Sistemas-L

"Felipe" +

" x x x x x x x "

F → l
e → i i → f
l → e
p → i

"Felipe" → "liefii"
→ "efilff"
→ ""

"F" (forward, avanzar)

REGLAS { "F" : "F<F>>F<F" }

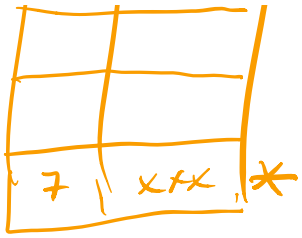
"F" 1→ "F<F>>F<F" 2→ "F<F>>F<F<F<F>>F<F>>F<F>>F<F<F<F>>F<F"

3→

1	8
2	36

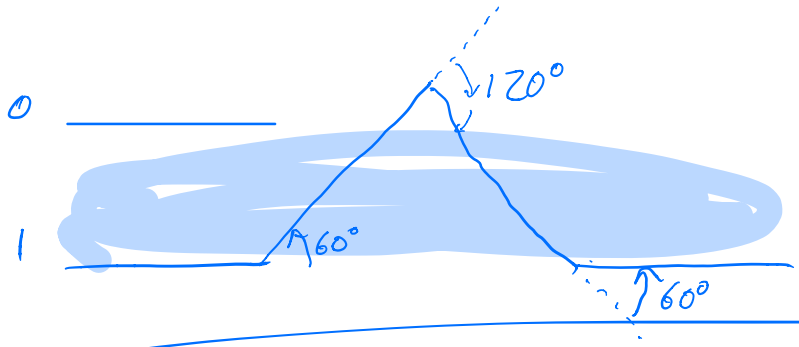
TRADUCCIÓN

F = avanza 1 paso

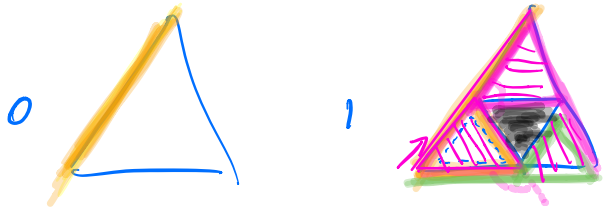


< = gira 60° antihorario

> = gira 60° horario



→ Koch



"(" = escala ½x

")" = escala 2x

F → (F>>F>>F>>)F

Python

(Tipos de datos, estructuras de datos)

Listas

[1, 2, 3]

["manzana", "plátano"]

[1, "manzana", 7, "Plátanos"]

mutables

Tuplas

(1, 2, 3)

("manzana", "plátano")

inmutables (no se pueden modificar)

Diccionarios

claves: "apple", "pear"

valores: "manzana", "pera"

d = {"apple": "manzana",
"pear": "pera",
}

"onion" : "cebolla" }

inmutables

```
print( d["pear"])
```
→ "pera"

```
print( d.get("onion"))
```
→ "cebolla"

```
print(d ["banana"])
```
→ ERROR!!!

```
print(d.get("banana"))
```

```
a=[1, 2, 3, 4, 5, 6]
print (a[0])
```
→ 1 `print(a[5])` → 6

b= [1, 4, 9, 16, 25, 36, ..., 10000]

1ª forma (tradicional)

```
b = [ ]
for i in range(1,101):
    b.append(i**2)
```

0, 1, 2, ..., 99

2ª forma (Pytonada) Pythonics

"Expansión de lista"

```
b= [i**2 for i in range(1,101)]
```

EJERCICIOS

① ranar (100 , 0 , -1)

② `[3*x+2 for x in range(10)]`

③
```
d={1:2, 2:3, 3:1}
print(d[1]+d[2]*d[3])
```

④ `'-'.join(['a', 'ante', 'bajo', 'con', 'contra', 'de'])`

TRADUCCION

F = avanza un paso

K = retrocede un paso

< = gira izq (antihorario) 60°

> = gira der 60°

+ = gira izq 90°

- = gira der 90°

L = gira izq 10°

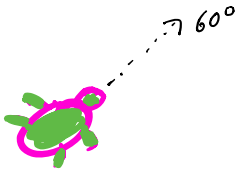
R = gira der 10°

(= escala el paso a 1/2

) = escala el paso a 2

[= inicia rama

] = termina rama

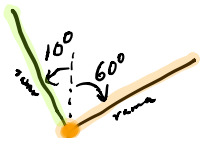


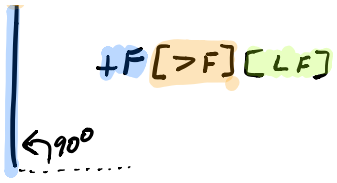
escala = 1
posicion = (0,0)
~~angulo = 0~~

"<"
angulo = 60
"F"

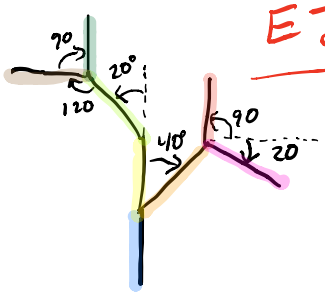
posicion = posicion + escala * (cos angulo, sen angulo)

= (0,0) + 1 * (cos 60, sen 60)





EJERCICIO



Agrega esta rama al programa