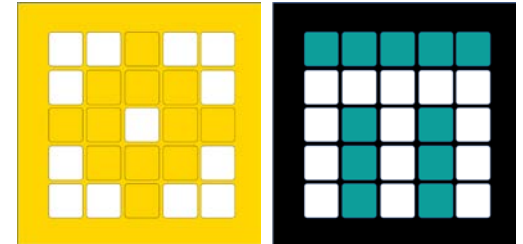


PRIME LESSONS

By the Makers of EV3Lessons



RECURSIVITATEA

BY SANJAY AND ARVIND SESHAN

This lesson uses SPIKE 3 software

OBIECTIVELE LECȚIEI

- Învățăm să creeăm o funcție recursivă

INTRODUCERE RECURSIVITATE

- Definiția recursivității (n):

- Vezi recursivitatea

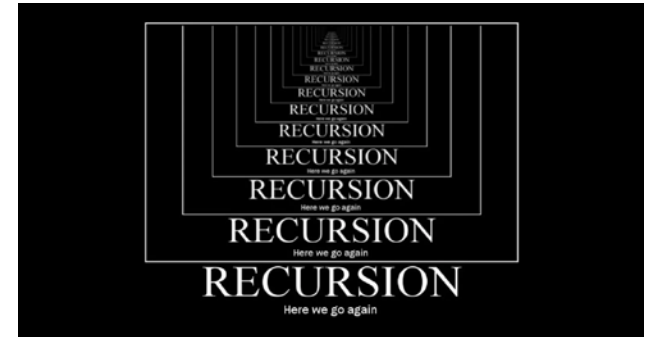
- Definiția se referă la sine însuși (ca un loop)

- Cele mai faimoase exemple sunt:

- Fibonacci series: $f_n = f_{n-1} + f_{n-2}$ 1, 1, 2, 3, 5, 8, 13,

- Factorial: $n! = n * (n - 1)!$ $5! = 5 * (4 * (3 * (2 * (1)))) = 120$

- În Python: o funcție care se apelează pe sine însuși



PROGRAMAREA UNEI FUNCȚII RECURSIVE

- Sunt 2 părți în recursivitate:
 - Cazul de bază → un caz cunoscut
 - Uneori sunt mai multe cazuri de bază
 - Cazul recursiv → tot restul

```
def recursiveFunction():  
    if (this is the base case):  
        return something non-recursive  
    else:  
        return something recursive
```

RECURSIVITATEA: FACTORIAL

- Caz de bază: $\text{factorial}(1) = 1$ (i.e. $1! = 1$)
- Caz recursiv: returnează $n * (\text{factorial}(n-1))$

```
def factorial(n):  
    if (n == 1):  
        return 1  
    else:  
        return n*factorial(n-1)
```

RECURSIVITATEA: FIBONACCI

- Cazul de bază 1: $\text{fibonacci}(1) = 1$
- Cazul de bază: $\text{fibonacci}(2) = 1$
- Caz recursiv: întoarce $\text{fibonacci}(n-1) + \text{fibonacci}(n-2)$

```
def fibonacci(n):  
    if (n == 1):  
        return 1  
    elif (n == 2):  
        return 1  
    else:  
        return fibonacci(n-1) + fibonacci(n-2)
```

PROVOCAREA: PELL SEQUENCE

- Creează o funcție recursivă pentru a primi o valoare nth value în secvența Pell
- Secvența Pell este 0, 1, 2, 5, 12, 29, 70, 169, 408, 985,
- Matematic, este definită ca

$$P_n = 2 * P_{n-1} + P_{n-2}$$

- Printează numărul 5th PELL pe matricea LED

SOLUȚIA PROVOCĂRII

```
from hub import light_matrix
import runloop, sys

# Funcția pentru a opri programul utilizând o excepție de sistem
def stopAndExitProgram():
    sys.exit("Stopping")

def PELL(n):
    if (n < 1):
        return "Invalid"
    elif (n <= 2):
        return n - 1
    else:
        return 2 * PELL(n-1) + PELL(n-2)

async def main():
    await light_matrix.write(str(PELL(5))) # convertește numărul la un șir înainte de a scrie
    stopAndExitProgram()

runloop.run(main()))
```


CREDITS

- Această lecție a fost creată de Sanjay Seshan și Arvind Seshan for SPIKE Prime Lessons
- La această lecție au contribuit membrii comunității FLL Share & Learn.
- 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/).