
Introducció al pensament computacional a través de la Programació

Python (pyTurtle) i eSeeCode



60 min



Jornades
Educació Matemàtica

Definició

Hi ha moltes definicions de pensament computacional.

La que utilitzarem està basada en la de la Janette Wing (2017)

“El **pensament computacional** és el procés de pensament que s'utilitza al formular un problema i expressar les seves solucions de tal forma que un *ordinador* - humà o màquina - pot executar-ho.”

En el marc de l'escola també interpretem el pensament computacional com a un llenguatge estructurat. D'aquesta manera hem de pensar en l'ensenyament d'aquesta disciplina com l'aprenentatge d'una llengua no materna. No es busca precipitar-se en els coneixements formals, sinó en canvis d'actituds i predisposició.

Com ens enfrontem a un problema de PC

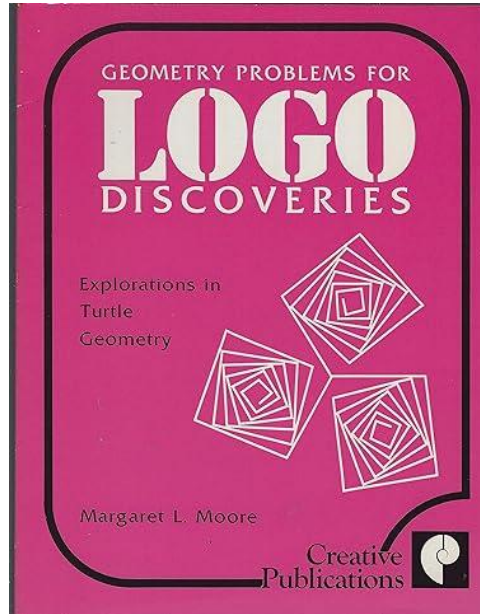
Quan es planteja un repte de programació podem seguir una estratègia similar a Pólya:

1. **Entendre el problema** - conèixer el què es vol aconseguir.
Ho podem fer a mà?
2. **Dissenyar una estratègia** - entendre el flux del programa.
Dibuixar un diagrama o esquema
3. **Execució de l'estratègia** - programar l'estratègia.

Programació fent Baby steps.

4. **Avaluar la solució** - executar el programa.
El resultat és el què volia?

I tot això és nou?



Logo provides a unique environment for creating geometric experiences. [...] But, for a student to give instructions for the turtle to draw a square, the student must have some understanding of the attributes of a square. If the student's definition has irrelevant attributes, the students instructions to the turtle will have irrelevant directions. [...] *Logo* provides an environment for **testing** definitions, for **refining** definitions, and for **verbalizing** definitions.

Margaret L. Moore *Logo Discoveries*, 1984

Resolució de problemes

Lògica

Execució

Descomposició

Desenvolupar

Algoritmes

Avaluar

Reconeixement de patrons

Abstracció

Entorn de treball

Escollir entorns online i multiplataforma facilita la gestió.

Buscar entorns gratuïts.

Dos exemples:

1.- Python: <https://trinket.io>



2.- eSeeCode: <https://eseecode.com>

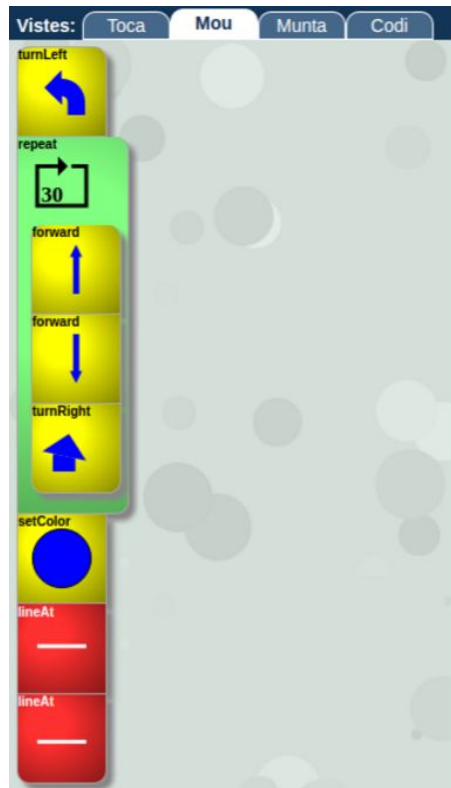


Plataforma de creació pròpia.

Per tal de reforçar l'idea que el llenguatge és secundari, està bé mostrar-ne més d'un.

eSeeCode: Llenguatge propi educatiu

Vistes: Toca Mou Munta Codi



turnLeft
repeat
forward
forward
turnRight
setColor
lineAt
lineAt

The toolbar contains several icons: a blue arrow pointing left for 'turnLeft', a green square with a black arrow and the number '30' for 'repeat', a blue arrow pointing up for 'forward', a blue arrow pointing down for 'forward', a blue arrow pointing right for 'turnRight', a blue circle for 'setColor', and two red rectangles with white lines for 'lineAt'.

Vistes: Toca Mou Munta Codi

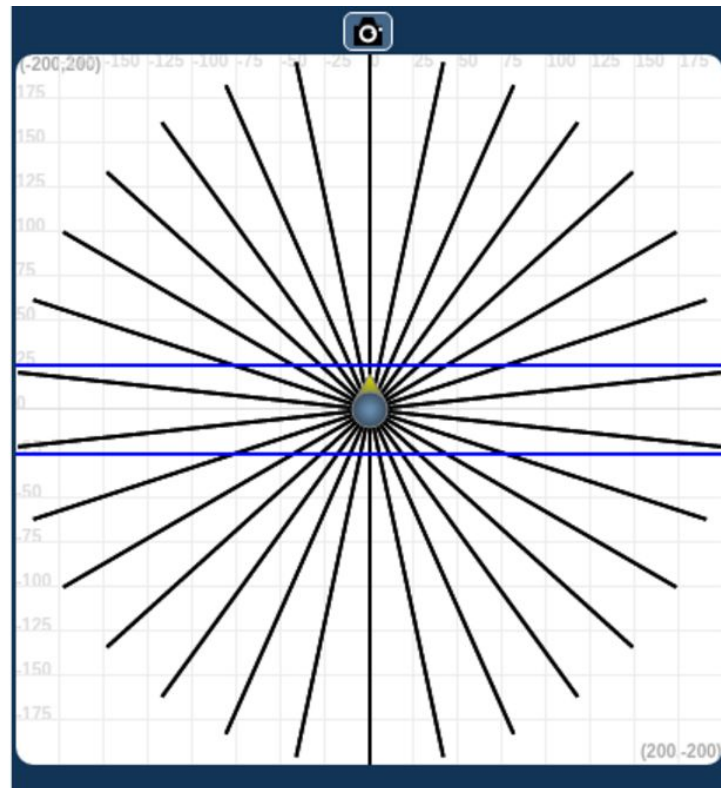
```
turnLeft(90)  
repeat (30) {  
  forward(200)  
  forward(-200)  
  turnRight(12)  
}  
setColor("blue")  
lineAt(-200, 25, 200, 25)  
lineAt(-200, -25, 200, -25)
```

The screenshot shows the 'Munta' (Edit) view of the code editor. The code is displayed in a light green background with syntax highlighting: 'turnLeft(90)' is yellow, the 'repeat' block is green, 'setColor("blue")' is yellow, and the 'lineAt' commands are red.

Vistes: Toca Mou Munta Codi

```
1 turnLeft(90)  
2 - repeat (30) {  
3   forward(200)  
4   forward(-200)  
5   turnRight(12)  
6 }  
7 setColor("blue")  
8 lineAt(-200, 25, 200, 25)  
9 lineAt(-200, -25, 200, -25)  
10
```

The screenshot shows the 'Codi' (Code) view of the code editor. The code is displayed in a white background with a dark blue border. The code is the same as in the previous screenshot, but with line numbers on the left side.



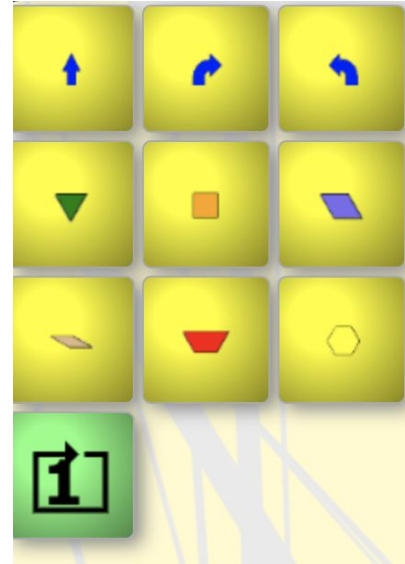
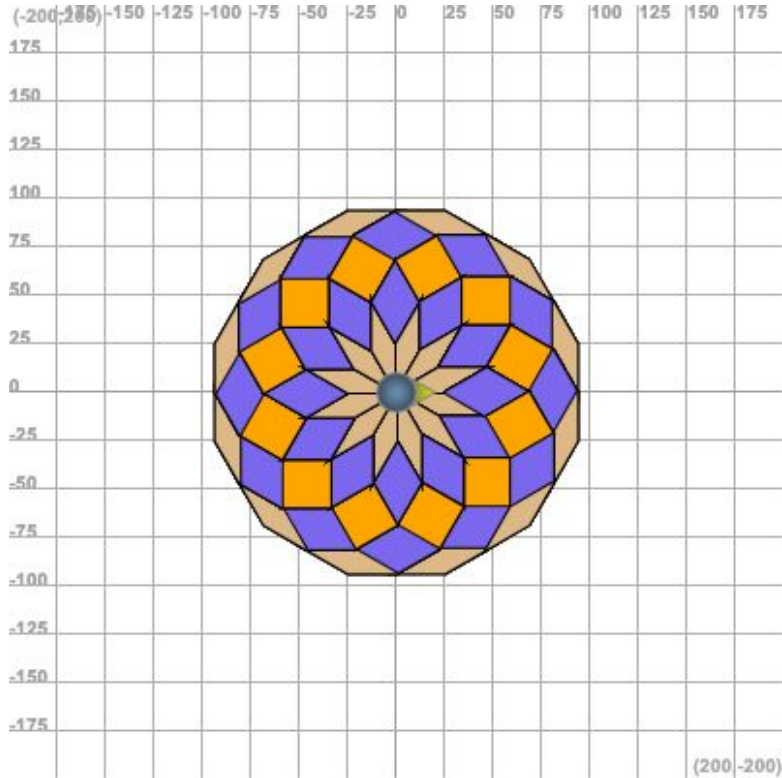
The screenshot shows the 'Toca' (Touch) view of the code editor. The drawing area is a grid with a central blue dot. The drawing consists of a fan of lines radiating from the center, with a vertical line and two horizontal lines. The lines are black, and the horizontal lines are blue. The grid has x and y coordinates ranging from -200 to 200.

Problemes per practicar

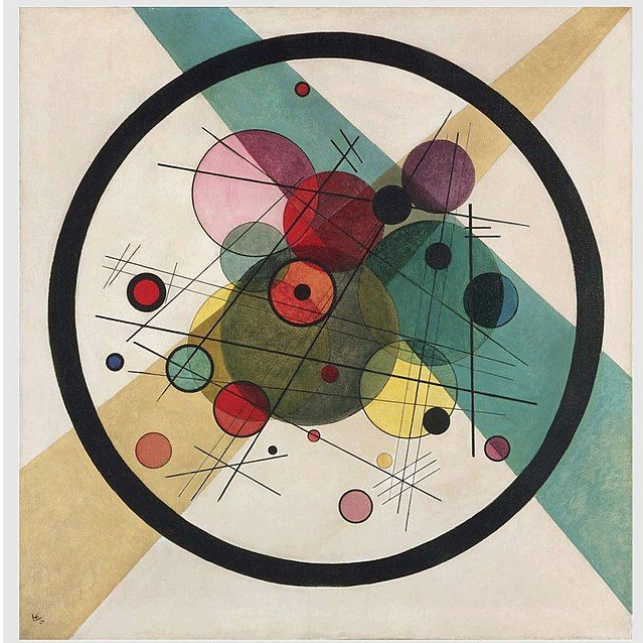
Pattern Blocks (Primària)



<https://eseeco.de/mfj5q>



Som artistes (6è primària)

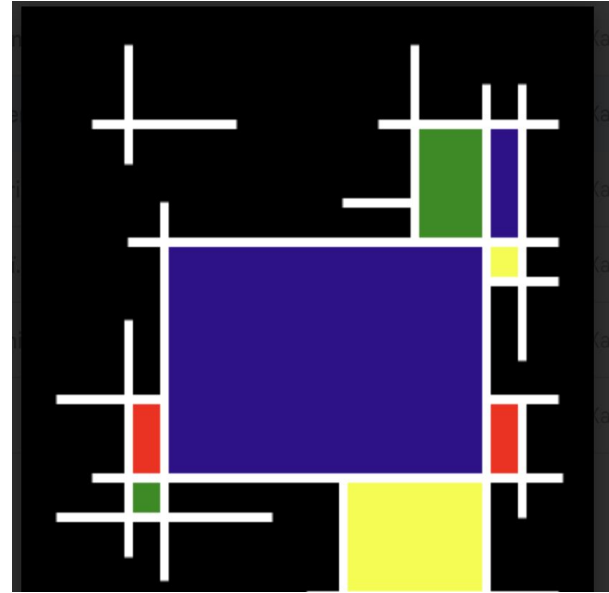
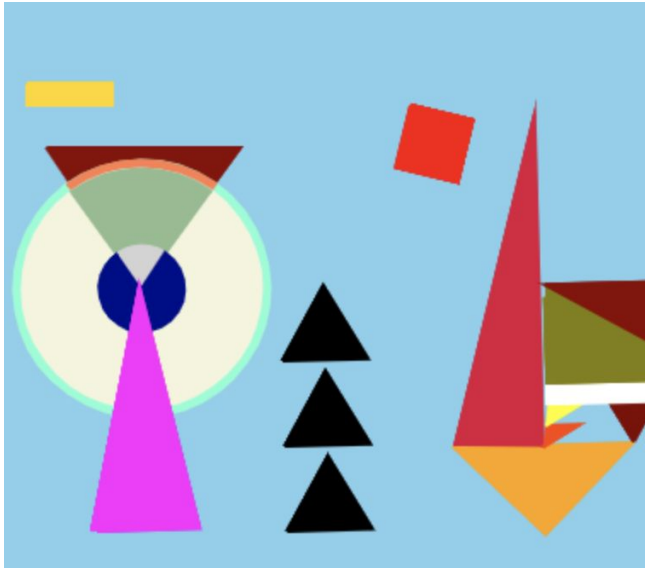


Circles in a Circle W. Kandinsky



Treball de Daniel (11 anys)

Som artistes (6è primària)

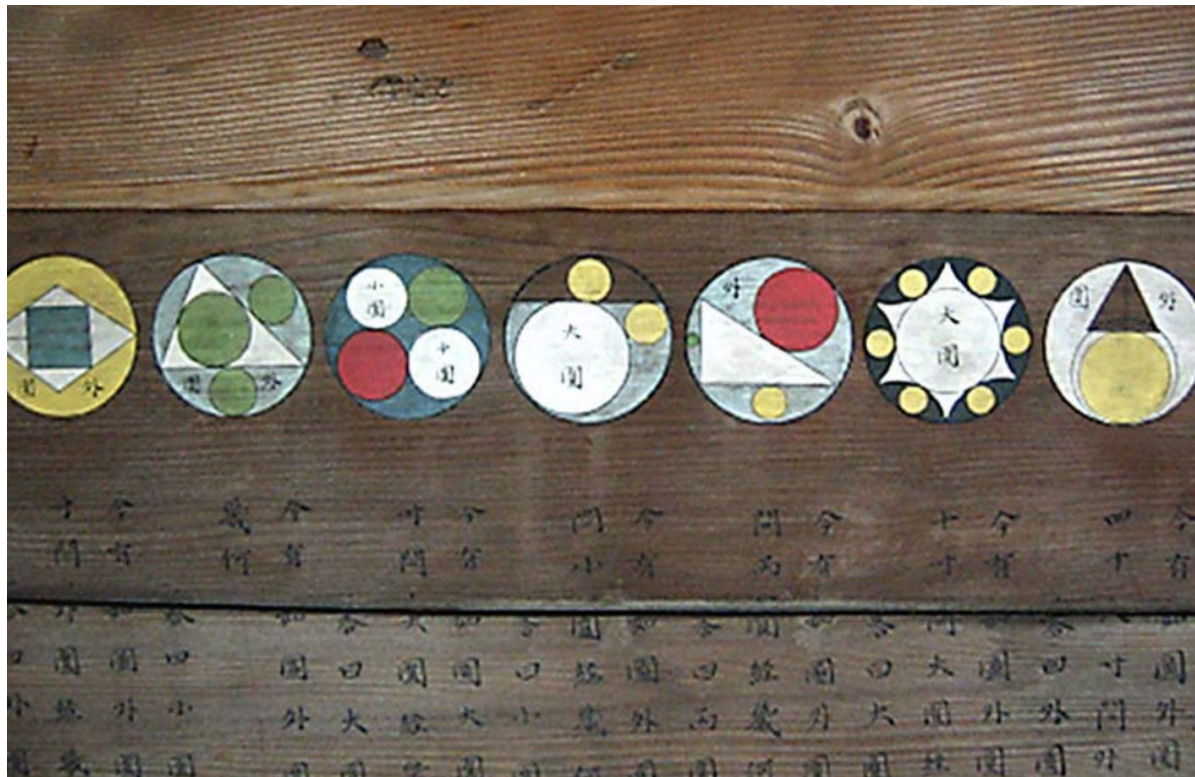


Altres treballs de 6è

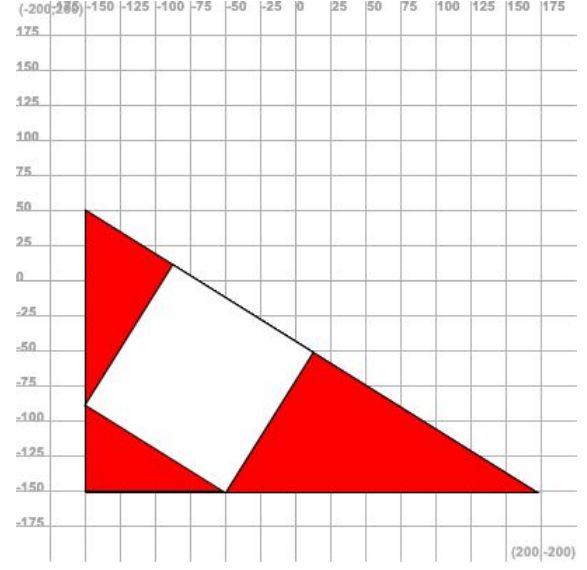
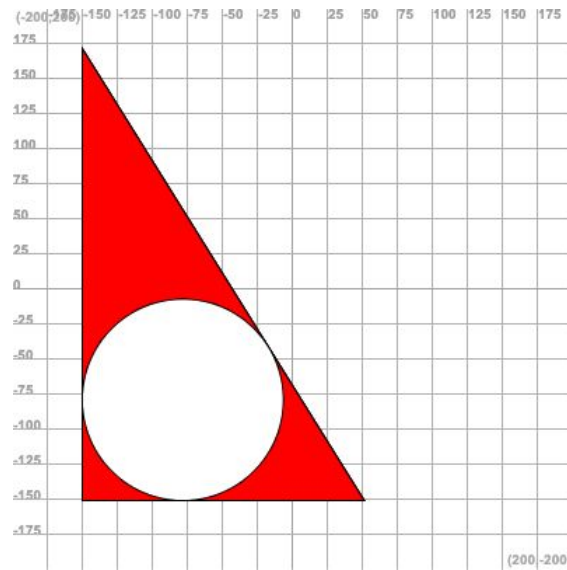
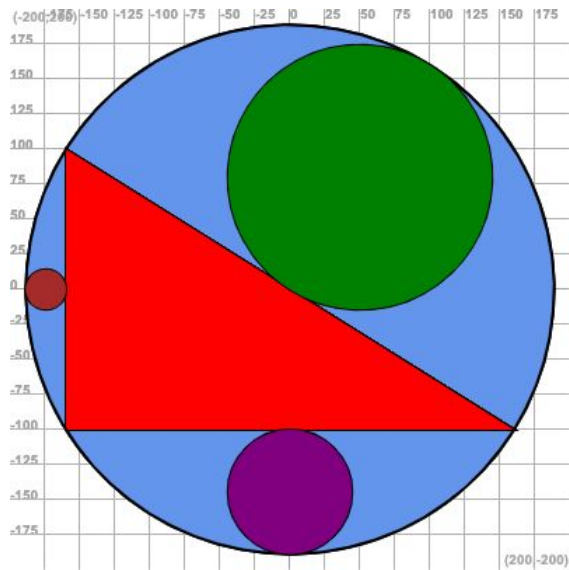
Geometria Japonesa Sangakus (ESO-Batxillerat)

Tauletes de fusta penjades als temples de Japó.

Aquí una tauleta de Suwa (Nagasaki)

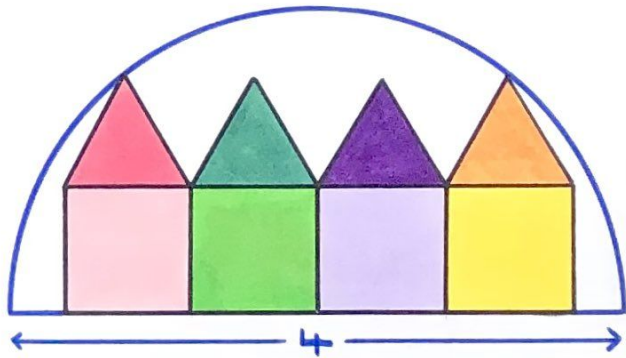


Geometria Japonesa Sangakus (alguns exemples)

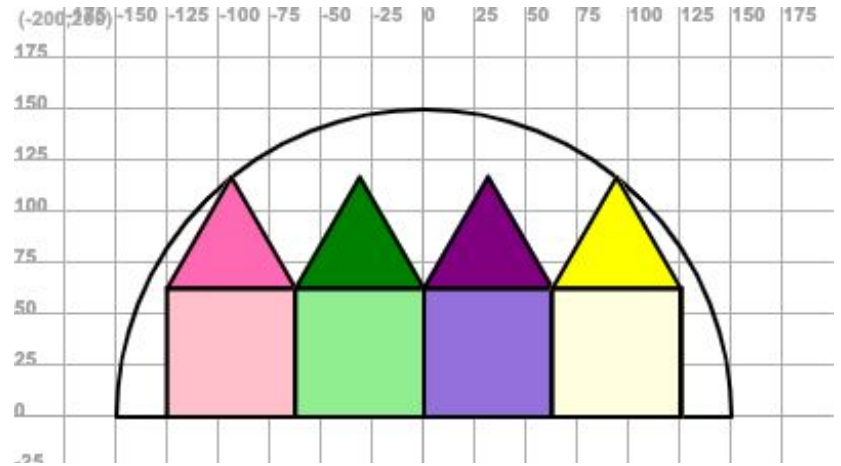


“Sangakus” moderns: Catriona Agg

<https://twitter.com/Cshearer41>



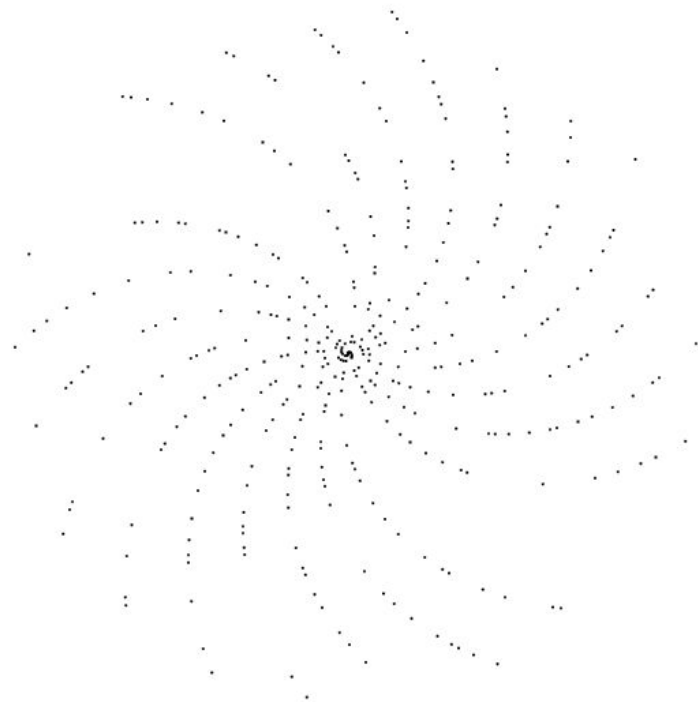
Original



Versió per ordinador

Explorem teoria de nombres (Batxillerat)

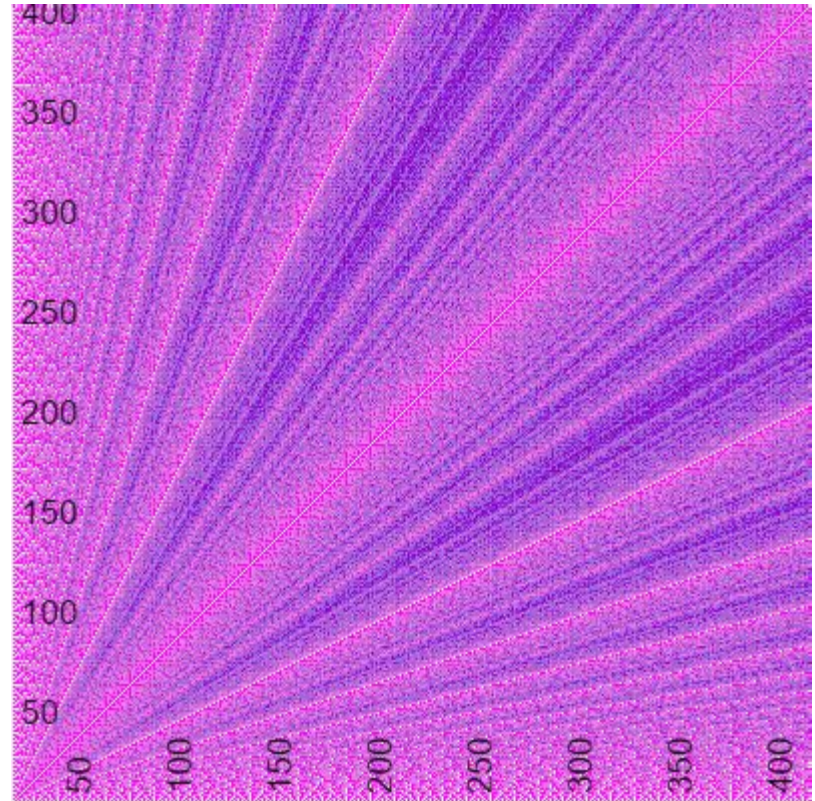
Podem dibuixar els nombres
primers en coordenades
polars.



<https://www.youtube.com/watch?v=EK32jo7i5LQ&t=33s>

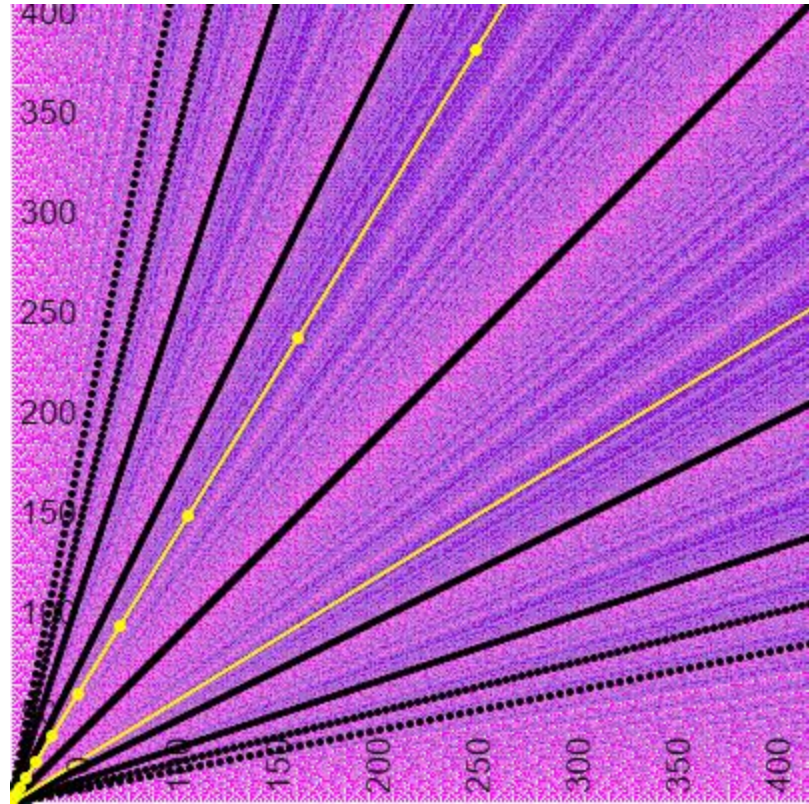
El màxim comú divisor

1. Considerem una graella de 400x400 punts.
2. Per un punt donat (x,y) calcularem quantes crides necessitem per calcular el màxim comú divisor de x i y .
3. Pintarem aquest punt amb un color específic depenent del nombre de passos.
4. Nota considerarem el nombre de passos de x i y el mateix que per a y i x , encara que difereixin en una unitat.



El màxim comú divisor

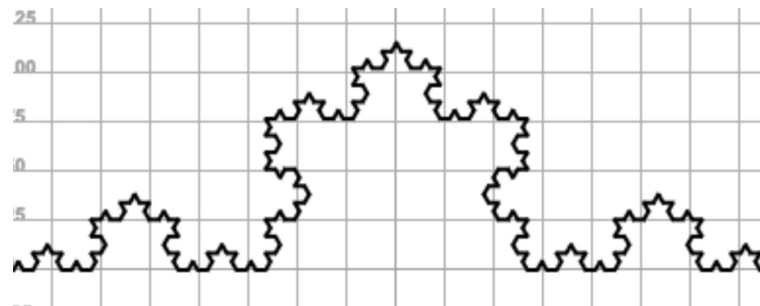
1. Considerem una graella de 400x400 punts.
2. Per un punt donat (x,y) calcularem quantes crides necessitem per calcular el màxim comú divisor de x i y .
3. Pintarem aquest punt amb un color específic depenent del nombre de passos.
4. Nota considerarem el nombre de passos de x i y el mateix que per a i y i x , encara que difereixin en una unitat.



Altres idees:

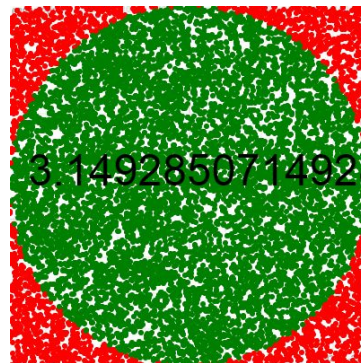
Recursivitat per entendre l'inducció:

<https://eseecode.com/en/merry-everything>



Atzar i probabilitat:

<https://eseecode.com/en/content/pi-day-2016>



I ara? Posem-nos a jugar!

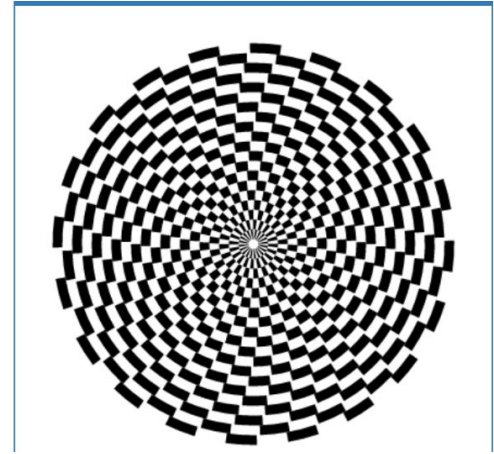
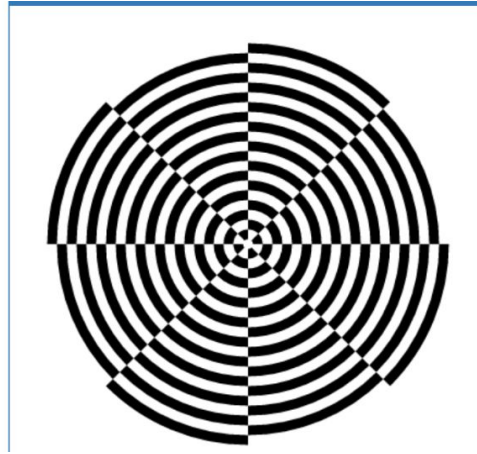
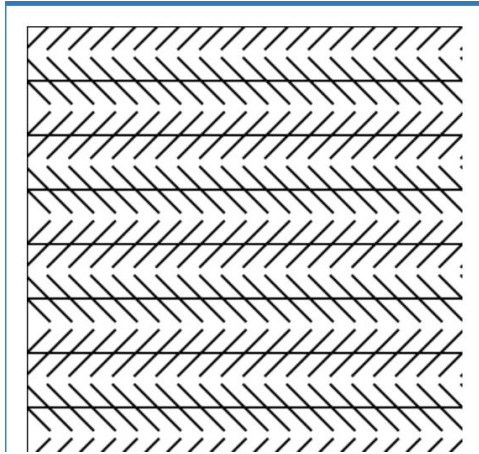
Gràcies!

Joan Alemany Flos jalemany@aula-ee.com

Jacobo Vilella Vilahur jvilella@aula-ee.com

Una idea més:

Il·lusions òptiques!



Les pròximes diapositives són com introduir-ho a classe:

Comencem amb paper!

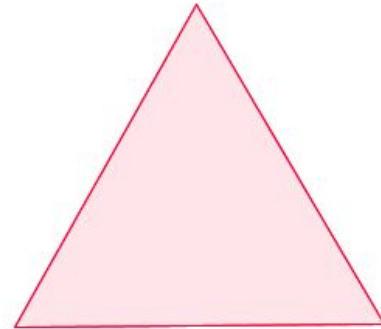
Recordem que l'ordinador no té sentiments i que fa el què li diguem.

Possibles preguntes

Pregunta 1: Com podríem fer un triangle equilàter de costat 100?

Instruccions usades:

- **forward**
- **turnLeft**

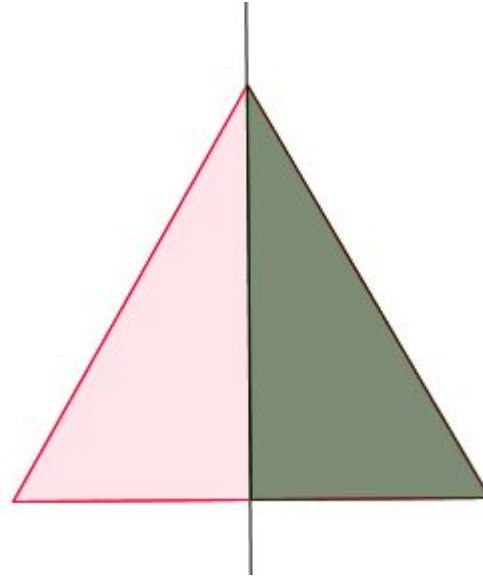


Possibles preguntes

Pregunta 2: Com podem fer la meitat d'aquest triangle equilàter (costat 100)?

Instruccions usades:

- forward
- turnLeft
- **getSquareRoot**

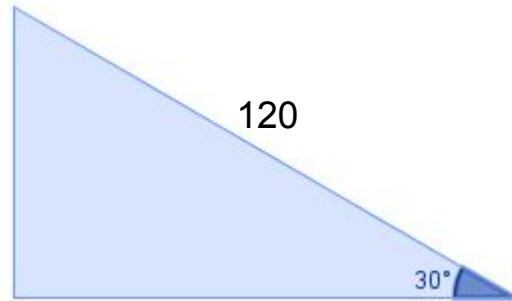


Possibles preguntes

Pregunta 3: Com podríem fer un triangle rectangle de costat 120 i angle 30° ?

Instruccions usades:

- forward
- turnLeft
- getSquareRoot
- **var x**

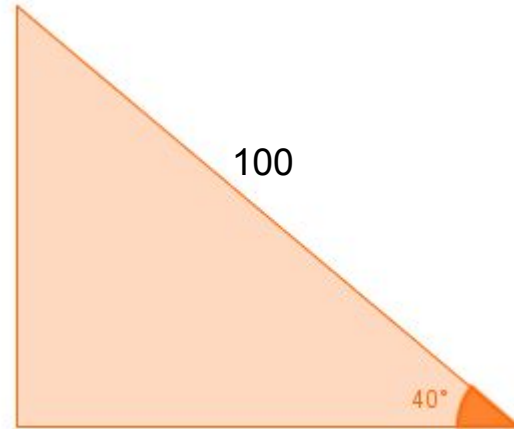


Possibles preguntes

Pregunta 4: Com podríem fer un triangle rectangle de costat 100 i angle 40° ?

Instruccions usades:

- forward
- turnLeft
- getSquareRoot
- var x
- **getCosine**

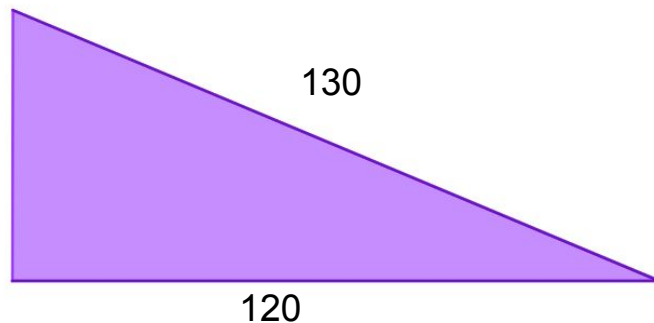


Possibles preguntes

Pregunta 5: Com podríem fer un triangle rectangle sabent que el catet contigu i la hipotenusa mesuren 120 i 130, respectivament?

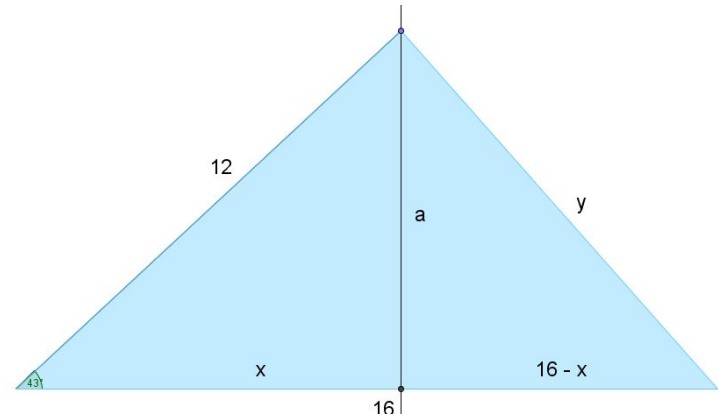
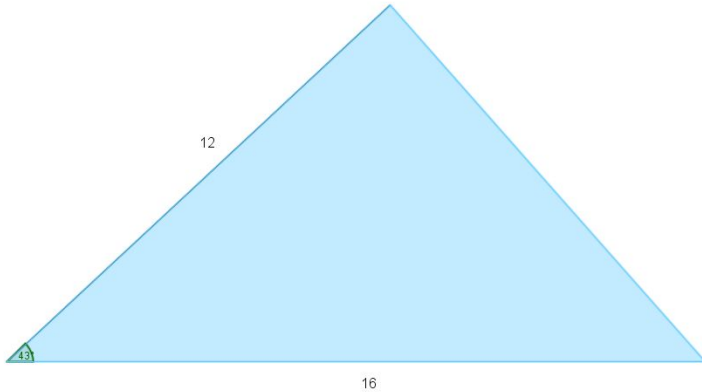
Instruccions usa:

- forward
- turnLeft
- getSquareRoot
- var x
- **getArcCosine**



Possibles preguntes

Pregunta 6: Com podem fer un triangle qualsevol, sabent que un angle mesura 43° i els costats que el formen mesuren 120 i 160 cm?



Primer problema (apte per totes les edats!)

Fes un programa per dibuixar la següent figura:

