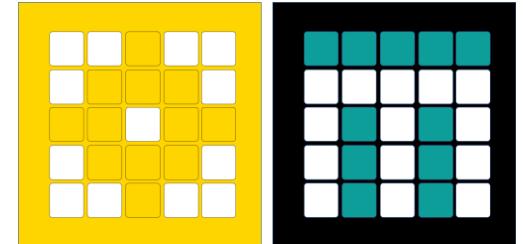


PRIME LESSONS

By the Makers of EV3Lessons



LOOPS

BY SANJAY AND ARVIND SESHAN

OBIECTIVUL LECȚIEI

- Învățăm cum să repetăm o acțiune utilizând LOOPS

COD DE REPETARE

- Să spunem că dorești ca robotul tău să repete o acțiune din nou și din nou.
 - Vei copia codul din nou și din nou?
 - Cum ar fi dacă ai dori ca acțiunea să se repete la infinit?
- Poți folosi **LOOPS** pentru a repeta o acțiune pentru un anumit număr de ori sau până când se îndeplinește o anumită condiție
- Python are 2 tipuri de **LOOPS**: **Loops** și **While loops**

WHILE LOOPS

- Să spunem că dorim ca un task să fie executat până când o condiție se îndeplinește (Adevărat)
 - E.g. Cât sunt în librărie, stai liniștit
- In Python, we use `while (statement):` to run code while the statement is True
- In the example on the right, `x==8` is always True, so “Yay!” will print forever
 - If you insert `x=10` inside the loop, “Yay!” will only print once, for example
- While loops are useful for repeating a task until a certain sensor reading:

```
x = 8
while (x == 8):
    print("Yay!")
```

Output:

Yay!
Yay!
Yay!
... [repeats forever]

```
# Mergi înainte până când senzorul de distanță returnează o valoare <=10cm
while (getDistance() > 10):
    moveForward()
# Ia în considerare că getDistance() primește valoarea distanței în
# centimetri și moveForward() mișcă robotul înainte
```

Notă:
Adu-ți aminte să introduci codul pe care vrei să-l rulezi într-un LOOP

WHILE LOOPS NEDEFINITE

- Poți utiliza while loops pentru a repeta la infinit

`while True:`

Code

- Prin setarea unei condiții care să fie **Adevărată** tot timpul, loop-ul va repeta codul tot timpul

PROVOCARE

- Creează o variabilă x și atașează-i o valoare
- Creează un While loop care afișează toate pătratele (e.g., 4, 9, 16, ...) care sunt mai mici ca x pe HUB

PROVOCAREA SOLUȚIE

```
from spike import PrimeHub, LightMatrix
hub = PrimeHub()

# aceasta crează variabila x și setează valoarea lui x la 51
x = 51

# aceasta creează variabila y care va fi folosită ca a
# loop counter. Începem cu y = 1
y = 1

# aceste loops repetă acțiunea până când pătratul lui y este >= x
while ((y**2)< x):
    hub.light_matrix.write(y**2)
    # trebuie să creștem y pentru a calcula pătratul valorii sale
    y += 1
```

FOR LOOPS

- Similar cu while loops, dar rulează pentru un calcul fix
 - E.g. sari de 10 ori
- Un exemplu de for loop de bază este ca în exemplul din dreapta
- În “for i în intervalul (start, sfârșit, rată de creștere):”
 - intervalul() crează un set de numere între un număr de start și un număr mai mic decât numărul de final (sau doar un număr de sfârșit când este prezent unul din parametri) care au ca regulă o rată de creștere. Startul și valorile de creștere sunt opționale.
 - Variabila i ia valoarea următoare din set de fiecare dată (poți să denumești această variabilă oricum dorești; convenția standard este i, j, k)
 - În exemplu, i va fi doar între 0 și 9, din moment ce îndeplinește condiția de verificare $< n$ (nu \leq)

```
for i in range(0,10):  
    print("Jump!")  
    print(i)
```

Output:

```
Jump!  
0  
Jump!  
1  
...  
Jump!  
9
```


ANALIZĂ: PENTRU LOOPS CU INTERVAL()

- Structură de bază:

```
for i in range(4):  
    print(i)
```

Output:

0
1
2
3

- Poți seta o poziție de start:

```
for i in range(2, 4):  
    print(i)
```

Output:

2
4

- la în considerare ca 4 nu este inclusă. Funcția interval () exclude maximul pe care îl setezi.

- În final, poți crește cu diferite valori peste 1

```
for i in range(2, 7, 2):  
    print(i)
```

Output:

2
4
6

↑
Increment

PENTRU LOOPS CU O LISTĂ DE NUMERE

- Funcția loops poate fi utilizată pentru a itera peste o paranteză separată de o listă de numere (enclosed by brackets [])

```
for i in [0, 2, 6]:  
    print(i)
```

Output:

0

2

6

Notă: Acest exemplu utilizează liste, care nu au fost încă acoperite.

EXAMPLE LOOP

```
# A for loop repeats an action a specific number of
times
# based on the provided range
def sumFromMToN(m, n):
    total = 0
    # note that range(x, y) includes x but excludes y
    for x in range(m, n+1):
        total += x
    return total
```

```
def printStarRectangle(n):
    # print an nxn rectangle of asterisks
    for row in range(n):
        for col in range(n):
            print("*", end=" ")
        print()
```

```
# use while loops when there is an
indeterminate number of iterations
def leftmostDigit(n):
    n = abs(n)
    while (n >= 10):
        n = n//10
    return n
```

PROVOCARE: NUMERELE PRIME

- Țelul tău este să verifici dacă orice număr întreg pozitiv n este număr prim
- Indicii:
 - Numerele prime sunt numerele divizibile cu 1 și cu el însuși
 - Trebuie să verifici divizibilitatea numerelor dintre 2 și $n-1$
 - Modulo (%) te va ajuta aici (numărul poate fi testat ca întreg dacă $n \% a == 0$)

SOLUȚIA PROVOCĂRII

```
n = 3 # numărul tău aici
prime = True # începeți prin a asuma ca e un număr prim
if (n <= 1): # 1 și mai mic de 1 nu sunt prime
    prime = False
for factor in range(2,n): # verificați toți factorii posibili[2, n)
    if (n % factor == 0): # n%factor == 0 când este divizor
        prime = False # setează că n nu este prim
if prime:
    print("is prime") # numărul este prim
else:
    print("not prime") # numărul nu este prim
```

CREDITS

- Această lecție de SPIKE Prime a fost realizată de Sanjay Seshan și Arvind Seshan.
- Mai multe lecții sunt disponibile pe www.primelessons.org
- Această lecție a fost tradusă în limba română de echipa de robotică FTC – ROSOPHIA #21455 RO20



This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-nc-sa/4.0/).