו משדכים את השם של הפונקציה עם שם האובייקט שעליו אנחנו רוצים שיבוצעו הפעולות שבתוך הפונקציה:

```
goto match
grab()
goto pile
drop()
```


לחיצה על לחצן **function** תכתוב את הטקסט הבא בקוד שלכם. שימו לב לשורת ההערה:

function_name = (argument) >
….#your code here

לולאת "until" מכילה בלוק של קוד שימשיך לפעול ***עד*** שתנאי ספציפי יתמלא. תנאי זה נקרא תנאי לולאה. המחשב בודק את המצב בהתחלה. אם התשובה כוזבת, הלולאה תמשיך לפעול. היא תעצור רק כאשר התשובה תהיה אמת.
לחיצה על לחצן **until** תכתוב את הטקסט הבא בקוד שלכם.
שימו לב לשורת ההערה:

דוגמה

```
until near match
    step 1
```

until condition
….#your code here

© 2015 קודמאנקי סטודיוז בע"מ

אנחנו יכולים להשתמש בפונקציה "near" עם תנאי לולאה. הערך שמוחזר על ידי הפונקציה "near" יקבע מתי לולאת ה-"until" תפסיק להתבצע. לחיצה על לחצן **near** תכתוב את המילה "near" בקוד שלכם.

בחלק השלישי, אנחנו פוגשים את החתול. החתול יתקוף את העכבר אם הוא יראה אותו, ולכן אנחנו חייבים לחכות שהחתול ירדם. "sleeping()" היא עוד פונקציה שבה אנחנו יכולים להשתמש עם תנאי לולאה. "wait()" היא פונקציה ללא ארגומנט. כשנשתמש ב-"sleeping()", אנחנו צריכים לכתוב:

```
until cat.sleeping()
    wait()
```

לחיצה על לחצן **run** תגרום לקוד מימין לרוץ. תוכלו לראות את התוצאה על ידי התבוננות בסצנה משמאל.

לחצן האיפוס (reset) ימחק את כל מה שכתבת בקוד בצד הימין ויאפס את הקוד למצב שבו הוא היה בתחילת השלב.

הסרגל הוא כלי שעוזר לכם למדוד את המרחק בין אובייקטים שונים במשחק, למשל, את המרחק בין הקוף והבננה. הסרגל יכול גם לעזור לכם למדוד את הזוויות בהן על הקוף או הצב לפנות כדי לעמוד מול אובייקט אחר שעל המסך, כמו בננה. כדי להשתמש בסרגל, יש ללחוץ עליו פעם אחת, ולאחר מכן להשתמש בעכבר כדי להזיז את הסרגל עד לנקודה שתרצהו להתחיל למדוד ממנה. יש ללחוץ על העכבר שוב, ולאחר מכן לגרור אותו לנקודה הסופית. מספר יופיע בנקודה הסופית ויציין את המרחק. השתמשו במספר זה עם הפקודה "step". מספר נוסף יופיע ליד נקודת ההתחלה, והוא יציין את הזווית בין האובייקט הראשון לשני. השתמשו בו עם הפקודה "turn".

© 2015 קודמאנקי סטודיוז בע"מ

לאחר כל שלב, תקבלו דירוג כוכבים לפתרון שלכם.הכוכבים מחולקים כך:

- הכוכב הראשון ניתן אם יש לכם את כל הבננות
- הכוכב השני ניתן אם השתמשת במה שלמדת
- הכוכב שלישי ניתן אם הקוד שלכם הוא קצר ולעניין

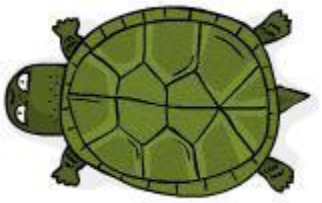
© 2015 קודמאנקי סטודיוז בע"מ

סקירת הדמויות

דמות	תיאור

גורדו, שנקרא על שמו של הקוף הראשון בחלל, הוא המדריך שיעזור לכם וייתן לכם הוראות לאורך הדרך. ההערות שלו מצחיקות ומועילות. תמיד אפשר ללחוץ עליו ולקרוא את ההוראות.

הקוף הוא הדמות הראשית. תצטרכו לעזור לו לאסוף בננות על ידי כתיבת שורות קוד. רק שתדעו, קופים לא אוהבים להירטב והם מאוד ידידותיים.



בשלב מספר 13, תוכלו לפגוש את הצב הנאמן שלנו. הצב יעזור לך להשיג את הבננות הערמומיות האלו. כדי להורות לצב לבצע "turn" או"step", אנחנו צריכים קודם כל ללחוץ עליו. זה יכתוב את השם שלו בקוד, ולאחר מכן יפריד אותו מהפעולה שאנחנו רוצים שהוא יבצע באמצעות שימוש בנקודה

(.).

דוגמה:

```
turtle.step 10
```

בשלב 50 נפגוש את הבונים. הבונים אוהבים מאוד עץ, והם הסכימו לעזור לכם לחצות את המים ולקבל יותר בננות על ידי דריכה על העץ שלהם. הם יכולים לבצע רק "step" קדימה ואחורה. כדי להשתמש בהם אנחנו צריכים להשתמש בנקודה (.) בין שמם לבין הפעולה שאנחנו רוצים שהם יבצעו. לדוגמה: beavers[0].step 10.

את התנינים נפגוש בשלב מספר 56. הם משמשים ליצירת גשר על המים, כדי לעזור לקוף להגיע לבננות. הם יכולים לבצע רק "turn" או "turnTo". אנחנו בדרך כלל נשתמש בתנינים עם לולאות "for".

```
for c in crocodiles
    c.turn right
```


© 2015 קודמאנקי סטודיוז בע"מ

אנחנו פוגשים את העכברים כמה פעמים במשחק, אבל רק בשלב מספר 71 אנחנו באמת יכולים לשלוט בהם. עכברים אוהבים לאסוף פריטים. בחלק השלישי, אנחנו עוזרים להם לאסוף גפרורים. כדי לעשות את זה, נשתמש בפונקציות הפשוטות הבאות: "()grab" ו-"()drop".

את הנמלים נפגוש בשלב מספר 89 והן יעזרו לנו להדגים את לולאת "until". הנמלים גוררות את הגפרורים היקרים שלנו, ואנחנו צריכים לבצע "step" עד שנגיע אליהם וניקח את הגפרורים בחזרה.

עם החתול ניפגש בשלב מספר 96 והוא יעזור לנו להדגים את לולאת "until" עם הפונקציה "()wait".החתול יתקוף את העכבר אם הוא יראה אותו, ולכן אנחנו חייבים לחכות שהוא ירדם לפני שנצא לאסוף גפרורים.

© 2015 קודמאנקי סטודיוז בע"מ

תמיכה

לא מצאתם את מה שחיפשתם? ניתן ליצור איתנו קשר בכתובת הדוא"ל info@cm-studios.com

או לחילופין להיכנס לקבוצת קודמאנקי בפייסבוק בכתובת:

https://www.facebook.com/groups/1535155600107645/

© 2015 קודמאנקי סטודיוז בע"מ